

Context-Aware and Adaptable eLearning Systems

PhD Thesis

Stanimir N. Stoyanov

Software Technology Research Laboratory

De Montfort University

Leicester - United Kingdom

March 2012

Abstract

This thesis proposed solutions to some shortcomings to current eLearning architectures. The proposed DeLC architecture supports context-aware and adaptable provision of eLearning services and electronic content. The architecture is fully distributed and integrates service-oriented development with agent technology. Central to this architecture is that a node is our unit of computation (known as eLearning node) which can have purely service-oriented architecture, agent-oriented architecture or mixed architecture. Three eLearning Nodes have been implemented in order to demonstrate the vitality of the DeLC concept. The *Mobile eLearning Node* uses a three-level communication network, called InfoStations network, supporting mobile service provision. The services, displayed on this node, are to be aware of its context, gather required learning material and adapted to the learner request. This is supported through a multi-layered hybrid (service- and agent-oriented) architecture whose kernel is implemented as middleware. For testing of the middleware a simulation environment has been developed. In addition, the DeLC development approach is proposed. The second eLearning node has been implemented as *Education Portal*. The architecture of this node is purely service-oriented and it adopts a client-server architecture. In the education portal, there are incorporated education services and system services, called *engines*. The electronic content is kept in Digital Libraries. Furthermore, in order to facilitate content creators in DeLC, the environment *Selbo2* was developed. The environment allows for creating new content, editing available content, as well as generating educational units out of preexisting standardized elements. In the last two years, the portal is used in actual education at the Faculty of Mathematics and Informatics, University of Plovdiv. The third eLearning node, known as *Agent Village*, exhibits a purely agent-oriented architecture. The purpose of this

node is to provide intelligent assistance to the services deployed on the Education Portal. Currently, two kinds of assistants are implemented in the node - *eTesting Assistants* and *Refactoring eLearning Environment (ReLE)*. A more complex architecture, known as *Education Cluster*, is presented in this thesis as well. The Education Cluster incorporates two eLearning nodes, namely the Education Portal and the Agent Village. eLearning services and intelligent agents interact in the cluster.

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the software Technology Research Laboratory (STRL), at De Montfort University, United Kingdom.

No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

This thesis is written by me and produced using MS Word.

Acknowledgments

I am thankful to all the people who helped and inspired me during the preparation of this thesis.

Most importantly, I would like to express my profound thanks and eternal gratitude to my supervisors, Professor Hussein Zedan and Dr Feng Chen, without whose support, encouragement and guidance this thesis would have been a hard way for me.

I would like to thank all my colleagues at the Distributed eLearning Center (DeLC) group – University of Plovdiv for their constant support and responsiveness, especially Emil Doychev, Damyan Mitev, Ivan Minov, Dr. Asya Stoyanova-Doycheva, Veselina Valkanova, Vladimir Valkanov and for help by implementation of the architecture proposed in the thesis. I had the opportunity to participate in the "Software Engineering: Computer Science Education and Research Cooperation" project supported by DAAD. I would like to thank my colleagues in this project and especially Prof. Klaus Bothe (University of Humboldt, Berlin) for the successful collaboration. I wish to thank my partners from Telecommunications Research Centre (University of Limerick, Ireland) for the good teamwork within the SEDiLia project.

Last but not least, I would like to thank all colleagues at the Software Technology Research Laboratory for stimulating this research and the support whenever needed, especially Mrs. Lynn Ryan and Mrs. Lindsey Trent.

Selected Publications Included in the Thesis

- [1] Ganchev I., S. Stojanov, M. O'Droma, D. Meere, Development of InfoStation-based and Context-aware mLearning System Architectures, In: Advanced Learning. Raquel Hijón-Neira, editor. In-Teh. ISBN: 978-953-307-010-0, 2009 Pp. 115-139. (to be included – 25 pages)
- [2] S. Stoyanov, I. Ganchev, D. Mitev, V. Valkanov, M. O'Droma, Service-oriented and Agent-based Architecture Supporting Adaptable, Scenario-based and Context-aware Provision of Mobile e-Learning Services, International Journal of Computer Information Systems and Industrial Management Applications, Volume 3, Dynamic Publishers, Inc., USA, 2011, ISSN 2150-7988, pp. 771-779. (to be included – 9 pages)
- [3] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, A. Elci, M. T. Kone, M. A. Orgun (Eds.): “Semantic Agent Systems: Foundations and Applications”, Book Series: Studies in Computational Intelligence, Vol. 344, Springer Verlag, 2011, ISBN: 978-3-642-18307-2 (to be included – 30 pages)
- [4] Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma, An Approach for the Development of a Context-Aware and Adaptive eLearning Middleware, V. Sgurev et al. (Eds.): Intelligent Systems: From Theory to Practice, SCI 299, Springer-Verlag, Berlin-Heidelberg, 2010, ISBN: 978-3-642-13427-2. DOI: 10.1007/978-3-642-13428-9_26. Pp. 519-535. (to be included – 17 pages)
- [5] S.Stoyanov, I.Ganchev, I.Popchev, I.Dimitrov, Request Globalization in an InfoStation Network, Compt. Rend. Acad. Bulg. Sci., 63, No 6, 2010, Pp.901-908. (to be included – 7 pages)

- [6] I.Ganchev, S.Stojanov, M.O'Droma, D.Meere, Infostation-Based Adaptable Provision of M-Learning Services: Main Scenarios, International Journal "Information Technologies and Knowledge" Vol.2 / 2008, Pp. 475-482.
- [7] I. Dimitrov, S. Stoyanov, I. Popchev, Optimal Resource Deployment in an InfoStation-Based Network, International Conference REMIA 2010, University of Plovdiv, 2010, pp. 247-253. (to be included – 5 pages)
- [8] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, V. Valkanov, A. Stoyanova-Doycheva, V. Valkanova , I. Minov, DeLC Educational Portal, Cybernetics and Information Technologies (CIT), Vol.10, No 3., Bulgarian Academy of Sciences, 2010, pp. 49-69 (to be included – 16 pages)
- [9] D.Mitev, S. Stoyanov, I. Popchev, Selbo2 – An Environment for Creating Electronic Content in Software Engineering, Cybernetics and Information Technologies (CIT), Vol.9, No 3., Bulgarian Academy of Sciences, 2009, pp. 96-105. (to be included – 10 pages)
- [10] S. Stoyanov, G. Cholakov, V. Valkanova, M. Sandalski, Personalized, Reactive and Proactive Providing of e-Learning Services, EdiLib Conference , Hawaii, 14-17 October, 2011. (to be included – 8 pages)
- [11] S.Stoyanov, A. Stoyanova-Doycheva, I. Popchev, M. Sandalski, ReLE - A Refactoring Supporting Tool, Compt. Rend. Acad. Bulg. Sci., Tome 64, № 7, 2011, pp. 1017-1026. (to be included – 8 pages)
- [12] S. Stoyanov, H. Zedan, E. Doychev, V. Valkanov, I. Popchev, G. Cholakov and M. Sandalski, Intelligent Distributed eLearning Architecture, V. M. Koleshko (Ed.), Intelligent Systems, InTech, March, 2012, 978-953-51-0054-6, Hard cover, 366 pages, pp. 185-218.

Publications Referred to the Thesis

Chapters in Books

- [1] S. Stoyanov, H. Zedan, E. Doychev, V. Valkanov, I. Popchev, G. Cholakov and M. Sandalski, Intelligent Distributed eLearning Architecture, V. M. Koleshko (Ed.), Intelligent Systems, InTech, March, 2012, 978-953-51-0054-6, Hard cover, 366 pages, 185-218.
- [2] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, A. Elci, M. T. Kone, M. A. Orgun (Eds.): "Semantic Agent Systems: Foundations and Applications", Book Series: Studies in Computational Intelligence, Vol. 344, Springer Verlag, 2011, ISBN: 978-3-642-18307-2.
- [3] Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma, An Approach for the Development of a Context-Aware and Adaptive eLearning Middleware, V. Sgurev et al. (Eds.): Intelligent Systems: From Theory to Practice, SCI 299, Springer-Verlag Berlin Heidelberg, 2010, ISBN: 978-3-642-13427-2. DOI: 10.1007/978-3-642-13428-9_26, 519-535.
- [4] D. Meere, I. Ganchev, M. O'Droma, M. O'hAodha, S. Stojanov, "Evolution of Modern Library Services: The Progression into the Mobile Domain", In M-Libraries 2: A virtual library in everyone's pocket, M. Ally and G. Needham, Eds.: Facet Publishing, May 2010. ISBN: 978-1-85604-696-1, 61-72.
- [5] Ganchev I., S. Stojanov, M. O'Droma, D. Meere, Development of InfoStation-based and Context-aware mLearning System Architectures, In: Advanced Learning. Raquel Hijón-Neira, editor. In-Teh. ISBN: 978-953-307-010-0, 2009, 115-139.
- [6] I. Ganchev, M. O'Droma, D. Meere, M. Ó hAodha, S. Stojanov, "M-learning and m-teaching architectures and the integration of evolving multi-campus educational support e-

services.” In “M-libraries.” Gill Needham and Mohamed Ally, editors. September 2008, 352pp ISBN: 978-1-85604-648-0, 159-172.

Journals

- [7] M. Sandalski, A. Stoyanova-Doycheva, I. Popchev, S.Stoyanov, Development of a Refactoring Learning Environment, Cybernetics and Information Technologies (CIT), Vol.11, No 2, Bulgarian Academy of Sciences, 2011, 46-64.
- [8] S. Stoyanov, Education Portal for Secondary School, Journal „Education and Technologies”, 2/2011, 71-73 (in Bulgarian).
- [9] V. Valkanova, M. Kostova, S. Stoyanov, Training of Students by Help of Education Portal, Journal „Education and Technologies”, 2/2011, 96-98 (in Bulgarian).
- [10] S. Stoyanov, I. Ganchev, D. Mitev, V. Valkanov, M. O’Droma, Service-oriented and Agent-based Architecture Supporting Adaptable, Scenario-based and Context-aware Provision of Mobile e-Learning Services, International Journal of Computer Information Systems and Industrial Management Applications, Volume 3, Dynamic Publishers, Inc., USA, (2011), ISSN 2150-7988, 771-779.
- [11] S.Stoyanov, A. Stoyanova-Doycheva, I. Popchev, M. Sandalski, ReLE - A Refactoring Supporting Tool, Compt. Rend. Acad. Bulg. Sci., Tome 64, № 7, 2011, 1017-1026.
- [12] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, V. Valkanov, A. Stoyanova-Doycheva, V. Valkanova , I. Minov, DeLC Educational Portal, Cybernetics and Information Technologies (CIT), Vol.10, No 3., Bulgarian Academy of Sciences, 2010, 49-69.
- [13] S.Stoyanov, I.Ganchev, I.Popchev, I.Dimitrov, Request Globalization in an

InfoStation Network, Compt. Rend. Acad. Bulg. Sci., 63, No 6, 2010, 901-908.

- [14] D.Mitev, S. Stoyanov, I. Popchev, Selbo2 – An Environment for Creating Electronic Content in Software Engineering, Cybernetics and Information Technologies (CIT), Vol.9, No 3., Bulgarian Academy of Sciences, 2009, 96-105.
- [15] Meere D., I. Ganchev, S. Stojanov, M. O'Droma. 2009. "Adaptation for Assimilation: The Role of Adaptable M-Learning Services in the Modern Educational Paradigm," in International Journal "Information Technologies and Knowledge", Vol. 3, Pp. 101-110. ISSN: 1313-0455.
- [16] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, An Approach for the Development of InfoStation-Based eLearning Architectures, Compt. Rend. Acad. Bulg. Sci., 61, No 9, 2008, 1189-1198.
- [17] S. Stoyanov, V.Valkanova , I. Popchev , I.Minov, A Scenario-Based Approach to Creating a Virtual Environment for Secondary School Instruction, CIT 3, 2008, 86-96.
- [18] I.Ganchev, S.Stojanov, M.O'Droma, D.Meere, Infostation-Based Adaptable Provision of M-Learning Services: Main Scenarios, International Journal "Information Technologies and Knowledge" Vol.2 / 2008, 475-482.

Refereed Conferences/Workshops

- [19] S. Stoyanov, E. Doychev, A. Stoyanova-Doycheva, V. Valkanova, V. Valkanov, Education Cluster Supporting eTesting and eLearning in Software Engineering, 2nd Annual International Conference on Web Technologies & Internet Applications (WebTech 2012), 7- 8 May 2012, Bali, Indonesia (to print).
- [20] S. Stoyanov, E. Doychev, V. Valkanova, G. Cholakov, Education Cluster for Intelligent Provision of eLearning Services, DBKDA 2012 : The Fourth International

Conference on Advances in Databases, Knowledge, and Data Applications, 29 February – 5 March 2012, Reunion, IARIA, 2012. ISBN: 978-1-61208-185-4, 45-50.

- [21] M. Nachev, S. Stojanov, Variant of Creating of Mobile Context-Aware Services by Using of Intelligent Agent and Infostation Architecture, Military Technologies and Systems (MT&S 2011) , 8-9 December 2011, Sofia.
- [22] S. Stoyanov, G. Cholakov, V. Valkanova, M. Sandalski, Personalized, Reactive and Proactive Providing of e-Learning Services, EdiLib Conference , AACE E-Learn 2011 – World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, Honolulu, Hawaii, USA, 17-21 October, 2011, 2527-2534
- [23] S.Stoyanov, V. Valkanova, G. Cholakov, M. Sandalski, Education Portal for Reactive and Proactive Service Provision, COGNITIVE 2011: The Third International Conference on Advanced Cognitive Technologies and Applications, 25-30 September, 2011, Rome, 99-103, ISBN: 978-1-61208-155-7
- [24] S. Stoyanov, V. Valkanova, Intelligent Assistants for Students, International Conference „Basic Competences in Education – Strategies and Practice”, 22-24.09.2011, Stara Zagora, Bulgaria, ISBN 978-954-691-071-4 (in Bulgarian).
- [25] V. Valkanova, S. Stoyanov, Using Electronic Tools for Interest Increasing in Mathematics, International Conference „Basic Competences in Education – Strategies and Practice”, 22-24.09.2011, Stara Zagora, Bulgaria, ISBN 978-954-691-071-4 (in Bulgarian).
- [26] S. Stoyanov, I. Popchev, M. Sandalski, G. Cholakov, eLearning Intelligent Environment, International Conference „Relationship Theory and Practice: Key Problems and Solutions”, Free University in Bourgas, 24-25.06.2011, 251-258 (in Bulgarian).

- [27] V. Valkanova, I. Popchev, M. Sandalski, S. Stoyanov, Education Portal for training of Students, International Conference „Relationship Theory and Practice: Key Problems and Solutions”, Free University in Bourgas, 24-25.06.2011, 287-294 (in Bulgarian).
- [28] S. Stoyanov, M. Sandalski, I. Popchev, G. Cholakov, E. Doychev, Personalized and Proactive Provision of eLearning Services in the DeLC Education Portal, 4th National Conference „Education in the Information Society”, 26-27 May, Plovdiv, 119-128 (in Bulgarian).
- [29] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, G. Cholakov, Architecture of the DeLC Education Portal, 4th National Conference „Education in the Information Society”, 26-27 May, Plovdiv, 129-138 (in Bulgarian).
- [30] Meere D., I. Ganchev, M. O'Droma, S. Stoyanov, S. Hayes, P. Noonan, Mobile Phones as Tools to Enhance the Traditional Educational Experience, Proc. of the IADIS International Conference on E- Learning 2011 (EL 2011), 22-26 July 2011, Rome, Italy.
- [31] S. Stoyanov, M. Sandalski, V. Valkanova, A. Stoyanova-Doycheva, E. Doychev, Environment for Provision of eLearning Services, International Conference „Electronic, Distance ... or Education of the 21st Century”, 6-8 April, 2011, Sofia, 207-214 (in Bulgarian).
- [32] S. Stoyanov, Context-Aware and Adaptable Architecture CA³, International Conference REMIA 2010, University of Plovdiv, 2010, pp. 97-104.
- [33] I. Dimitrov, S. Stoyanov, I. Popchev, Optimal Resource Deployment in an InfoStation-Based Network, International Conference REMIA 2010, University of Plovdiv, 2010, pp. 247-253.
- [34] S. Stoyanov, V. Valkanova, I. Ganchev, M. O'Droma, An Approach and Architecture Supporting Context-Aware Provision of mLearning Services, IEEE Second International

Conference on Mobile, Hybrid, and On-line Learning, 10-16 February 2010, St. Maarten, 11-16.

- [35] S. Stoyanov, I.Ganchev, V.Valkanova, M.O'Droma, Scenario-Oriented and Context-Aware M-Learning System Architecture, IADIS International Conference Mobile Learning 2010, Porto, Portugal, 250-254.
- [36] V. Valkanova, S. Stoyanov, H. Zedan, I. Popchev, A Model for Examination Students' Creative Thinking and Acting, 39th SBM Conference, 6-10.04.2010, Albena, 274-280 (in Bulgarian).
- [37] D. Meere, I. Ganchev, M. O'Dróna, S. Stojanov, V. Valkanova, 2010 "An Enhanced Context-Sensitive InfoStation-Based mLearning Architecture", 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST '10), 13-16 April 2010, Dubai, UAE.
- [38] D. Meere, I. Ganchev, M. O'Dróna, S. Stojanov, V. Valkanova, Adaptation for Assimilation: Shaping Context-Sensitive M-Learning Services within a Multi-Agent Environment, 2010 Sixth Advanced International Conference on Telecommunications, 9-15 May 2010, Barcelona, Spain, 74-79.
- [39] S. Stoyanov, I. Ganchev, I.Popchev, M.O'Droma, V. Valkanova, Agent-Oriented Middleware for InfoStation-based mLearning Intelligent Systems, 5th IEEE International Conference on Intelligent Systems IS'10, 07.07. – 09.07.2010, London, IEEE Catalog Number: CFP10802-CDR, ISBN:978-1-4244-5164-7, Library of Congress:2009934065, 91-95.
- [40] I.Ganchev, S. Stoyanov, V. Valkanova, M. O'Droma, Service-oriented and Agent-based Architecture Supporting Adaptable Context-Aware Provision of Mobile e-Learning Services, IADIS e-Learning 2010 Conference, Freiburg, Germany, 26 - 29 July 2010, 97-104.

- [41] I. Ganchev, S. Stoyanov, M. O'Droma, V. Valkanova, D. Meere, Pervasive InfoStation-based mLearning System, ICSNC 2010: The Fifth International Conference on Systems and Networks Communications, August 22-27, 2010 - Nice, France, 320-325.
- [42] I. Ganchev, S.Stoyanov, M. O'Droma,V.Valkanova, Context-Aware mLearning Service Execution in an InfoStations Environment, 2009 Fourth International Conference on Internet and Web Applications and Services (ICIW 2009), 25-28 June 2009, Venice/Mestre, Italy, pp. 632-637.
- [43] V.Valkanova, S.Stoyanov, H.Zedan, A Creativity Teaching Model for Secondary School, The Second International Conference on Creativity and Innovation in Software Engineering (CISE'09 in association with IEEE and IFIP), Ravda (Nessebar), Bulgaria, 10 - 12 June 2009.
- [44] S. Stoyanov, I. Ganchev, I. Popchev, M. O'Droma, Extended DeLC Approach for the Development of InfoStation-based eLearning Architectures, The Second International Conference on Creativity and Innovation in Software Engineering (CISE'09 in association with IEEE and IFIP), Ravda (Nessebar), Bulgaria, 10 - 12 June 2009.
- [45] D.Meere, I. Ganchev, M. O'Droma, S. Stoyanov, On Development of Mobile Personal Assistants within an InfoStation-based Personalized and Contextualized mLearning Environment, The Second International Conference on Creativity and Innovation in Software Engineering (CISE'09 in association with IEEE and IFIP), Ravda (Nessebar), Bulgaria, 10 - 12 June 2009.
- [46] S.Stoyanov, I.Ganchev, D.Mitev, I. Minov, V. Valkanov, T.Velev, M. Kalinov, G.Kraev, Multi-agent architecture for InfoStation infrastructure, , The Second International Conference on Creativity and Innovation in Software Engineering (CISE'09 in association with IEEE and IFIP), Ravda (Nessebar), Bulgaria, 10 - 12 June 2009.

- [47] S.Stoyanov , A. Stoyanova-Doycheva, M. Chigrichenko, Improving the creativity of Software Engineering education with Refactoring Learning Environment, The Second International Conference on Creativity and Innovation in Software Engineering (CISE'09 in association with IEEE and IFIP), Ravda (Nessebar), Bulgaria, 10 - 12 June 2009.
- [48] Ganchev I., M. O'Droma, D. Meere, M. Ó hAodha, S. Stojanov. 2009. "Evolution of Modern Library Services: The Progression into the Mobile Domain". Presented at the 2nd International M-Libraries Conference "Information on the move", 23-24 June, Vancouver BC, Canada.
- [49] I.Ganchev, D. Meere, M. O'Droma, S. Stoyanov, Approaches to the Development of an InfoStation-based mLearning System, Proceedings of the 2009 International Conference on E-Learning, E-Business, Enterprise Information Systems, & E-Government (EEE'09), WORLDCOMP'09, July 13-16, 2009, Las Vegas Nevada, USA, 10-15.
- [50] S.Stoyanov, I.Ganchev, M.O'Droma, H.Zedan, V.Valkanova, Agent-Oriented Middleware for Mobile eLearning Services, proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC 2009), Seattle Washington USA, 20-24 July, 2009, 62-66.
- [51] D. Meere, I. Ganchev, S. Stojanov, M. O'Droma, Adaptation for Assimilation: The Role of Adaptable M-Learning Services in the Modern Educational Paradigm, International Book Series "Information Science and Computing", Number 14, 2009, 101-110.
- [52] K. Bothe, K. Schützler, Z. Budimac, M. Ivanovic, Z. Putnik, S. Stoyanov, A. Stoyanova-Doyceva, K. Zdravkova, B. Jakimovski, D. Bojic, I. Jurca, D. Kalpic, B. Cico, Experience with shared teaching materials for software engineering across countries, Informatics Education Europe IV (IEE IV) 2009, November 5-6, Freiburg, Germany.
- [53] I. Ganchev, M. O'Droma, D. Meere, S.Stoyanov, M. Ó hAodha, On InfoStation-Based Mobile Services Support for Library Information Systems, The 8th IEEE International

Conference on Advanced Learning Technologies (ICALT'08), 1-5 July 2008, Santander, Spain, 679-681.

- [54] I.Ganchev, S. Stojanov, D. Meere, M.O'Droma, InfoStation-based mLearning System Architectures: Some Development Aspects, The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08), 1-5 July 2008, Santander, Spain, 504-505.
- [55] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, Service-oriented and Agent-based Approach for the Development of InfoStation eLearning Intelligent System Architectures, IS 2008 – IEEE International Conference on Intelligent Systems Varna, Bulgaria, September 6-8, 2008, 6-20 – 6-25.
- [56] S.Stoyanov, D.Mitev, I.Minov, T.Glushkova, eLearning Development Environment for Software Engineering Selbo 2, 19th International Conference on Database and Expert Systems Application (DEXA 2008), 1-5 September, Turin, Italy, 2008, 100-104.
- [57] S. Stoyanov, I. Ganchev, M. O'Droma, D. Mitev, I. Minov, Multi-Agent Architecture for Context-Aware mLearning Provision via InfoStations, The 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, October 28-31, 2008, Paris, 549-552.

Contents

CHAPTER 1 INTRODUCTION	27
1.1 MOTIVATION	28
1.2 THESIS SCOPE AND RESEARCH QUESTION	29
1.3 RESEARCH METHODOLOGY	31
1.4 CONTRIBUTION TO KNOWLEDGE	31
1.5 MEASURE OF SUCCESS	32
1.6 THESIS STRUCTURE	32
CHAPTER 2 ELEARNING SYSTEMS: STATE-OF-THE-ART	36
2.1 INTRODUCTION	37
2.2 E-SERVICE INTELLIGENCE	37
2.3 ELEARNING SYSTEMS AND SUPPORTING TECHNOLOGIES	38
2.3.1 <i>Virtual Learning Environments</i>	40
2.3.2 <i>The Benefits of VLEs</i>	42
2.3.3 <i>The Challenges of VLEs</i>	44
2.3.4 <i>The Architecture of VLEs</i>	46
2.3.5 <i>Features and Tools of VLEs</i>	48
2.3.5.1 Communication Tools	49
2.3.5.2 Productivity Tools	51
2.3.5.3 Learner Involvement Tools	51
2.3.5.4 Administration Tools	52
2.3.5.5 Course Delivery Tools	52
2.3.5.6 Curriculum Design	53
2.3.5.7 Hardware/Software	54
2.3.5.8 Pricing/Licensing	54
2.3.6 <i>Comparative Study of VLE Products</i>	55
2.3.6.1 Criteria for Evaluating VLEs	56
2.3.6.2 Features and Capabilities	56
2.3.6.3 Summary of Comparison	60

2.3.6.4	Technical Aspects.....	63
2.3.7	<i>Next Generation of VLEs.....</i>	65
2.4	MOBILE LEARNING (MLEARNING).....	67
2.5	DIGITAL LIBRARIES	71
2.6	SUMMARY	72
CHAPTER 3 DISTRIBUTED ELEARNING CENTER.....		73
3.1	INTRODUCTION	74
3.2	MODELING OF DeLC	74
3.2.1	<i>Conceptual Model.....</i>	74
3.2.2	<i>DeLC Infrastructure.....</i>	76
3.2.3	<i>Enhanced DeLC Network.....</i>	77
3.2.4	<i>DeLC Services.....</i>	79
3.3	THE DEVELOPMENT OF DeLC	85
CHAPTER 4 MOBILE ELEARNING NODE.....		88
4.1	INTRODUCTION	89
4.2	DEVELOPMENT APPROACH.....	89
4.3	SYSTEM ARCHITECTURE.....	92
4.4	CONTEXT-AWARE SERVICE PROVISION IN DeLC.....	94
4.5	SCENARIOS LEVEL.....	96
4.6	MIDDLEWARE.....	97
4.7	EDUCATION SERVICES LEVEL.....	102
4.8	RESOURCE DEFICIT ITERATION.....	110
4.9	SIMULATION ENVIRONMENT	112
4.10	SUMMARY.....	117
CHAPTER 5 DeLC EDUCATION PORTAL.....		119
5.1	INTRODUCTION	120
5.2	PERSONALIZATION	120
5.2.1	<i>Users</i>	121
5.2.2	<i>Electronic content</i>	122

5.2.3	<i>Events</i>	124
5.2.4	<i>Services</i>	125
5.3	PORTAL ARCHITECTURE	127
5.4	eCONTENT DEVELOPMENT ENVIRONMENT	130
5.5	SUMMARY	139
CHAPTER 6 EDUCATION CLUSTER.....		140
6.1	INTRODUCTION	141
6.2	REACTIVE AND PROACTIVE SERVICE PROVISION IN DELC.....	141
6.3	AGENT VILLAGE NODE.....	142
6.3.1	<i>System Infrastructure</i>	143
6.3.2	<i>eTesting Assistants</i>	144
6.3.3	<i>Refactoring eLearning Environment</i>	145
6.4	NODES INTERACTION IN THE EDUCATION CLUSTER	148
6.5	SUMMARY	151
CHAPTER 7 CONCLUSION AND FUTURE WORK.....		152
7.1	RESEARCH SUMMARY	153
7.2	RELATED WORK	154
7.3	EVALUATION	157
7.4	FUTURE DIRECTIONS	160
REFERENCES.....		163
APPENDIX: INCLUDED PAPERS.....		189

List of Figures

FIGURE 1: VLE AS A SUB-SYSTEM WITHIN THE eLEARNING ENVIRONMENT [78]	47
FIGURE 2: USER-EXPERIENCE ARCHITECTURE [76].....	48
FIGURE 3: THE TOTAL FEATURES OF THE TEN VLE PRODUCTS [78]	62
FIGURE 4: CHART OF USE OF WEB-BASED E-LEARNING SYSTEMS IN UNIVERSITIES [78].....	65
FIGURE 5: THE FUTURE OF VLES - THE VISUAL VERSION [77].....	65
FIGURE 6: SERVICES DEPLOYMENT IN THE INFOSTATION NETWORK	70
FIGURE 7: THE DeLC CONCEPTUAL MODEL	75
FIGURE 8: DeLC NETWORK MODEL	77
FIGURE 9: THE 3-TIER INFOSTATION-BASED NETWORK ARCHITECTURE	78
FIGURE 10: DeLC COMMUNICATION INFRASTRUCTURE SUPPORTING M-LEARNING AND M-TEACHING	79
FIGURE 11: THE SERVICE CLASSIFICATION MODEL	80
FIGURE 12: COMMUNICATION INFRASTRUCTURE SUPPORTING THE MOBILE SERVICE 'INTELLIGENT MESSAGE NOTIFICATION'	82
FIGURE 13: INTELLIGENT MESSAGE NOTIFICATION SERVICE PROVISION	83
FIGURE 14: DeLC INFRASTRUCTURE.....	86
FIGURE 15: THE APPROACH: LEVELS AND ITERATIONS	90
FIGURE 16: THE LAYERED SYSTEM ARCHITECTURE.....	93
FIGURE 17: THE AGENT-ORIENTED MIDDLEWARE ARCHITECTURE	99
FIGURE 18: TWO SERVICE MODELS	103
FIGURE 19: THE MAPPING BETWEEN OWL-S AND WSDL	104
FIGURE 20: THE OWL-S SERVICE ONTOLOGY	105
FIGURE 21: mTEST SERVICE PROVISION	108
FIGURE 22: APPROACH DIAGRAM	112
FIGURE 23: THE LOGICAL STRUCTURE OF THE SIMULATION ENVIRONMENT	113
FIGURE 24: THE ARCHITECTURE OF THE EXPERIMENT RUNNER (ER)	116
FIGURE 25: USERS' CLASSIFICATION SCHEME	121
FIGURE 26: RELOAD EDITOR.....	125
FIGURE 27: DeLC PORTAL HOME PAGE.....	128
FIGURE 28: PORTAL ARCHITECTURE	129
FIGURE 29: SELBO 2 ARCHITECTURE.....	133
FIGURE 30: SELBO 2 USER INTERFACE	138

FIGURE 31: AGENT VILLAGE NODE.....	143
FIGURE 32: ReLE ARCHITECTURE	147
FIGURE 33: CLUSTER NODES INTERACTION.....	150

List of Tables

TABLE 1: FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS OF VIRTUAL LEARNING ENVIRONMENTS [78].....	47
TABLE 2: MAIN FEATURES AND TOOLS OF VLEs [78].....	49
TABLE 3: GENERAL INFORMATION ABOUT THE SELECTED PRODUCTS [78].....	56
TABLE 4: CRITERIA FOR EVALUATING VLEs [78]	57
TABLE 5: SUMMARIES OF THE FEATURES AND CAPABILITIES OF VLE TOOLS [78].....	58
TABLE 6: THE COMPARISON BETWEEN SELECTED VLE PRODUCTS BASED ON LEARNER TOOLS [78].....	58
TABLE 7: THE COMPARISON BETWEEN SELECTED VLE PRODUCTS BASED ON SUPPORT TOOLS [78]	60
TABLE 8: THE COMPARISON BETWEEN SELECTED VLE PRODUCTS BASED ON TECHNICAL SPECIFICATIONS TOOLS [78].....	61
TABLE 9: THE FINAL RESULT OF THE COMPARISON BETWEEN TEN VLE PRODUCTS [78]	61
TABLE 10: COMPARISON BASED ON FOCUSING ON THE TECHNICAL ASPECTS OF THE VLE SYSTEMS [78].....	63
TABLE 11: COMPARISON BASED ON FOCUSING ON SOME FEATURES AND CATEGORIES OF VLEs [78]	64
TABLE 13: RESULTS OF THE ADAPTATION CATEGORY [78].....	64

List of Abbreviations

AAA	Authentication, Authorization, Accounting
ACL	Agent Communication Language
ADL	Advanced Distributed Learning
AMS	Agent Management System
AV	Agent Village
BDI	Belief, Desire, Intention
BE	Back-end
BSD	Business Server Directory
CAM	Content Aggregation Model
CC	Common Cartridge
CC/PP	Composite Capability/Preference Profile
CPI	Capability and Preference Information
DeLC	Distributed eLearning Center
DeTC	Distributed eTesting Center
DF	Directory Facilitator
DL	Digital Library
DOM	Document Object Model
EA	Evaluator Assistant
EO	Experiment Organizer
ER	Experiment Runner
FAM	Functional- Agents Mapping
FE	Front-end

FIPA	Foundation for Intelligent Physical Agents
GUI	Graphical User Interface
HISS	Hospital Information System for Students
HTML	Hypertext Markup Language
IDE	Integrated development Environment
IMS SS	IMS Simple Sequencing
IS	InfoStation
ISC	InfoStation Center
ITL	Interval Temporal Logics
IWT	Intelligent Web Teacher
J2EE	Java 2 Enterprise Edition
JADE	Java Agent DEvelopment Framework
JAR	Java Archive File Format
JDT	Java Development Toolkit
JELD	Java Environment for Learning Design
JPF	Java Plugin Framework
LMS	Learning Management System
LOC	Line of Code
MDA	Model Driven Architecture
MoDCA	Mobile Device Collaboration and Assessment
MTS	Message Transport Service
ODG	Optimal Deployment Graph
OMG	Object Management Group
OSS	Open Source Software
OWL-S	Ontology Web Language for Services

PDA	Personal Digital Assistant
RA	Refactoring Agent
RDF	Resource Description Framework
ReLE	Refactoring eLearning Environment
RG	Resource Globalization
RKB	Refactoring Knowledge Base
RTE	Run Time Environment
SAM	Scenarios-Agents Mapping
SC	Simulation Controller
SCO	Sharable Content Object
SCORM	Shareable Content Object Reference Model
SEM	Simulation Environment Manager
SH	SIS Handler
SIS	Simulated InfoStation
SISP	Simulated InfoStation Platform
SMD	Simulated Mobile Device
SMDP	Simulated Mobile Device Platform
SN	Sequencing and Navigation
SOAP	Simple Object Access Protocol
SPA	Simulated Personal Agent
SSP	Service's Server Part
SUP	Service's User Part
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
USDP	Unified Software Development Process

VAB	Virtual Address Book
VLE	Virtual Learning Environment
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	Extensible Markup Language

Chapter 1

Introduction

Objectives:

- To motivate the need for new generation eLearning systems.
- To identify the thesis scope and the research question.
- To present the research metrology.
- To highlight the original contribution of the thesis.

With the advent of the Internet and the continual advances in electronic devices, **eLearning** mode of education and learning has emerged as a vital alternative to the traditional face-to-face mode of education. This is in addition to various forces that helped to increase in their popularity and uptake. These forces include economic volatility which has forced learner to improve their education prospect with the hope to improve their social and standard of living. Furthermore, there are cultural social constraints which necessitate the need for establishing a mode of learning that could be utilised at any-time, any-place and using any device.

One of the major challenges facing eLearning is finding **new architectural solutions** that allow providing of education services and teaching material in a more personalized and adaptable way.

1.1 Motivation

Many eLearning systems have been proposed in the last years. Most of them take care mainly of electronic presentation of the teaching material. The student doesn't stay in the focus of considerations. However there are three dimensions that play crucial role in the next generation of eLearning systems. These are:

- **Advances in technologies:** technological advances is continually improving at an alarming rate - from mainframes to PCs (including laptops and palmtops), from handheld devices (PDs, iPads) to smart mobile phones. Also technological advances in telecommunication has revolutionized the way these devices are connecting and communicating together – forming wired networks to wireless networks and ad-hoc networks.

These advances provide an interesting and challenging dimension for the current eLearning systems, algorithms, architectures and correctness.

- **Economic down-turn:** Current economics is certainly volatile. This necessitates the need to further educate population while still at work.
- **Culture and religion constraints:** Culture and religion dimension plays a crucial role for motivating eLearning mode of education. In some cultures such as the Muslims culture, it is nearly impossible for females to leave their homes to attend traditional face-to-face education. In addition, in other cultures and due to biological constraint female learners do not have the time for traditional education, as they are primarily responsible for bringing up families.

Taking this consideration into account, fundamental requirement of the next generation of eLearning systems is that they have to be *highly personalized, adaptable, and intelligent*. Indeed, this conforms with our philosophy of learning *any-where, any-time and device- and context-independent*.

1.2 Thesis Scope and Research Question

The work in this thesis focuses on the following issues:

- **eLearning systems:** This research is based on critically analyzing the state-of-the-art in the field of eLearning. Alluding to some shortcomings of the existing systems and in the light of modern teaching requirements, we take the view that the new generation of eLearning systems are follows:
 - Student is a central point of a successful and effective education process.
 - Encourage the student for active collaboration in the education process.
 - Minimize the unfavorable effects of the dependences caused by factors such as time, distance, various type of user devices.

- **Software architectures:** We believe that an appropriate system architecture is an important factor for creating personalized, adaptable and intelligent eLearning systems. The research considers the lack of good architectural support in many cases. However, this research investigates the challenges regarding the successful architectural support. The use of hybrid multi-level architectures, especially an integration of service- and agent-oriented, is stated and demonstrated.
- **Context-awareness and adaptability:** Our research focuses on the notion of context-awareness and adaptability. We take the view that, personalized and intelligent eLearning systems could be successfully implemented by means of context-aware and adaptive architectures.

In conformity with motivation and scope, the following main research question can be formulated:

How can context-aware and adaptable architecture be used in the development of personalized, adaptable and intelligent eLearning systems?

In order to cope with complexity of this research programme, a number of research sub-questions have been identified. These sub-goals can be summarized as follows:

- Identify the correct nature of context.
- Establish a unified methodology to link between service-oriented architectural framework with agent-based computing.
- Identify the anatomy of adaptability. To this extend a taxonomy of adaptation and mobility need to be established.
- Realisation of such a taxonomy within the established architecture.
- Design and perform a number of experiments to demonstrate our approach.

1.3 Research Methodology

Our methodology follows a typical software Engineering steps. These are:

- Critical engagement with the literature appeared in the learned Journals and Conferences.
- Identify the need and the articulate our generic research question.
- Employ a traditional technique for handling complexity by decomposing our research question into a number and tangible research sub-questions.
- Design a complete computational model within which these questions can be investigated and explored.
- Design and architecture that satisfies/implement this model.
- Build a prototype to realise our architecture. In doing so a number of important algorithms may need to be formulated.
- Conduct a number of experimental tests to evaluate our model and results.
- Conduct a number of comparative studies to demonstrate the advantages of our approach.

1.4 Contribution to Knowledge

We take the view that, the contribution to knowledge belongs mainly to the field of **eLearning** system and their supporting environment. In particular, the following contributions could be summarized:

- **Learner-oriented philosophy, in which the learner is at the heart of the system.**
- **eLearning support systems must be mobile, adaptive, context-aware and highly personalised in an integrated fashion.**

This is achieved by introducing (see below for details)

- **Distributed eLearning Center (DeLC) concept.**
- **Three learning nodes with various architectural design; namely pure Service-oriented, purely Agent-oriented and a Hybrid architecture.**

1.5 Measure of Success

Establish a unified approach for integrating agent and Service technologies as applied within the eLearning Environment. Such a unified approach has indeed answered our research question.

We have fully demonstrated that our approach is unique and has many advantages over current eLearning systems as seen in our motivation section above.

In the thesis, we use the term **prototype** to mean a software system in its early stages of development. In our case, the environment was built to test the DeLC concept and architecture. In addition, some components of the environment are using in real education process. Prototyping serves to provide specifications for a future, real eLearning environment.

My personal contribution to the entire eLearning systems, described in this thesis, is the creation of concept and the development approach, building of the models, the design of the architecture and implementation of selected components of the system.

1.6 Thesis Structure

The thesis is structured as follows:

- **Chapter 2.** In this chapter we fully give a detailed critical review of current educational system, software design methodologies, computerized (electronically

based) teaching systems and Internet-based systems. The results of this review has identified many limitations in the current educational systems and there underlined supporting systems. Current systems do not consider the learner as integral part of education process. In addition, in the face of any technological advances (which is at a rapid speed) systems failed. Furthermore, current systems provide no or limited level of personalization. As a result we have articulated a generic research question of which the subsequent chapters will provide an efficient answer.

- **Chapter 3.** In this chapter we give an overview of an architecture for the next generation of eLearning environment and system. This architecture is known as Distributed eLearning Center (DeLC). DeLC enjoins the following properties:
 - Fully distributed;
 - Integrates service-oriented architectures with agent technology;
 - Highly adaptable;
 - “Context” is central to its realization;
 - Provably correct;
 - Compositional.

Central to our architecture is that a node is our unit of computation (known as E-Learning node) which can have a purely service-oriented architecture (known as Education Portal DeLC node), agent architecture (known as Agent Village) or a mixed architecture (known as Mobile eLearning node). Furthermore, a virtual more complex structure, known as Education Cluster, is an important component of DeLC.

The architecture and its properties are described in full within 15 published papers of which two seminal papers are included and the rest (incl. in Bulgarian) are given as references. In the following chapters we detail this architecture and its novelty.

- **Chapter 4.** In this chapter we introduce the concept of *Mobile eLearning Node (InfoStations)*. The Mobile eLearning Node has the following characteristics:
 - Agility;
 - Decision making;
 - Efficient cooperation;
 - Context-aware.

The functionality of mobile eLearning Node is to be aware of its context, gather required learning material and adapted to the learner request. The design of such a node and its analysis have been published in 27 of which we include 5 papers.

- **Chapter 5.** This chapter gives a detailed treatment of education portal DeLC. The architecture of this component is completely service-oriented and it adopts a client-server architecture. It interfaces with end user and the rest of the system. Its functionalities include three major group of eLearning services, *Planning and Organization*, *Education services* and *Reporting and Archiving*.

These services are supported by four runtime engines, *Test Engine*, *Event Engine*, *SCORM 2004 Engine* and *User Profiling Engine*.

Full detail has been reported 7 papers of which 4 are included.

- **Chapter 6.** This chapter deals with Education Cluster which integrates two nodes – Education Portal and Agent Village. Agent Village has the following characteristics:
 - Completely agent-oriented architecture.
 - Has three separate subagents and these are Evaluator (automatically evaluation of open assessments), FraudDetector (identify any possible frauds – acts as

forensic investigator) and Statistician (collate data and prepare statistical information).

- Together with education portal DeLC form a cluster, tightly coupled cluster such a cluster provides data hiding abstraction mechanism – this is seem to be an atomic transaction provider.
- Such a cluster should be noted that a cluster is created dynamically.
- Upon completion the cluster terminates and its constituency become singleton agents.

These results are reported in 4 papers of which 2 are included.

- **Chapter 7.** This chapter is the concluding chapter in which our findings are summarized and placed in the context of related work.

It also provides a list of open search questions as means of future research.

Chapter 2

eLearning Systems: State-of-the-Art

Objectives:

- To review the current state of eLearning systems in use.
- To discuss the underpinning notions of e-Service Intelligence and e-Service Personalization.
- To discuss briefly some trends in the usage of Digital libraries.
- To present some previous results of the author in the field of mLearning.

2.1 Introduction

In this chapter, a critical review of the state-of-the-art in the eLearning research area is presented. The existing eLearning systems and supporting technologies are in the focus of this review. Furthermore, some additional trends are discussed which make an impact on this research area. These trends are:

- e-Service Intelligence.
- Mobile Learning.
- Digital Libraries.

2.2 E-Service Intelligence

In [1], the term “*e-service intelligence*” has been defined as: “... *a new research field that deals with fundamental roles, social impacts and practical applications of various intelligent technologies on the Internet based e-service applications that are provided by e-government, e-business, e-commerce, e-market, e-finance, and **e-learning systems** ...*”. Objects of intensive research are topics of this field as intelligent techniques for e-services, e-service models, e-service management, intelligent web, information presentation and searching in intelligent web, personalization of services, intelligent data mining, intelligent support systems for e-services, e-service evaluation etc. Intelligent techniques and methodologies have already enjoyed considerable success in developing of e-services and they are used rather complementary and synergistic as competitive [2]. The most presented techniques refer to fuzzy logic [3,4,5], neural networks [6,7], evolutionary computation [8], machine learning [9], data mining [10], expert systems [11] etc. **E-service personalization** and **e-service support systems** are the most interesting aspects for the scope of this thesis.

E-service personalization is defined as “...*the process of getting web users’ information*

online, and using the information to tailoring web pages to individual users' preferences and deliver services to the users' needs" [1]. Additional aspect of service personalization, as for example improving the quality of users' interactions with the services, are discussed in [12,13]. Results of research in the field of e-service personalization could be summarized in the following aspects [1]:

- Profile based e-service personalization.
- Link based e-service personalization.
- Content based e-service personalization.
- Structure based e-service personalization.
- Recommendation based e-service personalization.

As e-services become common, many Internet-based **support systems** have been developed to assist users to receive high quality services in different aspects of e-services. These systems mainly perform tasks of intermediation and communication between users and the web [14], and many of them are developed under an intelligent framework. In the literature, different approaches for design and implementation of support systems are proposed. For example, fuzzy linguistic techniques and multi-agent systems are presented in [15, 16, 17]; e-negotiation as a kind of e-service support systems [18, 19, 20]

Another kind of e-service support systems is web-based eLearning systems; a detailed review of these systems is presented in the Section 2.3.

2.3 eLearning Systems and Supporting Technologies

Using the Internet to enhance eLearning has become a trend in modern higher education institutions. eLearning platforms are increasingly becoming a significant part of the strategy for delivering online and flexible training and education [21]. At present, an explosion is

occurring in the demand for eLearning platforms world-wide. The next generation of eLearning needs to provide greater dynamism and flexibility to support today's increasingly personalized eLearning requirements [22, 23].

Over the past decade or so, there has been an increase in the interests and demands for methodologies and technologies for an eLearning mode of education. eLearning has been defined as interactive learning in which the learning content is available online and provides automatic feedback and support to the learner's learning activities and experiences. While recognising that the world at large will continue to use terminology in different and often ambiguous ways, the term Virtual Learning Environment (VLE) is used here to refer to the 'online' interactions of various kinds that take place between learners and educators [24, 25, 26, 27].

eLearning systems are multi-disciplinary systems requiring a group effort, where educators, administrators, and users from a variety of other areas of expertise (e.g. psychologists, sociologists, software and electronic engineers) come together in order to serve a community of learners [28, 29]. eLearning offers institutions a number of benefits, the most important of which is the anytime and anywhere access. In order for current and future generations of personalised VLEs to improve learning efficiency and effectiveness, there are essential requirements that have to be realised. In the last decade, a number of VLEs have been adopted and developed by various institutions [30, 26, 31]. Currently, there are more than 250 well known eLearning packages; some of them are Open Source Software (OSS) as illustrated in Table 13 [30, 32, 33]. Perhaps, the best known systems are Moodle, Ilias, Eduplone, Claroline, SAKAI, WebCT and Bscw. They have wide developer communities and present convincing arguments for considering open source as a realistic alternative to commercial products. One open-source project worthy of note, which has emerged to meet the growing

interest in open-source platforms, is Modular Object-Oriented Dynamic Learning Environment (Moodle) [26, 32, 33]. Moodle is designed around pedagogical principles, namely a social constructivist philosophy using the collaborative possibilities of the Internet [34].

2.3.1 Virtual Learning Environments

With the advent of the Internet and the continual advances in communication technologies, a sharp rise in research and development to establish effective learning and teaching environments [35]. VLEs offer a number of benefits for institutions such as, [28, 32, 36],

- The provision of the anytime, anywhere access philosophy.
- Improved learner's motivation.
- Better integration of information and communication technology tools.
- Opportunities for independent learning.
- Increased parental engagement.

In this section we will outline the benefits of general VLEs, the advantageous and limitation of current VLEs and the challenges that next generation of VLEs are facing.

The production service of VLE systems are a form of eLearning software. They are a set of integrated learning tools designed to support a learner's learning experience [23]. The principal tools of a VLE package contain online support for educators and learners, curriculum mapping, Internet links to outside curriculum resources, learners tracking, and electronic communication [30, 36]. They support a range of learning contexts, ranging from conventional, classroom implementation to off-line or online learning, and distance learning [36].

VLEs are essentially restricted web services that provide various basic functions deemed valuable in the learning process. Usually, a number of tools and navigation aids are provided, with the aim of placing any online educational materials into a clear and well organized structure. The environment provides learners with easy access to online courseware, questionnaires, communication tools, course documents, and lecture notes [28, 27].

The Joint Information Systems Committee has defined VLEs, which seems to be the most widely accepted [32], as: *'A VLE is an electronic system that can provide online interactions of various kinds that can take place between learners and tutors, including online learning'*.

In their early stages, VLE software packages were seen as primarily of interest to the area of eLearning but they are now seen as having a wide range of applications in traditional institutes. This view has been shaped by factors such as [27]:

- The growing number of staff and students with access to the Internet.
- Learning, teaching and quality issues.
- The desire to make more effective use of IT infrastructure in institutions.
- The needs and challenges of teaching more students in larger groups.

This is echoed by [37], which has defined VLEs as follows: *'VLEs are learning management software systems that synthesize the functionality of computer-mediated communications software (e-mail, bulletin boards, newsgroups etc.) and on-line methods of delivering course materials (e.g. the WWW). To date, several different packages have appeared from both leading commercial vendors and university-based projects. Other systems are currently under development'*.

2.3.2 The Benefits of VLEs

As with any technology used in teaching and learning, VLEs have no intrinsic educational value in themselves. The way in which online courses and activities are designed and delivered can add value and increase effectiveness. The main reason why VLEs have become so popular and embedded in many institutions is that because there are actual and quantifiable benefits to be gained from the use of this technology. Ever increasing student numbers is one obvious aspect of Higher Education where VLEs can help. They can maintain good communication and there are opportunities for automated assessment. In terms of widening participation, VLEs can support and offer resources to, for example, part-time or disabled students who cannot travel to the institute campus all the time. Below are some commonly perceived advantages and disadvantages of using VLEs.

Advantages

- There are many advantages of using VLEs for the learner and indeed for the institution.
 - Tutors can use them to manage courses such as tracking student progress, making announcements, issuing timetable information, setting, receiving and marking assignments, creating multiple choice tests, and so on.
 - They also offer huge advantages to the universities and their staff, as they allow them to upload files onto the system for everyone to access; this saves them having to attach documents to e-mails and then sending them to the class.
- Both the educator and the learner can enjoy the privacy of their home environment. The Internet provides cheap and easy access to information sources of huge diversity. Interactivity is offered on a large scale and variety, and this technology even provides drills and exercises for basic skills.

- VLEs enable interaction between learners and instructors almost free of time and location constraints.
- VLEs enable the benefits of integrating individual and group learning facilities, and Web services will offer this for VLE systems.
- Easy online delivery of materials and easy to use for learners and educators.
- Offers flexible support for educators who do not need to be in a fixed time or place to support and communicate with learners.
- Has the potential for new ways of learning and teaching such as active and independent learning, which make use of online communication, online assessment and collaborative learning.
- Makes education available to the wider population.
- VLEs are used for a whole host of things that both educators and learners need:
 - Manage studying by using the online calendar.
 - Learners can submit their work and know when the deadline is.
 - If learners miss a lecture, they can get the slides on the VLE.
 - Contact important people such as admin, lecturers, instructor, etc.

Limitations

Although VLEs offer many advantages for educators, learners and institutes, they also have some limitations, and we will mention some of them as follows:

- Both educators and learners need training in order to be able to use a VLE. As with every new technology, it takes some time to learn and gain confidence.
- VLEs can be a dumping ground for material not designed for delivery online.
- Copyright, legislation and accessibility of materials need to be considered.

- Online support must be carefully planned and learning materials can become outdated.
- VLEs are expected to become more sophisticated and this requires more staffs.
- Hundreds of articles, long lists of links, and high-resolution pictures and videos all delay accessibility and lower satisfaction levels.

2.3.3 The Challenges of VLEs

The majority of VLEs face some challenges and they are still limited to the dissemination of teaching materials. Neither have the strengths of the Internet been fully exploited nor have their functions been fully utilized; for instance, supporting autonomous, exploratory, interactive, personalized and collaborative learning. Nevertheless, many developers have attempted to use the emerging computer and communication technologies to create effective learning and teaching environments [35].

The next generation of VLEs could address the limitations of the current online systems by providing a richer **social context** for online learning [38]. Existing eLearning systems have failed to provide support for the deeper interaction and collaboration necessary to realise a community based learner-centred pedagogy. These current systems have also failed to fully benefit from the powerful graphics and simulation capabilities of modern computers. Current eLearning systems are designed primarily as centralized server architectures and have scalability problems that limit their use for large-scale cross-institution collaboration [39, 38].

Another challenge that faces VLEs is to **identify and maximise the pedagogical possibilities** that distinguish it from traditional classroom learning. This is the challenge of using VLEs for highly productive eLearning and avoiding the temptation of merely replicating the traditional classroom, and instead, of focusing on realising the teaching and learning experiences that VLEs invite and make possible. Like distance learning itself, VLEs are neither a substitute for

nor a complete departure from the traditional classroom, but a compelling extension and development of that architecture and its pedagogies.

As in [40], VLEs try to capitalise on two existing active areas of development. The first is an SMS text messaging extension, which is used to support personalised messaging. The second is a Bluetooth-based communications service, named **BlueZone**, which is used to complement SMS text messaging in order to present an alternative communications platform to learners. The combination of these technologies supports a unified and charge- effective delivery mechanism for communicating with university learners on a large scale.

VLEs should not be designed in a vacuum rather they should match learners' needs and desires as closely as possible, and be adaptable during course progression. Therefore, as in [37, 40], there are a number of challenges within higher education that VLE developers should consider:

- Increasing student numbers.
- Automated assessment.
- Widening participation.
- Improved access to limited resources.

With enormous quantities of diverse learning materials, educators and learners still feel that retrieving and integrating educationally relevant learning materials in VLEs is a challenging assignment. Two significant matters that are often raised are how to increase the degree of relevancy among retrieved learning materials and how to deal with the heterogeneity of varying data sources [28]. Another important technical challenge is the **fusion** of these varying data/media items so that the resulting fused media will have the right educational and training impact. Such fusion has to be done when it is needed and at runtime.

2.3.4 The Architecture of VLEs

There is no consistent stranded architecture for VLE systems. All the available architectures depend on the kind of VLE product and the institution's requirements. Therefore, there are no particularly significant differences between them and every product has similar tools, and activities. The differences lie in the institution's requirements and some integrate one or more VLE product with each other.

Whilst an initial investment of time is needed to establish and configure VLEs, the implementation of the system should reduce administrative workload and free up educators' time for teaching. Table 1 displays some functional and non-functional requirements of VLEs [27].

In this section, we will discuss two VLE architectures as examples used by some institutions. As shown in Figure 1, VLEs act as a gateway to online learning, to support features, and to technical specifications tools, and these are shown below. The diagram is based on the one circulated widely in Autumn 1999 by BECTa, and its main purpose is to position VLE as a sub-system within the range of information systems and processes that a college's eLearning environment includes.

As in [39], the aim is to build particularly multi-user VLEs using the understanding gained from prior efforts in distributed information systems. Users experience a direct view of a locale within much larger and contiguous VLEs, as in Figure 2.

Table 1: Functional and Non-Functional Requirements of Virtual Learning Environments [78]

Functional Requirements	Non-Functional Requirements
<ul style="list-style-type: none"> ➤ VLEs should be compatible with relevant IT systems of the host institution both at a departmental and administrative level. ➤ The user interfaces should be simple, intuitive and easy to use. ➤ VLEs should provide opportunities to improve the quality and variety of teaching and learning that are not being achieved using current methods. ➤ In general, VLEs must offer the educator a dynamic way of teaching a particular subject using the Internet. 	<ul style="list-style-type: none"> ➤ VLEs usability for both learners and educators. ➤ Agility, ease of change for future improvements. ➤ They should realize a community based learner-centered pedagogy.

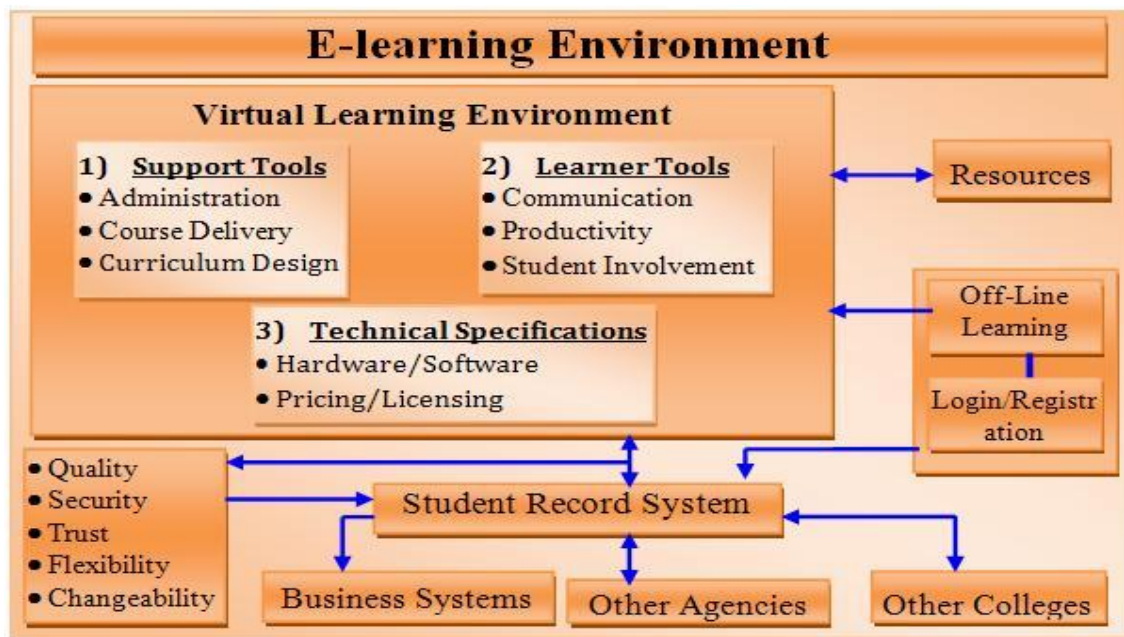


Figure 1: VLE as a Sub-System within the eLearning Environment [78]

Users are able to move in a continuous environment from one position to the next. The VLE functions as hyper-links to web pages and other Internet-deliverable services that can be viewed via the client embedded in a Croquet frame. Authenticated Users would be able to see

and interact with each other while their client caches data about a locale and renders the scene in real-time.

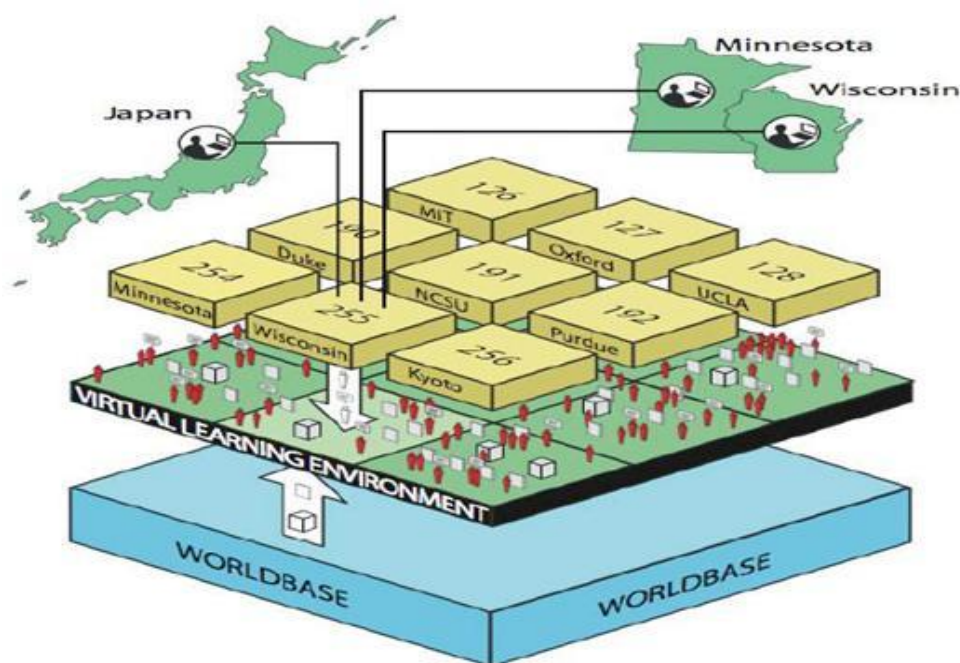


Figure 2: User-experience Architecture [76]

Interactivity servers are fundamentally long-lived Croquet clients, and admit connections from all clients for a VLE position to maintain transient or non-cacheable objects. Since the Interactivity server goes back up the list of users in their positions to new visitors, it has been accepted as the norm for authenticating a user's identity and supporting gathering credentials. The Worldbase servers distribute digitally signed, cacheable, authoritative data objects describing long-term persistent aspects of any position within the VLE, and a network locator for each position's Interactivity server.

2.3.5 Features and Tools of VLEs

This section gives the main features and components of VLEs and provides a set of guidelines for the purpose of conducting a meaningful comparative study of current VLEs systems and also for evaluating forthcoming systems.

VLEs enable educators to build resources fast and without the need to develop technical skills. They provide an integrated set of Internet tools, which allow the fast upload of materials and offer a consistent look and feel that can be customised by the user. VLE tools are criteria-based, and they enable developers to evaluate and select the most suitable VLE product. No single product can possibly meet all these criteria, and the most suitable within a specific context may not be perfect for interface, technical, functional, or cost reasons. Table 2 describes the tools and features that should be considered if a VLE is to facilitate a complete learning and teaching experience [41, 35, 37, 42, 43].

Table 2: Main Features and Tools of VLEs [78]

The Main Features and Tools of Virtual Learning Environments		
1. <u>Communication Tools</u> <ol style="list-style-type: none"> 1. Discussion Forums 2. File Exchange 3. Internal Email 4. Online Journal/Notes 5. Chat 6. Video Services 7. Whiteboard 	4. <u>Curriculum Design</u> <ol style="list-style-type: none"> 1. Accessibility Compliance 2. Course Templates 3. Curriculum Management 4. Customized Look and Feel 5. Instructional Standards Compliance 6. <u>Instructional Design Tools</u> 7. Content Sharing/Reuse 	7. <u>Hardware/Software</u> <ol style="list-style-type: none"> 1. Client Browser Required 2. Database Requirements 3. Server Software 4. UNIX Server 5. Windows Server
2. <u>Productivity Tools</u> <ol style="list-style-type: none"> 1. Bookmarks 2. Orientation/Help 3. Searching Within Course 4. Calendar/Progress Review 5. Work Offline/Synchronize 	5. <u>Course Delivery Tools</u> <ol style="list-style-type: none"> 1. Course Management 2. Instructor Helpdesk 3. Online Grading Tools 4. Student Tracking 5. Automated Testing and Scoring 	8. <u>Pricing/Licensing</u> <ol style="list-style-type: none"> 1. Company Profile 2. Costs 3. Open Source 4. Optional Extras 5. Software Version
3. <u>Student Involvement Tools</u> <ol style="list-style-type: none"> 1. Groupwork 2. Self-assessment 3. Student Community Building 4. Student Portfolios 	6. <u>Administration Tools</u> <ol style="list-style-type: none"> 1. Authentication 2. Course Authorization 3. Registration Integration 4. Hosted Services 	

2.3.5.1 Communication Tools

Communication tools are channels through which learners can make contact with each other and also with their educators. They make the VLE platforms work as one environment, and this facilitates connections between users. These tools have many options so that learners and educators can maintain their connections with each other anytime and anywhere, in an easy-to-use manner and without any cost. Examples of these tools are:

- **Discussion Forums:** Users can post their requests in these tools and discuss them with others. These are online tools that capture the exchanges of messages on a 24 hour basis. They are organised into categories, so that the messages and their responses are grouped together.
- **File Exchange:** This allows learners to upload, download and post Internet files, and share these files with instructors or other learners in online courses. It enables users to exchange many kinds of format and media such as doc, pdf etc.
- **Internal Email:** This is electronic mail that can be read and sent from inside online courses or otherwise external email of those on the course so that contacting course participants is facilitated.
- **Online Journal/Notes:** This enables learners to make comments in a personal or private journal. Learners can share personal journal entries with their educator or other learners but cannot share private journal entries.
- **Chat:** This tool is for conversation between users over the Internet and it involves exchanging messages. Some chat facilities allow the chats to be archived for later reference.
- **Video Services:** These tools enable tutors to either stream video from within the system, or else enable video conferencing, either between tutors and students or between students. They include tools for broadcasting video to those without a video input device. Some video services provide for two-method or multi-approach video conferencing, which may be point-to-point connections or mediated through a main server.
- **Whiteboard:** These tools include electronic versions of a dry-wipe board used by instructors and students in a virtual classroom and other synchronous services, such as application sharing, group browsing, and voice chat.

2.3.5.2 Productivity Tools

- **Bookmarks:** They allow learners to easily return to important pages within their course or outside their course on the Internet. Systems vary in allowing learners to store their bookmarks in a course folder and also allow bookmarks to be annotated.
- **Orientation/Help:** This is designed to help learners to learn how to use CMS. Typically, it is self-paced tutorials, user manuals, and email or telephone helpdesk support. It enables learners to make the best use of the software.
- **Searching within Course:** This helps learners to find the most suitable course that meets their search or request based on key words. It allows learners to locate course materials on the basis of word matching that go beyond the user's current browser page into other available courses.
- **Calendar/Progress Review:** This allows learners to document their plans for courses and associated activities. It allows learners to check their marks for activities and tests, as well as their overall progress through the course material. This tool works as an alarm for learners to alert them to the day for doing predetermined activities.
- **Work Offline/Synchronize:** This tool enables learners to work offline during their online course, and for their work to be coordinated and incorporated into the course the next time they log in.

2.3.5.3 Learner Involvement Tools

- **Groupwork:** This organizes a class into groups and provides groupwork space that allows the teacher to assign specific tasks or projects.
- **Self-assessment:** This tool enables learners to take practice or review tests online. It encourages learners to take responsibility for their own learning and to monitor their learning progress.
- **Learner Community Building:** This enables learners to generate study groups, clubs,

and/or collaborative teams, and can encourage the growth of learner friendships and partnerships.

- **Learner Portfolios:** This is an area where learners can showcase their work during a course, display their personal photos, and list demographic information.

2.3.5.4 Administration Tools

- **Authentication:** This tool gives learners the appropriate username and password and enables them to access the software and courses.
- **Course Authorization:** This is used to assign specific access privileges to course content and tools based on specific user roles.
- **Registration Integration:** This is used to add and drop learners from an online course. Administrators are responsible for this tool, but learners also use it when self-registration is available.
- **Hosted Services:** These indicate that the product provider offers CMS on a server at their location. Therefore, institutions do not have to provide any hardware. An important aspect of these tools is that the product provider takes responsibility for all technical support and maintenance of the server.

2.3.5.5 Course Delivery Tools

- **Course Management:** This tool enables teachers to control the progression of an online class through the course material. It is also used to make specific resources for a course.
- **Instructor Helpdesk:** This aids institute members in using the course management software. It contains a telephone contact (with the helpdesk of the product provider), documentation, and instructions. It also allows institute members to participate in online discussion forums with other institutes to share ideas and deepen knowledge.

- **Online Grading Tools:** These help teachers mark, provide feedback on learners' work, and manage grade books. They allow teachers to mark coursework online, store grades, and delegate the marking process to teaching assistants.

2.3.5.6 Curriculum Design

- **Learner Tracking:** This gives tutors the ability to track the usage of course materials by learners, and to perform additional analyses and reporting both of aggregate and individual usage. It contains statistical analyses of learner performance data and progress reports for individual learners on a course.
- **Automated Testing and Scoring:** This enables educators to create, administer, and score objective tests.
- **Accessibility Compliance:** This tool enables users with disabilities to access information online.
- **Curriculum Management:** This supports learners with customized programs or activities based on pre- requisites, prior work, or results of testing. It contains tools to manage multiple programs, to do skills/- competencies management, and to do certification management.
- **Instructional Standards Compliance:** This concerns how well a product conforms to standards for sharing instructional materials with other online learning systems and other factors that may affect the decision whether to switch from this product to another.
- **Instructional Design Tools:** These help educators create learning sequences, such as with lesson templates or wizards.
- **Content Sharing/Reuse:** This allows specific content created for one course to be conveniently shared with another tutor teaching a different course, perhaps even at a

different institution.

2.3.5.7 Hardware/Software

- **Client Browser Required:** This tool is the type and version number of the Web browser, such as Internet Explorer 3.0, Netscape 4.0, and works effectively with CMS.
- **Database Requirements:** These are the technical specifications for the database management software, such as Oracle and SQL, required by CMS.
- **Server Software:** Technical administrators use this to run the software on the server. It contains a diversity of software packages that support the infrastructure for CMS and allow technical administrators to effectively operate the server.
- **UNIX Server:** This means that CMS runs on a server using some variant of the Unix operating system.
- **Windows Server:** This means that CMS runs on a server that uses some version of the Microsoft Windows operating system.

2.3.5.8 Pricing/Licensing

- **Institutional Profile:** This tool contains public information about the company or organization which supports the course management software, such as founding date, owners, investors, location of the organization, number of employees, etc.
- **Costs:** These contain three kinds of cost, which are start-up costs, ongoing costs and technical support costs.
 - Start-up costs are the initial expenses involved in purchasing or using the software.
 - Ongoing costs are the expenses for maintaining the software after the initial set-up, such as upgrades and new versions of the software.
 - Technical support costs are the expenses of purchasing software assistance from the vendor.

- **Open Source:** This means the software is delivered with a source code and its license agreement gives the licensee the right to modify and redistribute the software.
- **Optional Extras:** These are features or extras added to the product that may or may not add to the cost. They are additional tools, products or services that are intended to work with the application.
- **Software Version:** This is the release number of a specific version of the software. It means that there are two or more active versions of the same product.

2.3.6 Comparative Study of VLE Products

It is important to make a comparison study between VLE products to select the suitable one and test it with our approach and also to explore their strengths and limitations. This comparative study is in two phases. The first phase is based on the features and capabilities of VLE tools, and the second is based on the technical aspects of the systems of VLEs.

VLEs have many features and capabilities such as forums, content management, quizzes with different kinds of questions, and a number of activity modules. Moodle has an additional number of contributed modules, including SCORM WebQuest and the Document Management System [44]. In this section, we have selected ten VLE products, including Moodle, to make comparisons between them, and our first comparison is based on the features and capabilities of VLE tools. The EduTools website [45] lists more than 80 VLE products and has performed a comparison of 42 VLE features and capabilities, as in Table 3.

Our comparison focuses on two kinds of products. The first is commercial e-learning systems and comprises Desire2Learn 8.1, ANGEL Learning 7.1, TeleTOP VLE, Blackboard (V6.2) and Scholar360. The second is OSS and comprises LON-CAPA, Sakai 2.3, dotLRN/OpenACS, ATutor 1.5.4 and Moodle 1.8. The comparison has two answers, Y or N. Y means the product has the feature and N means the product does not. Table 3 displays

information about the ten VLE software packages used in the first comparison. VLE Tools are criteria-based products that enable developers to evaluate and select the best VLE product. No single VLE product can possibly meet all these criteria and may not be the best for interface, technical, functional, or cost reasons. These criteria are described in Table 4 [26].

Table 3: General Information about the Selected Products [78]

No	Product	Developer name	Date	URL
1	LON-CAPA	Gerd Kortemeyer	Oct/2006	LON-CAPA Project
2	Desire2Learn 8.1	Desire2Learn Inc.	Oct/2006	Desire2Learn Inc.
3	ANGEL Learning 7.1	ANGEL Learning Inc	Oct/2006	Angel Learning
4	TeleTOP VLE	TeleTop B.V.	Oct/2006	TeleTop
5	Blackboard (V6.2)	BlackBoard	Nov/2006	Blackboard LSE
6	Sakai 2.3	Sakai 2.3	Nov/2006	Sakai
7	dotLRN/OpenACS	dotLRN	Jan/2007	dotlrn.org
8	Scholar360	Scholar360	Jan/2007	www.scholar360.com
9	ATutor 1.5.4	University of Toronto	April/2007	atutor.ca/atutor/index.php
10	Moodle 1.8	Moodlerooms	April/2007	www.Moodle.org

We begin but a description of the criteria over which we base our comparative study.

2.3.6.1 Criteria for Evaluating VLEs

It is important to evaluate the existing VLE systems in order to select the most suitable one that meets the future needs (requirements and specifications) of an academic institution. We outline four carefully selected criteria, which will be used as basis for our evaluation, as in Table 4 [27, 37].

2.3.6.2 Features and Capabilities

VLEs as an e-learning system have many features and capabilities. For simplicity, we have divided these features and capabilities into three phases, which are Learner Tools, Support Tools and Technical Tools, as illustrated in Table 5 which lists the features and capabilities of VLE tools that we have used in our comparison.

Learner Tools

This phase contains three kinds of tools: Communication Tools, Productivity Tools and Student Involvement Tools. Each Learner Tool has some features and capabilities as in Table 5. As we can see in Table 6, the comparison between the VLE products is based on Learner Tools. Four products are shown to be the best with almost the maximum number of features - 15 out of 16 features or capabilities of Learner Tools. These products are Moodle, Desire2Learn, ANGEL Learning Management Suite, and Sakai.

The LON-CAPA and dotLRN/OpenACS products have the minimum features and capabilities of Learner Tools, missing 5 out of 16 Learner Tools. TeleTOP VLE and The Blackboard Learning System have missed 2 out of 16 Learner Tools. Overall the best OSSs are Moodle, which missed 1 out of 16 Learner Tools.

Table 4: Criteria for Evaluating VLEs [78]

No	Criteria	Evaluation
1	Functionality	<ul style="list-style-type: none"> ➤ Does the system support a range of browsers/platforms? ➤ Can you import pre-existing materials? ➤ Can tutors add resources, including non-textual resources? ➤ Can the VLE be accessed from different computer platforms?
2	Management/assessment	<ul style="list-style-type: none"> ➤ Can the system store and view data about learners? ➤ Can the tutor create assignments/tests? ➤ Can the tutor assess assignments/test? ➤ Can students create/import content?
3	Flexibility/pedagogy	<ul style="list-style-type: none"> ➤ Can data be exported to University admin systems? ➤ Can assignments be submitted via the system? ➤ How easily can a course be modified after commencement? ➤ How easy is it to design a course?
4	Accessibility	<ul style="list-style-type: none"> ➤ Are accessibility design tools inbuilt? ➤ How easy is it for the tutor to follow accessibility guidelines? ➤ Are there special hardware/software requirements? ➤ Does the product address issues that relate to KSA?

Table 5: Summaries of the Features and Capabilities of VLE tools [78]

1) Learner Tools	2) Support Tools	3) Technical Specifications
1. Communication Tools <ul style="list-style-type: none"> • Discussion Forums • File Exchange / Internal Email • Online Journal/Notes • Real-time Chat • Video Services / Whiteboard 	1. Administration Tools <ul style="list-style-type: none"> • Authentication • Course Authorization • Registration Integration • Hosted Services 	1. Hardware/Software <ul style="list-style-type: none"> • Client Browser Required • Database Requirements • Server Software • UNIX Server • Windows Server
2. Productivity Tools <ul style="list-style-type: none"> • Bookmarks • Orientation / Help • Searching Within Course • Calendar / Progress Review • Work Offline/Synchronize 	2. Course Delivery Tools <ul style="list-style-type: none"> • Course Management • Instructor Helpdesk • Online Grading Tools • Student Tracking • Automated Testing and Scoring 	2. Pricing/Licensing <ul style="list-style-type: none"> • Company Profile • Costs • Open Source • Optional Extras • Software Version
3. Student Involvement Tools <ul style="list-style-type: none"> • Groupwork • Self-assessment • Student Community Building • Student Portfolios 	3. Curriculum Design <ul style="list-style-type: none"> • Accessibility Compliance • Course Templates • Curriculum Management • Customized Look and Feel • Instructional Standards Compliance • Instructional Design Tools • Content Sharing / Reuse 	

Table 6: The Comparison between Selected VLE Products based on Learner Tools [78]

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite	TeleTOP VLE	The Blackboard Learning System	Sakai 2.3	doLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools										
1. Learner Tools										
1.1. Communication Tools										
➤ Discussion Forums	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Discussion Management	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ File Exchange	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Internal Email	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Online Journal/Notes	N	Y	Y	N	Y	Y	N	Y	N	Y
➤ Real-time Chat	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Video Services	N	N	N	N	N	N	N	N	N	Y
➤ Whiteboard	N	Y	Y	Y	Y	Y	N	N	Y	Y
1.2. Productivity Tools										
➤ Bookmarks	Y	Y	Y	Y	N	Y	Y	Y	N	N
➤ Calendar / Progress review	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Orientation/Help	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
➤ Searching Within Course	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
➤ Work Offline/Synchronize	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
1.3. Student Involvement Tools										
➤ Groupwork	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
➤ Student Community Building	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Student Portfolios	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
Total Features	16	16	16	16	16	16	16	16	16	16
Total Available Features	11	15	15	14	14	15	11	12	12	15
Total Missing Features	5	1	1	2	2	1	5	4	4	1

Table 6 contains Learner Tools, which have many features and capabilities, and in order to understand what they mean, please refer to Chapter Three ‘Virtual Learning Environments as E-learning Systems’ where we explained them. We also list them in Table 5.

Support Tools

These tools contain three kinds of tools: Administration Tools, Course Delivery Tools, and Content Development Tools, and all of these tools have features and capabilities.

As we can see in Table 7, this comparison between the VLE products is based on Support Tools. In this phase, all products have all features and capabilities except Scholar360, TeleTOP Virtual Learning Environment and The Blackboard Learning System (V.7). This means that Moodle and the other remaining products are strong on Support Tools.

The Learner Tools in Table 7 have many features and capabilities, and to understand what they mean, please refer to the previous section. We have also listed them in Table 5.

Technical Specifications Tools

These tools contain two kinds of tools: Hardware/Software Tools and Pricing/Licensing; all kinds of Technical Specifications Tools have some features and capabilities, as in Table 8. The costs feature is different from other features because if the product has no cost, it means that product has an advantage and we will calculate it as Yes (Y). For example, in Table 8, Moodle has two N and we calculated N of cost as Y, so in the final score Moodle has missed just one feature.

Table 7: The Comparison between Selected VLE Products based on Support Tools [78]

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite	TeleTOP VLE	The Blackboard Learning System	Sakai 2.3	dotLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools										
2. Support Tools										
2.1. Administration Tools										
➤ Authentication	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Course Authorization	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Hosted Services	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Registration Integration	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
2.2. Course Delivery Tools										
➤ Test Types	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Automated Testing Management	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Automated Testing Support	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Course Management	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Online Grading Tools	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Student Tracking	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
2.3. Content Development Tools										
➤ Accessibility Compliance	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
➤ Content Sharing/Reuse	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
➤ Course Templates	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Customized Look and Feel	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Instructional Design	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Instructional Standards Compliance	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Total Features	16	16	16	16	16	16	16	16	16	16
Total Available Features	16	16	16	15	15	16	16	15	16	16
Total Missing Features	0	0	0	1	1	0	0	1	0	0

As we can see in Table 8, the comparison between the VLE products is based on Technical Specifications Tools. In this phase, the best product is ATutor 1.5.4, Moodle 1.8, Scholar360 and The Blackboard Learning System, which missed only 1 out of the 8 Technical Specifications Tools. The weakest product is LON-CAPA, which missed 5 out of the 8.

The Technical Specifications in Table 8 have many features and capabilities, and to understand what these features and capabilities mean, please refer to Chapter Three ‘Virtual Learning Environments as E-learning Systems’. We have also listed them in Table 5.

2.3.6.3 Summary of Comparison

From Table 9, we can see the final result of the comparison between the ten VLE products. The best product is Moodle 1.8, which has missed just 2 out of 40 features and capabilities, and the second products are Desire2Learn 8.1, ANGEL Learning Management Suite (7.1) and

Sakai 2.3 equally, which have missed 3 out of the 40. Also, Moodle is the best of the OSS products. The weakest product is LON-CAPA, which has missed 10 out of the 40.

We use the GraphPad Prism software to analyse, graph and present scientific data of VLE products because it is a powerful combination of basic biostatistics, curve fitting and scientific graphing in one comprehensive program.

Table 8: The Comparison between Selected VLE Products based on Technical Specifications Tools [78]

No	1	2	3	4	5	6	7	8	9	10	
Tools	Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite	TeleTOP VLE	The Blackboard Learning System	Sakai 2.3	dotLRN/Open/ACS	Scholar360	ATutor 1.5.4	Moodle 1.8
3. Technical Specifications											
3.1. Hardware/Software Tools											
➤ Client Browser Required	N	Y	Y	N	Y	Y	N	Y	Y	Y	
➤ Database Requirements	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	
➤ Unix Server	Y	N	N	Y	Y	Y	Y	Y	Y	Y	
➤ Windows Server	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	
3.2. Pricing/Licensing Tools											
➤ Company Profile	N	Y	Y	N	Y	Y	N	Y	Y	N	
➤ Costs	N	Y	Y	Y	Y	N	N	Y	N	Y	
➤ Open Source	Y	N	N	N	N	Y	Y	N	Y	Y	
➤ Optional Extras	N	Y	Y	Y	Y	N	N	Y	Y	Y	
Total Features	8	8	8	8	8	8	8	8	8	8	
Total Available Features	3	6	6	4	7	6	4	7	8	7	
Total Missing Features	5	2	2	4	1	2	4	1	1	1	

Table 9: The Final Result of the Comparison between Ten VLE Products [78]

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite (7.1)	TeleTOP Virtual Learning Environment	The Blackboard Learning System (V 7)	Sakai 2.3	dotLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools										
Total Features	40	40	40	40	40	40	40	40	40	40
Total Available Features	30	37	37	33	36	37	31	34	35	38
Total Missing Features	10	3	3	7	4	3	9	6	5	2

It has been dedicated to creating software exclusively for the international scientific community. More than one hundred scientists in over one hundred countries rely on Prism to analyse, graph and present their scientific data. Since 1984, created by scientists for scientists, Prism's intuitive programs have provided researchers worldwide with the tools they need to simplify data analysis, statistics and graphing [46]. Figure 3 shows the comparison between the ten products of VLE systems. The total features are 40 but no product has reached this number. In Figure 3, P1, P2 etc mean the VLE product as mentioned in Table 9 respectively.

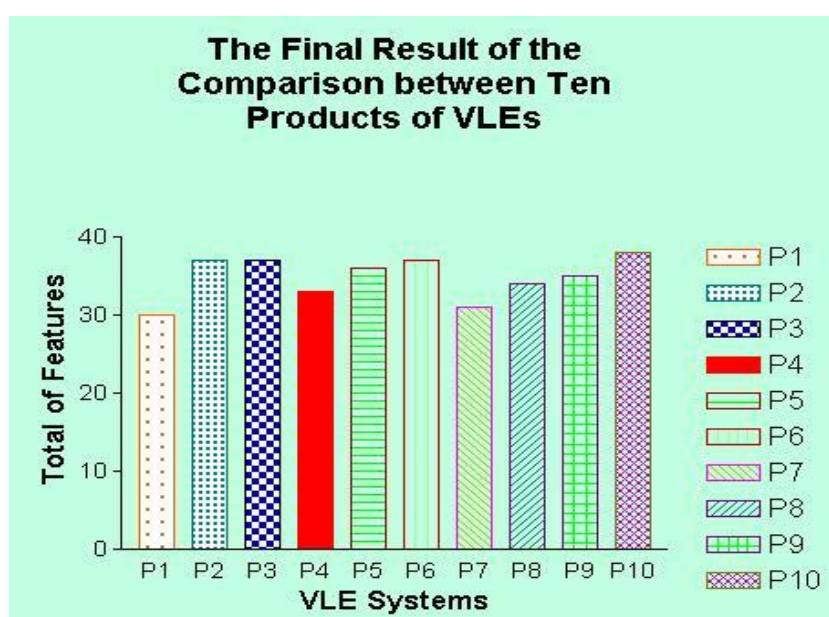


Figure 3: The Total Features of the Ten VLE Products [78]

As in Figure 3, the best VLE product is P10 (Moodle 1.8), which has 38 out of 40 features and capabilities, and the weakest is P1 (LON-CAPA), which has 30 out of the 40. P10 (Moodle) has 38 out of the 40 features and capabilities and is the number 1 out of the 10 VLE products. It is number 1 out of the OSS products, which itself has missed just 2 out of the 40 features and capabilities.

2.3.6.4 Technical Aspects

In this session, the comparison between the systems is based on technical categories. All VLE systems will be compared with the Moodle system as part of our study. We have selected four studies focusing on this kind of comparison.

- **First Study** - Table 10 reveals that ATutor, while strong in features and usability, has serious architectural limitations, and although some features in ATutor warrant further investigation, it may be that candidates will opt for Moodle. ILIAS, while promising, has a complex architecture with tight coupling that is hard to work with and debug.

Table 10: Comparison based on focusing on the technical aspects of the VLE systems [78]

Category \ Product	ATutor	ILIAS	Moodle
Architecture	Weak	Complex	Good
Implementation	Weak	Complex	Good
Interoperability	Bad	Good	Average
Cost of ownership	Medium	High	Low
Strength of the community	Low	Medium	High
Licensing	GPL	GPL	GPL
Internationalization	Weak	Average	Good
Accessibility	Excellent	Bad	Average
Document transformation	No	Average	No

Moodle has a good architecture, implementation, inter-operability, and internationalization, and also has the strength of the community. It is free and its accessibility is average. On the other hand, it has limitations, as mentioned in [47].

- **Second Study** - Table 11 shows the comparison between 4 products of VLE systems. The comparison is based on categories as [48] determined. This study has proved that Moodle outperforms all other systems and scored 4.467 out of 5. In contrast, Boddington gained the lowest score, at 2.439.

Table 11: Comparison based on focusing on some Features and Categories of VLEs [78]

Category \ Product	Moodle	Sakai	ATutor	Boddington
Functionality	1.25	.75	.25	.25
Usability	.8	.8	.6	.65
Documentation	.645	.465	.54	.54
Community	.6	.384	.24	.288
Security	.42	.34	.28	.42
Support	.4	.15	.35	.15
Adoption	.352	.336	.208	.336
Total Score (out of 5)	4.467	3.225	2.468	2.439

- **Third Study** - in [40], the study reports that the result of the evaluation shows that Moodle has the best rating in the adaptation category; it can be seen in Table 12 as the best system concerning adaptation issues. Concerning the other platforms, ILIAS obtained the best values in the categories for technical aspects, administration, and course management.

Table 12: Results of the Adaptation Category [78]

Feature \ Product	Adaptability	Personalization	Extensibility	Adaptively	Ranking
1 ATutor		#	#		3
2 Dokeos		0	*	+	2
3 dotLRN	+	+	*	0	2
4 ILIAS	+	#	*	0	2
5 LON-CAPA	+	#	#		2
6 Moodle	#	+	*		1
7 OpenUSS	#	#	#	0	2
8 Sakai	0	0	*	0	3
9 Spaghettilearning	+	#	+	0	3

- **Fourth Study** - in [49], the study reports the percentage of universities that are developing or using e-learning frameworks. The result of e-learning survey conducted the use of web-based learning management systems for higher education. As we can see in Figure 4, Moodle is the best product that has 34.55% and the second product is WebCT/Blackboard that has 27.27%. The weakest product is Sakai and dotLRN that have same percentage 10.91%.

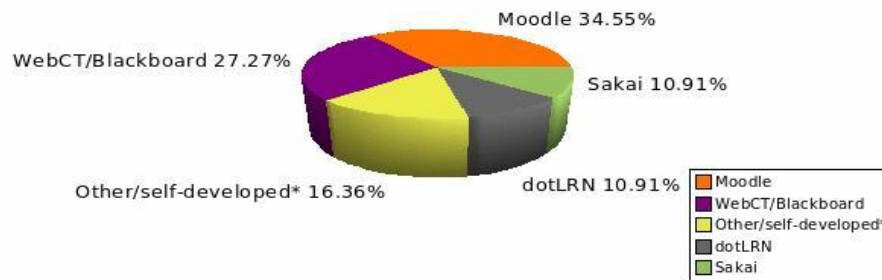


Figure 4: Chart of use of web-based e-learning systems in Universities [78]

2.3.7 Next Generation of VLEs

The increasing dependence on eLearning systems for learning and education through technology (today the Internet, tomorrow Digital Television) will promote further challenges to the universities in their traditional market- place. By investing controlled resources into the future development of VLEs, institutions will be best placed to deliver learning into the next decade, as in Figure 5. Therefore, Web services (together with agent technologies) will play an important role with this future to enhance VLE products [50, 35, 51, 52, 53, 54].

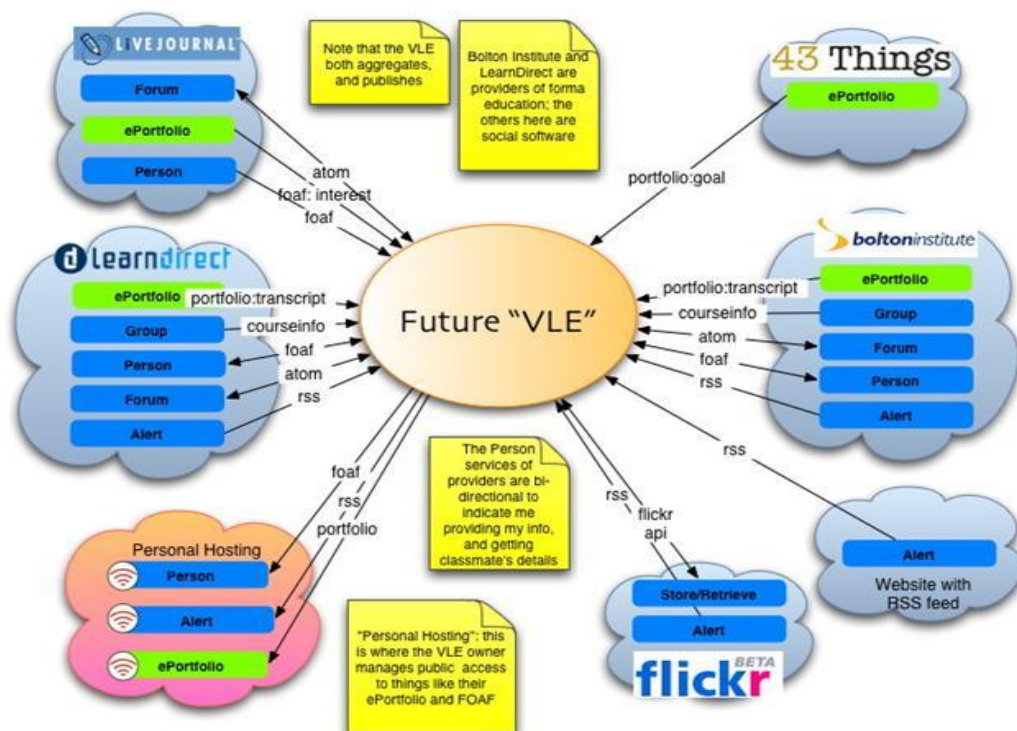


Figure 5: The Future of VLEs - The Visual Version [77]

One useful way of thinking about how learning environments might develop is to look at current trends in Internet technologies, and at a visual representation, as in Figure 5. VLEs should be modular to make future development of applications more easy and cost-effective. At the same time, the development of future educator training should be based upon the same principals as are foreseen for future educational and training practices: general monitoring, reflection and research of emergent practices (especially of successfully incorporating technology), and individual and flexible delivery of courses and other services [50, 55].

The future of VLEs will develop **high-quality pedagogical standards and methodologies to support the users (educators and learners) online in relation to state-of-the-art pedagogical and didactic methods and content matters in the light of future teaching**. In addition, OSS will enable the future of VLEs to make integration of generative tools, such as visualising, mind-mapping, collaborative design tools and digital libraries, all with archive capabilities. However, it is necessary to take the needs of current educators and learners into account before more extensive innovations can be incorporated [50, 56].

Currently, advances in academic computing, including the ubiquitous deployment of high-connectivity, have finally made it feasible for educators to shape the new media in transformative ways. A persistent, unified, massively multi-user and self-organizing virtual environment for learning, which capitalises on the rich collaborative capabilities of open-source technologies such as Croquet, is what is needed to take eLearning to the next plateau, enabling online ‘interactivity’ in a constructivist sense e.g. ‘interactivity’ that is synonymous with vital, self-organizing communities of practice [38].

Dynamic OSS provides social interaction, and thus facilitates online communities (virtual villages). More and more online communities will come into view around the globe.

Educators should not hesitate to create and maintain their Web presence to help their learners to build computer and information literacy [57].

2.4 Mobile Learning (mLearning)

Another important development in VLEs is that which utilizes current advances in mobile devices and wireless communication technologies. This will take us one step closer to achieve the overall goal of eLearning, namely the provision of anywhere, anytime education and training. This development is often known as mLearning, allowing learners to use variety of mobile device. Mobile technologies offer new opportunities for electronic and distance learning and enable provision of educational electronic services and teaching content anywhere and anytime. Recently, a huge amount of research projects in the field of mLearning are under development. In [79], the online distance learning system JELD (Java Environment for Learning Design), designed to implement different kinds of approach to learning, is presented. The system is a J2EE [80] based architecture that allows users to define teaching strategies, pedagogical approaches and educational goals. Moreover, JELD can adapt learning activities to the client device; in particular modules for PCs, personal digital assistants (PDAs) and mobile telephones. mCLT, an innovative mobile platform for computer-supported collaborative learning, in which traditional methodologies of collaboration have been improved, based on 3rd-generation mobile telephones, has been introduced in [81]. Students can collect and share live data immediately, anywhere and at any time. In this way they can participate active in the education process. Application of ambient technology as a tool which researchers can use when testing theories of learning with mobile devices, and developers can use when evaluating mobile learning systems is described in [82]. Extension of a state-of-the-art e-learning system, known as Intelligent Web Teacher (IWT), to support multimodal mobile access is presented in [83]. The extended platform can offer

customised e-learning experiences depending on the type and capabilities of the user's mobile device. Development and prototypical implementation of a context-aware team interaction support system is presented in [84]. This offers up-to-the-moment information about the location, state and activities of learning teams, thus helping to build up awareness of the surroundings in which team learning processes can take place. In [85], questions are introduced that are important when designing adaptive mobile learning systems. Adaptivity is defined as one form of adaptation. An important issue is to weigh the usefulness or limitations of adaptivity in learning systems. Other essential questions include issues of choice between different adaptive techniques and choosing the foundation for them – in this case, user modelling; and the challenges of gathering information on learners. In the HISS (Hospital Information System for Students) project students of medicine were trained to use handheld devices connected through a wireless local area network to record patients' data [86]. The Mobile Technologies for Ad-hoc Learning project is a joint initiative of pedagogical and cognitive science, involving technological experts, educators and psychologists, to research the possibilities of using mobile platforms with internet access for educational purposes at school level [87]. The project designs, develops, tests and evaluates a handheld learning environment based on emerging technology that facilitates on-site learning. Collaboration between a web-based multi-user environment that enhances a student's learning experience through the use of Virtual Reality and multimedia and an agent-based intelligent tutoring system is described in [88]. In [89], a conceptual model called MoDCA (Mobile Device Collaboration and Assessment) is proposed. This model defines how the learning objects should be presented and also how the pedagogy should be tailored across the different wireless platforms in such a manner as to promote effective learning and assessment for individuals with diverse learning needs which are continuously changing and adapting. The MOBIlearn project aims to support a wide range of services and applications for learners

using mobile computing devices such as phones, personal digital assistants (PDAs) and laptops [90]. An object-oriented, feature-based architecture for a context-awareness subsystem to be implemented within the MOBIlearn project is presented. In [91], a learning organiser for handheld computers is presented. The aim was to investigate whether an integrated set of learning tools would be useful, which tools would be adopted and the contexts in which the tools would be used. A mLearning project that was a 3-year pan-European collaborative research and development programme supported by the European Union (EU) is presented in [92]. The consortium is a partnership of organisations combining skills in pedagogy and technology. The project has developed prototype products and innovative approaches designed to support learning using handheld devices such as mobile phones and palmtop computers or pocket computers. A key objective is to engage with and motivate young adults who are not taking part in education or training, including those who are unemployed, homeless or disadvantaged.

In the preceding, but related with the topic of this thesis research, a own general architecture for mLearning had been proposed. Generally, the architecture was proposed to operate across a single University Campus [53, 58, 59, 60]. It facilitates efficient user access to services and resources via InfoStations deployed at key locations within a University Campus. The architecture has a 3-tier structure involving the user mobile devices, the InfoStations, and the InfoStation Center [61, 62, 63]. User access to the mLearning services is facilitated by utilizing existing Bluetooth and WiFi wireless standards.

The first tier of the architecture encompasses user mobile devices, equipped with intelligent agents that act as Personal Assistants to the users. The second tier consists of InfoStations, deployed within the University Campus, facilitating the user mobile access to services through wireless connections. The third tier is the InfoStation Center - the core of the overall

architecture. It is concerned with the control of the InfoStations, and the overall updating and synchronization of synchronization of information across the system. Various electronic services can be deployed in the InfoStation in the InfoStation network (

Figure 6) [64, 65]. In addition, different scenarios for execution of services over the InforStation network had been specified [66, 54]. In this way, we may view such an architecture as a multi-context system, in which each InfoStation defines a local context within which a number of mobile devices are located which may move from one context to another. Within each context, there is a number of services that can be provided and the locations of services are transparent to the user. If a user request can not be performed by its context InfoStation, then it is redirected to the InfoStation Center.

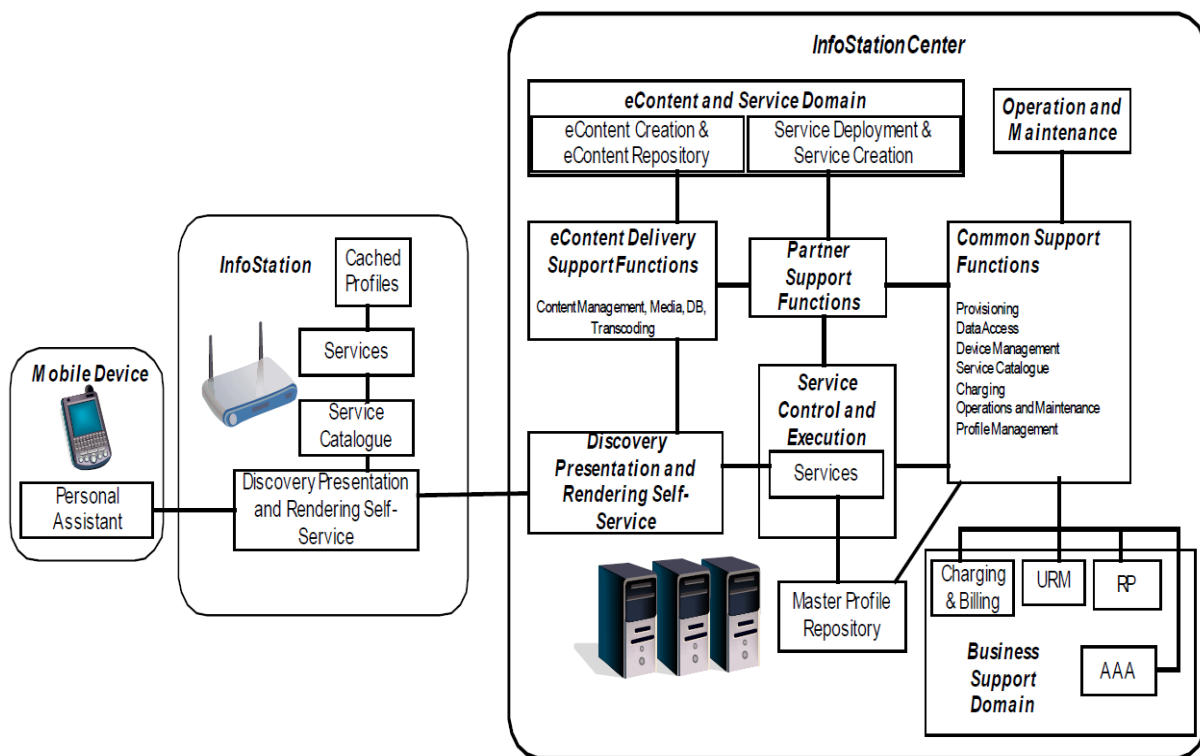


Figure 6: Services Deployment in the InfoStation Network

2.5 Digital Libraries

Generally, the classical notion of a “library” is that it should be a commons for bringing together knowledge, and learners within an environment that will ultimately enhance both the knowledge itself and learner’s education. Such a sentiment should be maintained in the context of Digital Libraries.

The extensive deployment of digital libraries over the last years is hardly surprising [67]. The term Digital Library (DL) has been widely used recently within academic institutions to denote an electronic repository within which academic’s research outputs can be easily stored and retrieved . Recently, various working digital libraries are academic and provide free access to their content. Some examples are following:

- EULER - a portal for mathematics publications [68]
- Alexandria Digital Library - a distributed digital library with collections of georeferenced materials [69].
- PERSIVAL – digital library for medical information [70].
- VARIATIONS – digital library in the field of music [71].
- Perseus Digital Library project – aims to give more people access to source material in the humanities [72].
- Google Scholar - provides access to peer-reviewed papers, theses, books, abstracts and other scholarly literature [73].

The simple rationale is that it will increase visibility of the staff as, with the popularity of the Internet, it enhances and facilitates the accessibility of their research outputs by their peers world-wide. It has certain design rationale/principles and there are specific concerns/constraints with their implementation. Whilst it contributes, somewhat, to the

training and education of staff and, perhaps, learners in general, it has limited use within the specific requirements of eLearning systems.

Along with modern information technology, the digital resources play a fundamental role in eLearning. Lecture courses, examinations, exercises, different profiles, etc. are the manifold information ingredients that have to be managed and made accessible in an eLearning system [74]. A digital library provides powerful and efficient support for content management, different kinds of metadata, as well as services for effective content search, access, annotation, filtering, and dissemination [75].

2.6 Summary

In this chapter we have presented a detailed study of eLearning systems and their various platforms and provided a comparative study of current eLearning technologies. This has included a set of generic criteria which act as a benchmark for any future eLearning system. We have also established architectural design and technical specification of future eLearning systems and in particular of their Digital Library component.

Chapter 3

Distributed eLearning Center

Objectives:

- To present the architecture of the Distributed eLearning Center (DeLC).
- To provide the basic models of DeLC.
- To demonstrate the current state of the DeLC.
- To show some previous contributions of the author related to the development of DeLC.

3.1 Introduction

In this chapter we give overview of an architecture that can be used for development of next generation of eLearning environment and system. This architecture is known as Distributed eLearning Center (DeLC). DeLC enjoys the following properties:

- Fully distributed.
- Integrates service-oriented architectures with agent technology.
- Highly adaptable.
- “Context” is central to its realization.
- Provably correct.
- Compositional.

Central to our architecture is that a node is our unit of computation (known as E-Learning node) which can have a purely service-oriented architecture, agent architecture or a mixed architecture.

The architecture and its properties are described in full within 15 published papers (11 published before April 2008) of which one seminal paper is included in the chapter and the rest are given as references.

3.2 Modeling of DeLC

3.2.1 Conceptual Model

The conceptual model of DeLC [93, 98, 99] consists of five components (Figure 7) the main two of which are:

- **eContent generator** – used by educators and instructors to create an electronic content (eContent) needed for the learning process.

- **eContent interpreter** – used by learners for the interpretation of the created eContent.

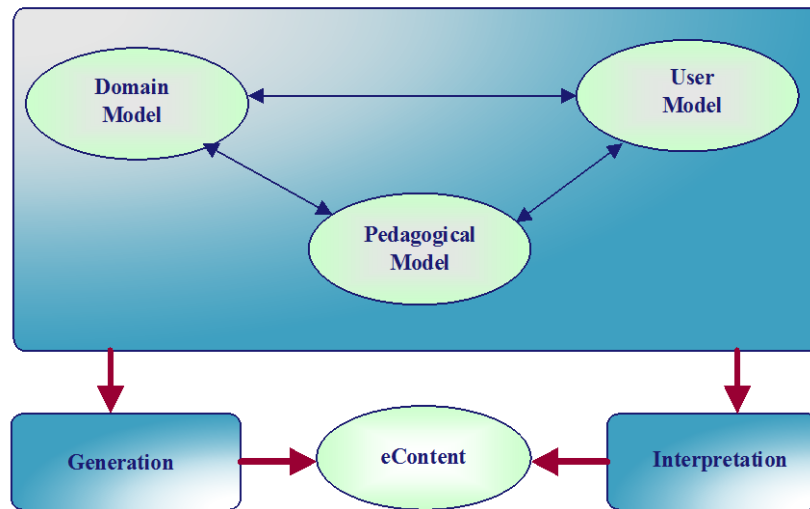


Figure 7: The DeLC Conceptual Model

The creation and interpretation of the eContent depends on some important circumstances and limitations, which could be generalized in the following three models [94]:

- **Domain model** – represents the knowledge of a particular learning area/field; usually contains basic (atomic) concepts and different semantic links between them presented as ontologies.
- **Pedagogical model** - contains information about possible didactical solutions used by the Learning Management System (LMS), e.g., to explain how to study the teaching material provided as eContent.
- **User model** – usually stores specific information for individual users, which could be used for personalization of their eLearning process. Depending on the chosen approach this information could represent the preliminary degree of learners' preparation, preferences, affinity to a particular course/module, type of mobile devices in use, ID data etc. The following main approaches could be used for the implementation of this model:

- *Stereotypes* – the simplest way of modeling by user categorization in groups (categories). For each group the system supports corresponding description called *stereotype*. Each user is categorized to one or more stereotypes (e.g., beginner, middle level, advanced, expert). Each stereotype has predefined properties inherited by corresponding users' group.
- *Overlays* – the user knowledge is considered as a subset of the expert knowledge. This way, it is possible to use the same presentation for the domain model and the user model.
- *Mixed* – a combination of the previous two approaches.

3.2.2 DeLC Infrastructure

The Distributed eLearning Center aims to provide a distance e-Learning and e-Teaching facility available at any place and at any time to individuals and groups of students/educators both in on-line (synchronous mode) and off-line (asynchronous mode). DeLC infrastructure is modeled as a network structure (Figure 8), which consists of separate nodes, called **eLearning Nodes** (eLNs). The nodes model real units (laboratories, departments, faculties, colleges, and universities), which offer a complete or partial educational cycle. In this way the separate eLNs host various education services and teaching content.

Remote eService activation and integration is possible only by means of previously defined virtual structures, named **clusters**. In the network model we can easily create new structures, reorganize or remove existing structures (the reorganization is done on a virtual level, it does not affect the real organization). For example, the reorganization of an existing virtual structure can be made not by removing a node but by denying the access to the corresponding eServices offered by it. The reorganization does not disturb the function of other nodes (as

nodes are autonomous self-sufficient educational units providing one or more integral educational services).

Towards the access to education services and teaching content an eLN can be either **fixed** or **mobile**.

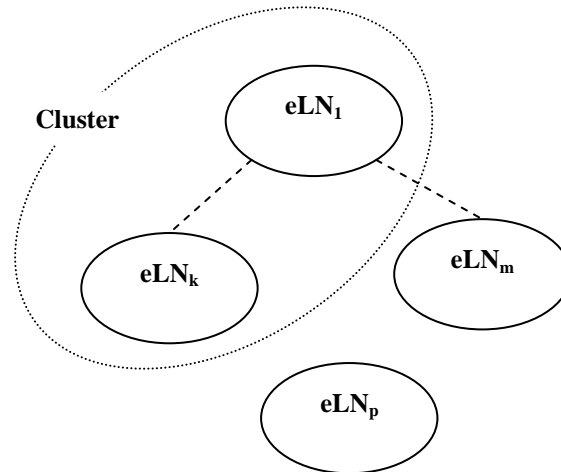


Figure 8: DeLC Network Model

3.2.3 Enhanced DeLC Network

In order to support the development of a generic mobile eLearning Node, the DeLC network was enhanced with an InfoStation-based communication environment with distributed control [97, 96, 103]. The InfoStation paradigm is an extension of the wireless Internet, where mobile clients interact directly with Web service providers (i.e., InfoStations) [108]. The users request services (by their mobile devices) from the nearest InfoStation utilizing Bluetooth or WiFi wireless communication. Of course in future, the incorporation of technology such as WiMAX will greatly aid the deployment of this architecture and the delivery of services to a much wider community. An InfoStation network consists of the following three tiers as shown in Figure 9:

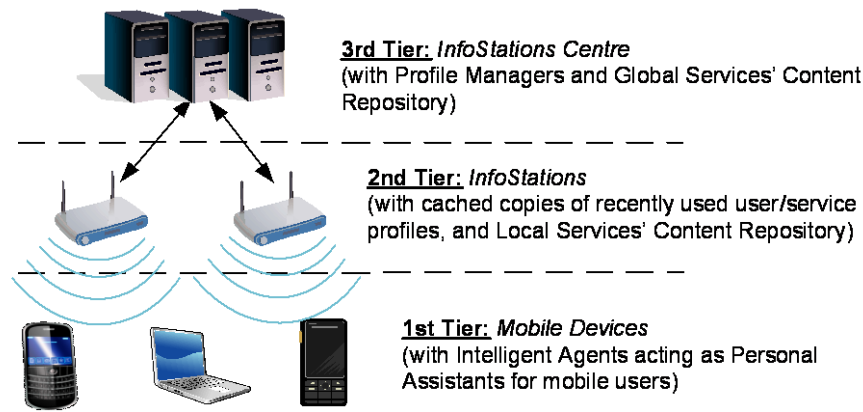


Figure 9: The 3-tier InfoStation-based network architecture

- *1st tier* – encompassing the user mobile devices (cell phones, laptops, PDAs), equipped with intelligent agents acting as Personal Assistants to users. The Personal Assistant gathers information about the operating environment onboard the mobile device, as well as soliciting information about the user. Supplied with this information, the InfoStation can make better decisions on applicable services and content to deliver to the Personal Assistant.
- *2nd tier* – consisting of InfoStations, satisfying the users' requests for services through Bluetooth and/or WiFi wireless mobile connections. The InfoStations maintain connections with mobile devices, create and manage user sessions, provide interface to global services, offered by the InfoStation Center, and host local services. Supporting services within specific localised regions throughout the University campus, we can enrich the service users experience within these localities. A prime example of how this type of local service can enrich a learners experience, is the deployment of library-based services [95]. Within the library domain, library users experience can be greatly enhanced through the facilitation of services offering resource location capabilities or indeed account notifications. The division of global and local services allows for a reduction of the workload placed on the InfoStation Center. In the original, the InfoStations operated only as mediators between the user mobile devices and a centre,

on which a variety of electronic services are deployed and executed [108]. The InfoStations within this architecture do not only occupy the role of mediators, they also act as the primary service providing nodes.

- *3rd tier* – this is the InfoStation Center, concerned with controlling the InfoStations, and overall updating and synchronisation of information across the system. The InfoStation Center also acts as the host for global services.

The DeLC communication infrastructure, supporting m-Learning and m-Teaching [101], is depicted in Figure 10.

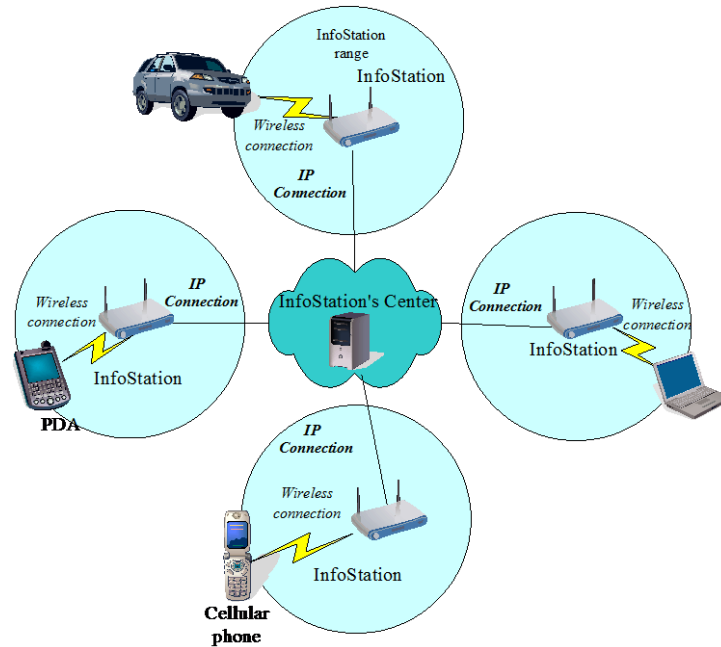


Figure 10: DeLC communication infrastructure supporting m-Learning and m-Teaching

3.2.4 DeLC Services

In DeLC, the concept of service control is developed around a service classification model presented in Figure 11. This model serves the implementation of the service management as part of the system run-time module. The service model is open for extensions including new values of already specified dimensions or entirely new dimensions. In order to separate the

architecture from services' implementation, an independent service classification model was proposed. The model consists of two main components:

- *Service characteristic space (or meta-model).*
- *Subject models.*

The meta-model classifies services within an **n**-dimensional discrete space [93, 106].

Currently, the next aspects (dimensions) are defined:

- *Functionality* – a classifier of services according to their content.
- *Target Group* - a classifier of services according to their potential users.
- *Invocation* - a classifier of services according to the activation option provided in the infrastructure (i.e., local, remote, and back-end services).
- *Mobility* – specifies whether the service is a mobile one or a stationary one.
- *Standard-conformance* – whether the service is conformant to an eLearning standard (e.g., SCORM 2004).

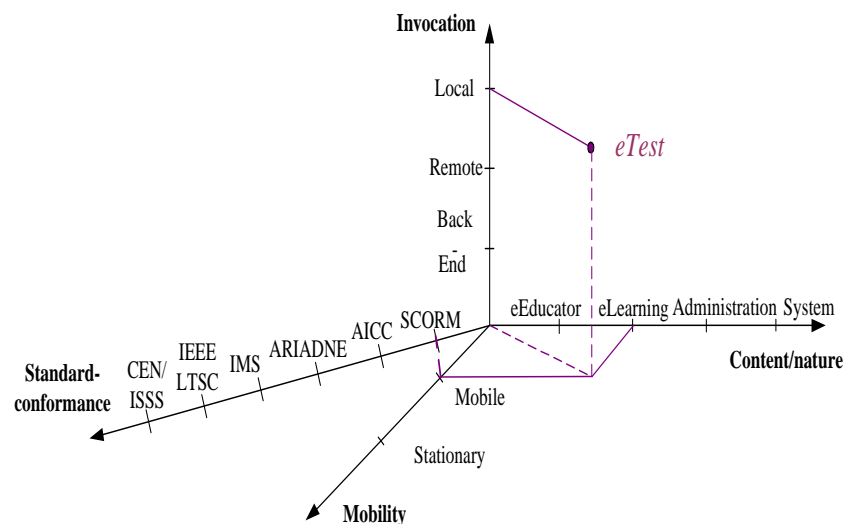


Figure 11: The service classification model

The subject models specify the functionality of the offered education services and are presented by help of appropriate modeling means (e.g., UML diagrams [104, 100]). The subject model of *Intelligent Message Notification Service* is shown as an example [105, 107]:

- **Service name:** '*Intelligent Message Notification*'
- **Short description:** This service is needed to broadcast MMS/SMS messages to a group of users, e.g., message notification sent by lecturer to a class of students about canceling/postponing the lecture, message sent by librarian announcing a library demonstration. The lecturer/librarian types and sends the message on his/her mobile device via available Ultra Wide Band, Bluetooth or WLAN connection to the nearest InfoStation. The InfoStation then forwards this message to the University's message gateway, which is deployed as a module in the InfoStations' Center and functions as an *intelligent message redirector* deciding what is the most appropriate, quickest and cheapest way of delivering this message to each student in the class according to his/her current individual location (and device in possession) specified in the student profile. All registered DeLC users (lecturers, librarians and students) have profiles in the DeLC containing (among other things) information about the best way of forwarding urgent messages to them at any particular moment, e.g. by MMS/SMS, email, fax, voice mail or otherwise.
- **Communication infrastructure:** Figure 12.

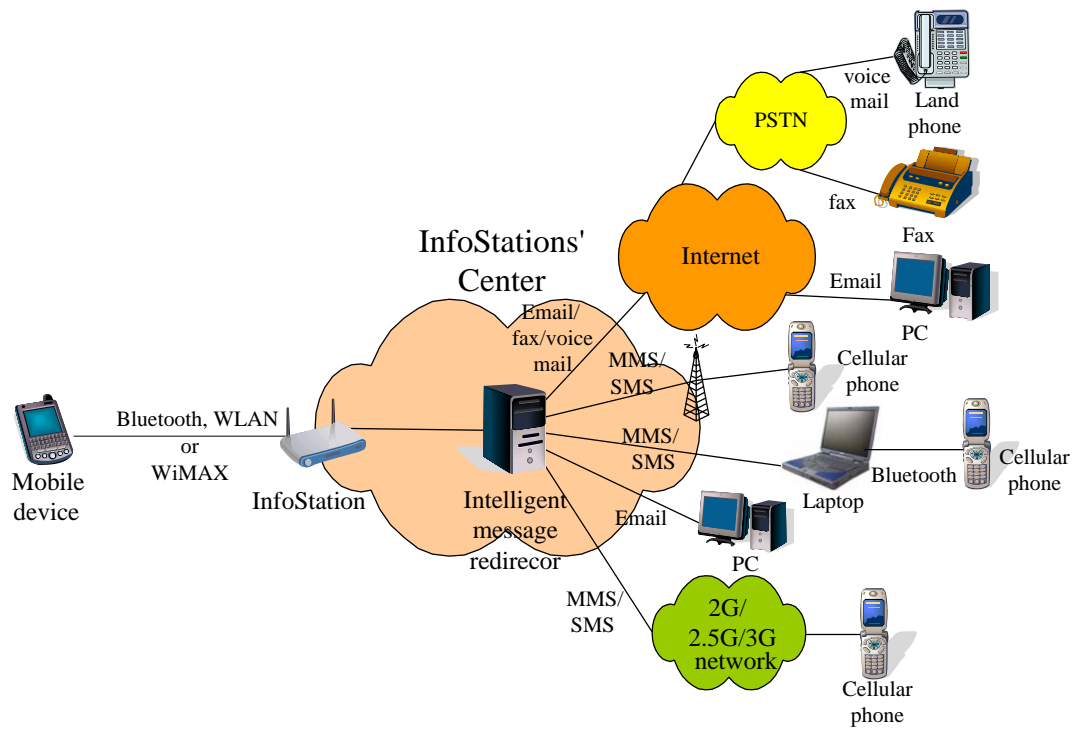


Figure 12: Communication infrastructure supporting the mobile service 'Intelligent Message Notification'

- **Service provision:** The Figure 13 illustrates sample interactions between entities involved in the facilitation of this service (i.e., InfoStation Center, InfoStation, Mobile Device, User). A mobile user (e.g. Lecturer, Librarian or Student) is within range of an InfoStation, s/he goes through the normal AAA procedure and selects a service for use, in this case the *Intelligent Message Notification Service*.

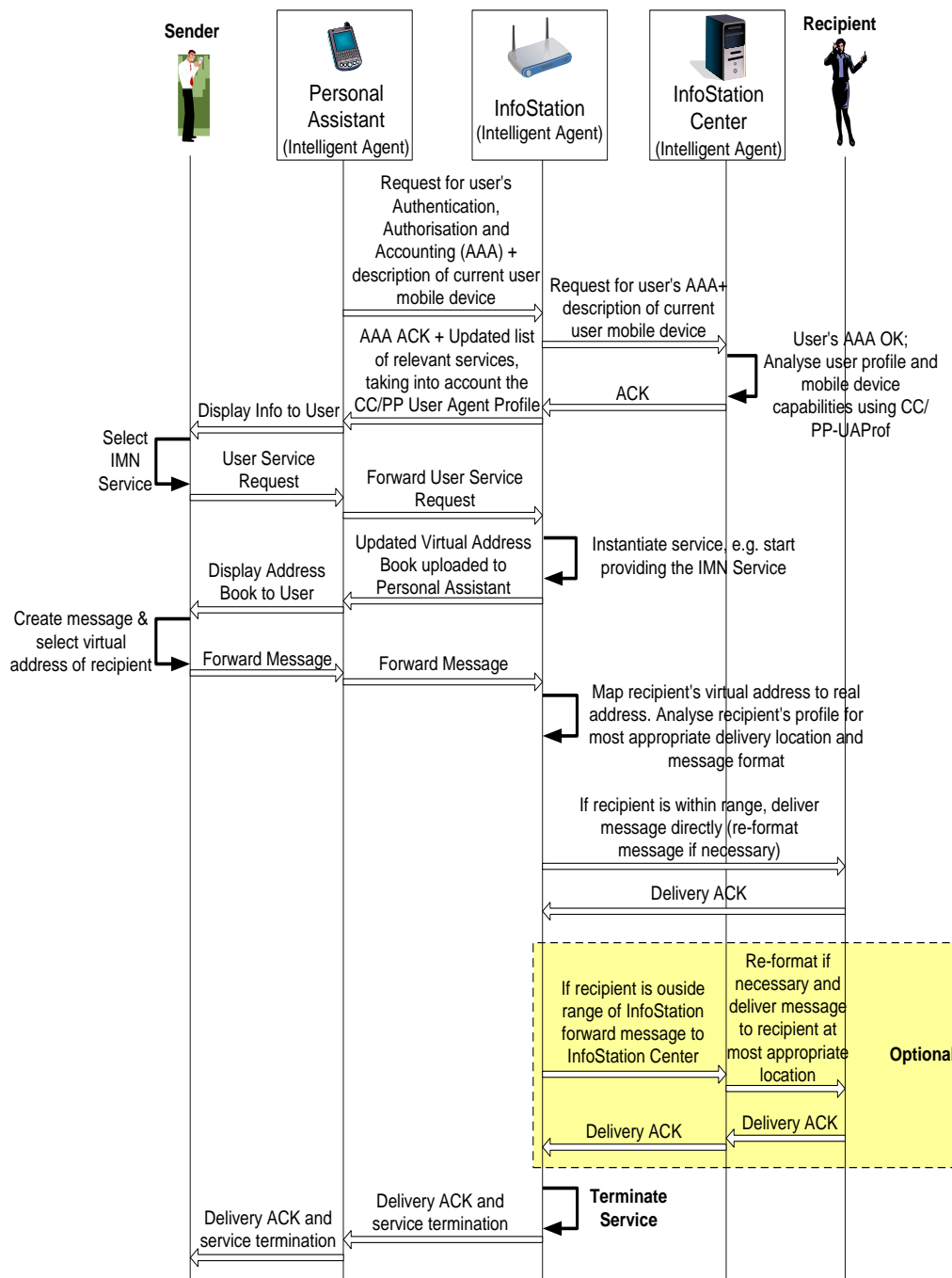


Figure 13: Intelligent Message Notification Service provision

An intelligent agent (Personal Assistant) on the mobile device forwards this user service request to the InfoStation, which instantiates the service. The InfoStation updates the user mobile device's Virtual Address Book (VAB), which lists all currently available registered users. This is constantly updated and maintained by the system. From the VAB, the user selects the desired destination of the message and

creates it in the form of SMS, MMS, email etc., which is then forwarded by the Personal Assistant to the nearest InfoStation. The InfoStation analyses the recipients' profiles and makes a decision on the most reliable, appropriate, quickest and cheapest way of delivering the message to each of the recipients. This decision takes into account factors such as current recipient's location, mobile device, and other most recent information contained within his/her profile. If the recipient is within range of the InfoStation, the InfoStation forwards the message directly to him/her (and optionally reformats the message beforehand if necessary e.g. when an SMS/MMS is to be delivered to an e-mail address and vice versa). If however the message recipient is outside the range, the InfoStation forwards the message onto the InfoStation Center, which in turn delivers the message to its destination (i.e. most appropriate recipient's location). Once the message has been delivered to the relevant user(s), the InfoStation Center forwards a delivery acknowledgment to the InfoStation, which then forwards it to the Personal Assistant, installed on the user mobile device, and terminates the service. The assistant in turn displays this confirmation to the user.

Besides the classification, each service is considered as consisting of two parts [102]:

- *Service's user part (SUP)* – interface of the service, used for its control during initialization (searching, discovering, identification...) and finalization (in case of device change). SUP is deployed in the mobile devices and the InfoStations.
- *Service's server part (SSP)* – deployed and executed by the only non-changeable component - the InfoStation Center. SSP contains the realization of the functionality provided by a service.

This service description could be mapped to different models or protocols for service control. Some services directly use knowledge presented in the *domain model*. From an architectural point of view the possibility for dynamic switch/change (during the run-time) from one learning area to another is very important. Today's approaches for the implementation/realization of eLearning models usually use *ontologies* [211]. Besides this, depending on their nature, some services could be parameterized by retrieving the parameter values from the *user model* and the *pedagogical model*. This increases the possibilities for architectural support for one of the main characteristics of the eLearning systems, namely service **personalization**. The possibility for parameterization, parameter types, their allowable values, the time needed to assign these values (during service development or dynamically during the run-time) are defined in the corresponding subject models.

3.3 The Development of DeLC

The actual implementation of DeLC incorporates internal and external eLearning nodes. The internal eLearning nodes are used directly in the real education in the university, while the external eLearning are developed and in use out of the university. Commonly, the next eLearning nodes are developed (Figure 14):

- Mobile eLearning node – an internal node presented in Chapter 4.
- Education portal – an internal node presented in Chapter 5.
- Agent Village – an internal node presented in Chapter 6.
- Education cluster – a cluster, incorporating the Education portal and the Agent Village, presented in Chapter 6.
- Distributed eTesting Center (DeTC) – an external node specialized for testing in the Faculty of Economics and Social Sciences (University of Plovdiv) and in the

University of Food Technologies, Plovdiv (in [109] supervised from the author of this thesis).

- Education portal for secondary school - an external node supporting eLearning in secondary school (in [110] supervised from the author of this thesis).

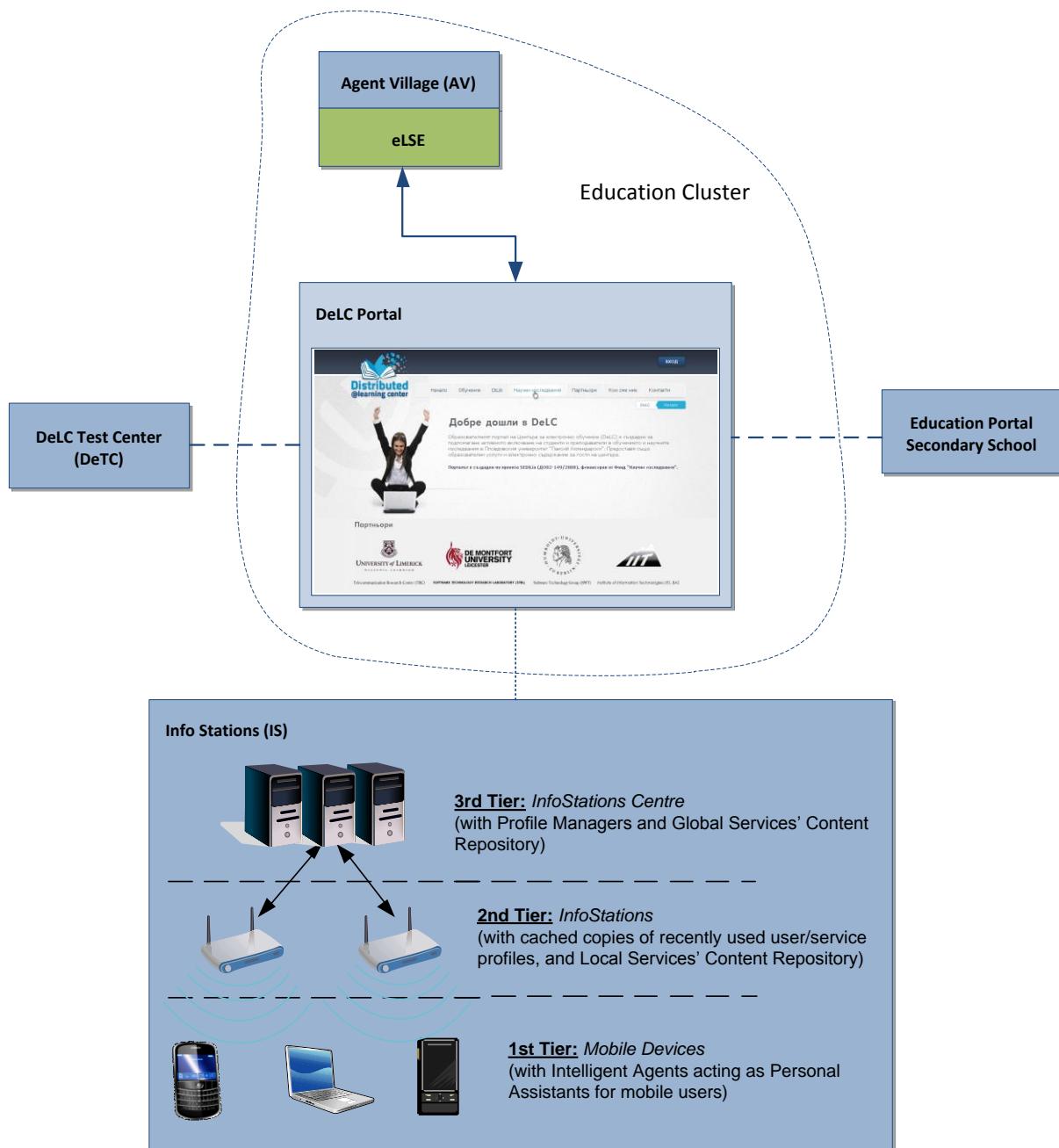


Figure 14: DeLC infrastructure

3.4 Summary

A general overview of the DeLC architecture is given in this chapter. Besides, the modeling of the basic architecture components is presented in some detail. The current implementation of the proposed architecture is discussed as well.

Chapter 4

Mobile eLearning Node

Objectives:

- To present the agent- and service-oriented architecture of the mobile eLearning Node of DeLC.
- To provide the development approach of the node.
- To present the system architecture of the node.
- To discuss the context-aware service provision in DeLC.
- To present the scenarios level, middleware and services level of the architecture of the mobile eLearning node.
- To discuss the *Base Architecture* and *Resource Deficit* iterations.
- To provide the simulation environment using for testing of the middleware.
- To demonstrate the operation of the middleware with the *mTest* service.

4.1 Introduction

In this chapter we introduce the concept of *Mobile eLearning Node (InfoStations)*. The Mobile eLearning Node has the following characteristics: agility, decision making, efficient cooperation, and context-aware. The functionality of mobile eLearning Node is to be aware of its context, gather required learning material and adapted to the learner request. The node uses a three-level communication network, called InfoStations network, supporting mobile service provision. The following four contributions, related to this problem, are presented briefly in this chapter:

- Development approach.
- System architecture of the node.
- Middleware architecture.
- Simulation environment.

The Mobile eLearning Node is described in full within 27 published papers of which 6 papers are included in the chapter.

4.2 Development Approach

Development of context-aware, flexible and adaptive architectures is exceptionally difficult task, which is impossible to complete without a clearly-defined development approach. The proposed approach aims at the creating of an agent- and service-oriented architecture for efficient use in on-line e-Learning systems employing an InfoStation infrastructure for getting mobile access to eLearning services and educational resources within a University campus [113, 117, 118, 135]. The approach is based on the ideas suggested by the MDA specification of OMG [138]. In conformity with the approach the mobile eLearning Node was built on three levels (Figure 15):

- *Education Services Level* – implements and provides eLearning services supported by the node.
- *Scenarios Level* – presents scenarios for accessing the eLearning services and the teaching material over the InfoStation network. The scenarios are an abstract presentation of the underlying communication infrastructure and as such they hide redundant technical details. The middleware is developed so that it support a correctly execution of the proposed scenarios.
- *Middleware Level* – an agent-based multi-layered architecture playing a mediator role between the services level and the scenarios level. It contains agents, implementing scenarios execution, and agents supporting services provision.

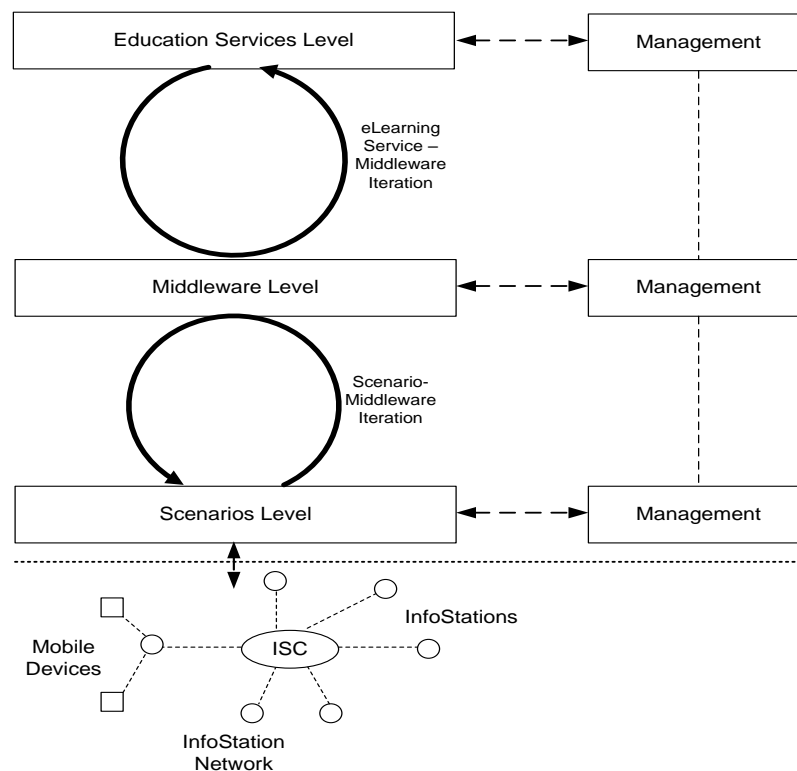


Figure 15: The approach: levels and iterations

The services are convenient for the implementation of specific business functionality, but are static, however. The necessary flexibility, required for a context-aware and adaptable architecture, is ensured by intelligent agents (which, however, are unsuitable for the

implementation of functionality). The agents have to spot any change in the environment, carry out the adaptation necessary for correct processing of service requests, and activate the desired eLearning services.

The architecture is developed iterative and incremental by means of separate iterations. The term *iteration* – borrowed from the Unified Software Development Process [139] – means a workflow or cooperation between the developers at different levels so as to be able to use and share particular products and artefacts. In our approach, the iterations afford an opportunity to step-to-step development and extension of the middleware. The approach defines iterations between:

- *The Scenarios Level and the Middleware Level*, and
- *The Education Services Level and the Middleware Level*.

Currently, the next *scenario-level-to-middleware-level* iterations are envisaged for realization in the node:

- First iteration, called *Base Architecture*, aims at the implementation of software support of four scenarios – ‘No change’, ‘Change of user mobile device’, ‘Change of InfoStation’ and ‘Change of InfoStation and user mobile device’.
- Second iteration is known as *Scenario-Based Management*. During this iteration some temporal aspects of scenario-based management are examined [119, 128]. Important changes in the user context during the execution of the users’ service requests (e.g., device mobility, when a user/device leaves the service area (range) of one InfoStation and enters another) can be detected and identified by the system only if the temporal aspect of this process is taken into consideration.

- Third iteration, known as *Adaptation*, concerns with problems related to strengthening the architecture, e.g. to support adaptability. We take the view that, personalized eLearning process could be successfully realized by means of adaptive architectures, supporting three models that influence the learning process – the user model, the domain model, and the pedagogical model.
- Fourth iteration, known as *Resource Deficit*, states that in some cases the user request for a service cannot be satisfied fully by the local InfoStation due to resource deficit. In this case the service provision must be globalized in a manner involving other InfoStations.

An *eLearning-level-to-middleware-level* interaction implements a mapping of an education service to the existing middleware. If the middleware doesn't provide full support to the provision of the service, it will be extended with new agents.

4.3 System Architecture

The software system architecture (Figure 16) is distributed, where the components are deployed across the different tiers/nodes, i.e. on mobile devices, InfoStations, and InfoStation Center. In this architecture the role of the InfoStations is expanded, enabling them to act (besides the mediation role) as hosts for the local mobile services and for preparation, adaptation, and conclusive delivery of global mobile education services [129, 136].

This way the service provision is efficiently distributed across the whole architecture. Each of the system network nodes have a different structure depending on their functioning within the system. However, each node is built upon a Communication Layer whose main task is to initialize, control and maintain communications between different nodes. This layer is also concerned with choosing the most appropriate mode of communication between a mobile

device and an InfoStation – whether that be Bluetooth or WiFi, or indeed as the platform evolves perhaps WiMAX in the future.

The software architecture of the InfoStations and InfoStation Center includes a Service Layer on the top. The main task of this layer is to prepare the execution of the users' service requests, to activate and receive the results of the execution of different services (local and global). The InfoStations' middle layer is responsible for the execution of scenarios and control of user sessions. It is at this layer where the user service requests are mainly processed by taking into account all contex-aware aspects and applying corresponding adaptive actions. The middle layer of the InfoStation Center ensures the needed management and synchronisation during particular scenarios. In addition, different business supporting components, e.g., for user accounting, charging and billing, may operate here.

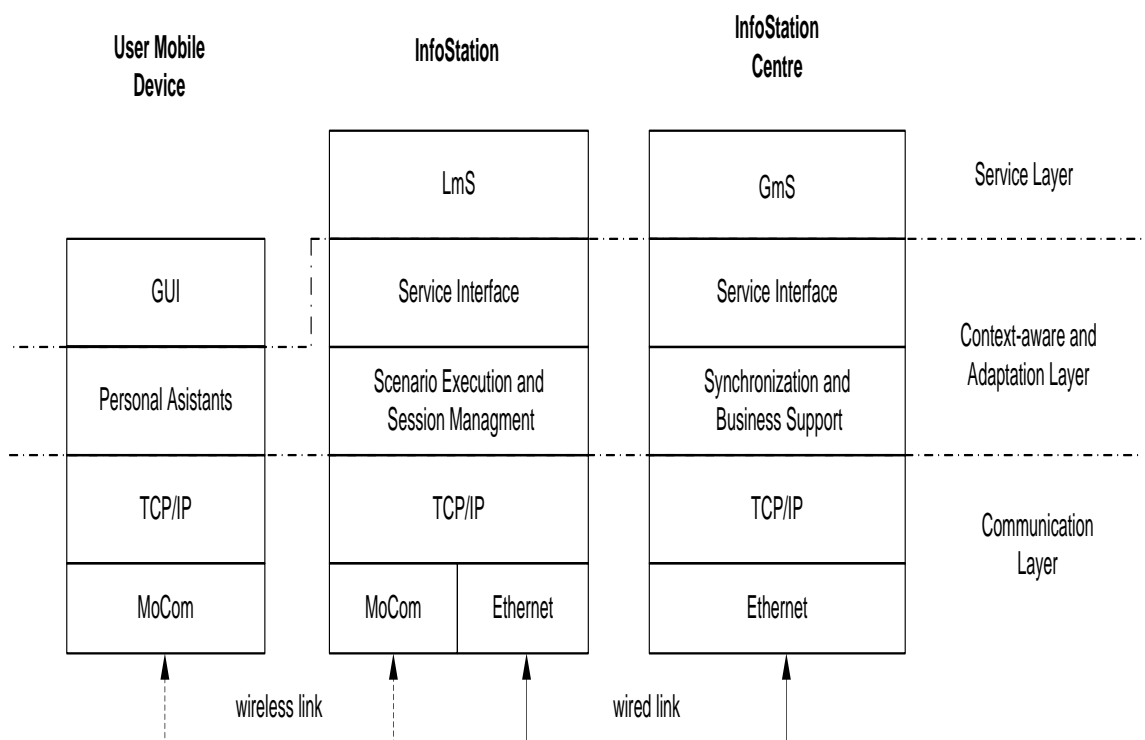


Figure 16: The layered system architecture

The software architecture of the user mobile devices contains two other layers:

- *Personal Assistant* – its task is to help the user in specifying the service requests sent to the node, accomplish the communication with the InfoStations’ software, receive and visualise the service requests’ results to the user, etc. Moreover, the assistant can provide information needed for the personalisation of services (based on information stored in the user profile) and/or for the synchronisation of scenario execution.
- *Graphical User Interface* – its task is to prepare and present the forms for setting up the service requests, and visualise the corresponding results received back from the node.

4.4 Context-Aware Service Provision in DeLC

In order to ensure a context-aware service provision, the software of the eLearning Nodes is built as an integration of two components [112]:

- A standardized *middleware*, which is able to detect the dynamic changes in the environment during the processing of user requests for services (*context-awareness*) and correspondingly to ensure their efficient and non-problematic execution (*adaptability*).
- A set of *eLearning services* realizing the functionality of the application area (*business functionality*), which could be activated and controlled by the middleware.

As the middleware is concerned with the context-awareness and adaptability aspects, it is important to first clarify these concepts. Within DeLC development approach, Dey’s definition [140] was adopted, according to which “*context is any information that can be used to characterize the situation at an entity*”. An entity could be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user

and applications themselves. Context could be of different type, e.g., location, identity, activity, time.

Dey's definition is utilized here as a basis for further discussions. In order to elaborate on this definition a working one for the creation of the desired middleware architecture, we first solidify the definition. We want clearly to differentiate context-awareness from the adaptability. Context-awareness is the middleware's ability to identify the changes in the environment/context as regards:

- *Device mobility* – characterizes the mobile device's location. In some cases this mobility leads to changing the serving InfoStation. This is especially important due to the inherent mobility within the system, as users move throughout the University campus. This information has a bearing on the local services deployed within a particular area (i.e., within the University Library).
- *User mobility* – characterizes the changing of the mobile device used from the user. This mobility offers different options for the delivery of the service request's results back to the user. What is important here is to know the capabilities of the new device activated by the user, so as to adapt the service content accordingly.
- *Communication type mobility* – depending on the current prevailing wireless network conditions/constraints, the user may avail of different communications possibilities (e.g., Bluetooth or WiFi).
- *User background and preferences* – service and teaching content personalisation may be needed as to reflect the changes made by users in their preferences or background; e.g., the way the service content is visualised to them, etc.
- *Pedagogical preferences* - goal-driven sequencing of tasks engaged in by the user in order to obtain the desired educational results.

- *Environmental context* - issues such as classmates and/or learner/educator interactions.

The goal of adaptability is to ensure trouble-free, transparent and adequate fulfilment of user requests for services by taking into account the various aspects of the context mentioned above. In other words, after identifying a particular change in the service environment, the middleware must be able to take compensating actions (counter-measures) such as handover of user service sessions from one InfoStation to another, re-formatting/transcoding of service content due to a change of mobile device (varying device capabilities), service personalisation, etc.

4.5 Scenarios Level

To ensure adequate support for user mobility and device mobility (the first two aspects of the context change), the following four main communications scenarios are identified for support in our middleware architecture [115, 130]:

- *'No change' Scenario* – an eLearning service is provided within the range of the same InfoStation and without changing the user mobile device.
- *'Change of user mobile device' Scenario* – due to the inherent mobility, it is entirely possible that during an eLearning service session, the user may shift to another mobile device; e.g., with greater capabilities, in order to experience a much richer service environment and utilize a wider range of resources.
- *'Change of InfoStation' Scenario* – within the InfoStation paradigm, the connection between the InfoStations themselves and the user mobile devices is by definition geographically intermittent. With a number of InfoStations positioned around a University campus, the users may pass through a number of InfoStation serving areas during the service session. This transition between InfoStation areas must be

completely transparent to the user, ensuring the user has continuous access to the service.

- '*Change of InfoStation and user mobile device*' Scenario – most complicated scenario whereby the user may change the device simultaneously with the change of the InfoStation.

To support the third aspect of the context change (different communication type), the development of an intelligent component (agent) working within the communication layer is envisaged. This component operates with the capability to define and choose the optimal mode of communication, depending on the current prevailing access network conditions (e.g., congestion level, number of active users, average data rate available to each active user, etc.). The user identification and corresponding service personalisation is subject to a middleware adaptation for use in the particular application area. In the case of eLearning, the architecture is extended to support the three fundamental eLearning models – the educational domain model, the user/learner model, and the pedagogical model.

4.6 Middleware

Currently, the middleware is developed as result of the *Base Architecture* iteration. The main challenges related to the support of distributed control, as the system should be capable of detecting all relevant changes in the environment (context-awareness) and according to these changes, facilitate the service offerings in the most flexible and efficient manner (adaptability). The system architecture (presented in this chapter) is implemented as a set of cooperating intelligent agents. An agent-oriented approach has been adopted in the development of this architecture in order to:

- Model adequately the real distributed infrastructure.

- Allow for realisation of distributed models of control.
- Ensure pro-active middleware behaviour which is quite beneficial in many situations.
- Use more efficiently the information resources spread over different InfoStations.

Moreover, the agent-oriented architecture can easily be extended with new agents (where required) that cooperate with the existing ones and communicate by means of a standardized protocol (FIPA -Agent Communication Language (ACL) [146]). Indeed, the InfoStations and InfoStation Center exist as networks of interoperating agents and services, with the agents fulfilling various essential roles necessary for system management [123, 124]. Within each of these platforms, agents take responsibility for selecting and establishing a client-server cross-platform connection, conveyance of context information and the delivery of adapted and personalised service content. This multi-agent approach differs from the classic multi-tier architectures in which the relationships between the components at a particular tier are much stronger. Conceptually we define different layers in the system architecture in order to present the functionality of the middleware that is being developed in a more systematic fashion. The middleware architecture is considered as a set of interacting intelligent agents. Communication between the user mobile devices and the serving InfoStations is supported in two ways:

- An agent operating within the InfoStation discovers all new devices entering the range and subsequently initiates communication with them or
- Personal Assistant agents on the user mobile devices are the active part in communication, and initiate the connection with the InfoStation.

Currently, the former approach is used for Bluetooth communication, whereas the latter applies for WiFi communication.

Figure 17 highlights the main components necessary to ensure continuity to the service provision; i.e., support for the continuous provision of services and user sessions in the case of scenario change or resource deficiency. The agents, which handle the connection and session establishment, perform different actions, such as: searching for and finding mobile devices within the range of an InfoStation, creating a list of services required by mobile devices, initiation of a wireless connection with mobile devices, data transfer to- and from mobile devices. Also illustrated within Figure 17 are the components which serve to facilitate a level of context sensitivity and personalisation to the presented services. A short description of the agents (for Bluetooth communication) within the architecture is presented below.

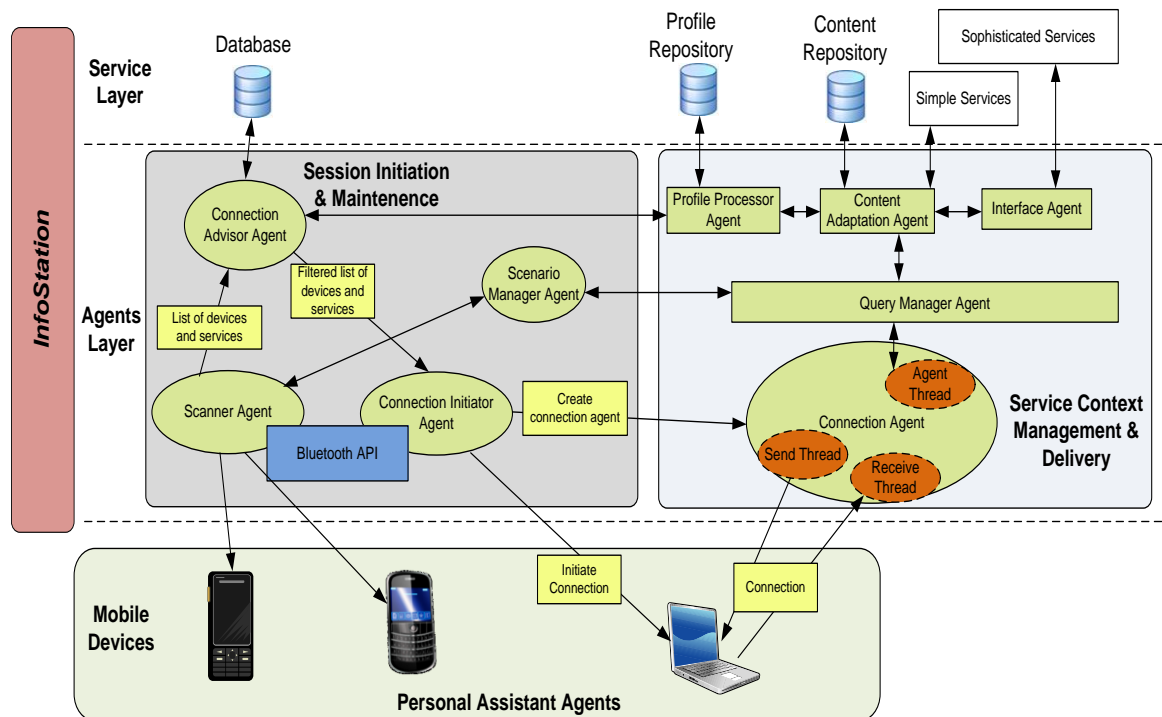


Figure 17: The Agent-Oriented Middleware Architecture

The first step in the delivery of the services involves the Scanner agent, which continuously searches for mobile devices (Personal Assistant agents) within the scope of the InfoStation. In addition, this agent retrieves a list of services required by users, as well as the profile information, detailing the context (i.e., device capability and user preference information,

received as XML file), which itself is extracted from the content of an ACL message. The contents of this XML file are then passed on (via the Connection Advisor agent) to the Profile Processor agent, which parses the received profile and extracts meaningful information. This information can in turn be utilized to perform the requisite alterations to services and service content. This information is also very important in relation to the tasks undertaken by the Scenario Manager agent. The role of this agent is to monitor and respond to changes in the operating environment, within which the services are operating (i.e., change of mobile device). In the event of a significant change of service environment, this agent gathers the new Capability and Preference Information (CPI) via the Scanner agent. Then, in conjunction with the Query Manager agent and the Content Adaptation agent, facilitates the dynamic adaptation of the service content to meet the new service context. The main duty of the Connection Adviser agent is to filter the list (received from the Scanner agent) of mobile devices as well as requested services. The filtration is carried out with respect to a given (usually heuristic) criterion. Information needed for the filtration is stored in a local database. The Connection Adviser agent sends the filtered list to the Connection Initiator agent, who takes on the task of initiating a connection with the Personal Assistant onboard the mobile device. This agent generates the so-called Connection Object, through which a communication with the mobile device is established via Bluetooth connection. Once this connection has been established, the Connection Initiator generates dynamically an agent (named Connection agent) to which it hands over the control of the connection. From this point on, all communications between the InfoStation and the Personal Assistant are directed by the Connection agent. The internal architecture of the Connection agent contains three threads:

- An agent thread used for communication with the Query Manager agent.
- A Send thread.

- A Receive thread.

The Query Manager performs one of the most crucial tasks within the InfoStation architecture. It determines where information received from the mobile device is to be directed, e.g. directly to simple services, or via Interface agents to sophisticated services. It also transforms messages coming from the Connection agent into messages of the correct protocols to be understood by the relevant services; i.e., for simple services – UDDI [147] or SOAP [148], or for increasingly sophisticated services by using more complicated, semantic-oriented protocols (e.g., OWL-S). The Query Manager agent also interacts with the Content Adaptation agent in order to facilitate the Personal Assistant with increasingly contextualised service content. This Content Adaptation agent, operating under the remit of the Query Manager agent, essentially performs the role of an adaptation engine, which takes in the profile information provided by the Profile Processor agent, and executes the requisite adaptation operations on the service content (e.g., file compression, image resizing etc.).

The Query Manager agent receives user service requests via the Connection agent, and may communicate with various services. Once it has passed the request on to the services, all service content is passed back to the Query Manager via the Content Adaptation agent. The Profile Processor agent parses and validates received profiles (XML files) and creates a Document Object Model (DOM) tree [149]. Using this DOM tree the XML information may be operated on, to discern the information most pertinent to the adaptation of service content. The Content Adaptation agent receives requests-responses from the services, queries the Profile Processor agent regarding the required context, and then either selects a pre-packaged service content package which closely meets the requirements of the mobile device, or applies a full transformation to the service content to meet the constraints of the operating environment of the device.

The tasks undertaken by the Content Adaptation agent, the Scenario Manager agent and the Profile Processor agent, enable the system to dynamically adapt to changing service environments, even during a particular service session. Once the connection to a particular service has been initialized and the service content adapted to the requisite format, the Connection agent facilitates the transfer of the information to the user mobile device.

4.7 Education Services Level

On the top of the middleware operates a Service Interface Layer including agents whose task is to choose and activate service(s) corresponding to the current user request. We are going to investigate agents which could support two service models (Figure 18) [137]:

- *Web Services like model* – the Web Services are wide used in the current service-oriented applications. However this model suffers from the lack of their ability to provide constructs and concepts that enable reasoning about runtime service behavior.
- *More adaptable service models* – in these models services are composed using a specification technique that characterizes ongoing behavior of the services (e.g., [141]). In these models context-awareness and adaptation can be enhance even at the service level.

The agents in this level support a communication with more sophisticated services performed by using the standard Ontology Web Language for Services (OWL-S) protocol [142]. OWL-S provides a set of constructs for creation of ontologies, which are machine understandable descriptions of the service. These ontologies enable agents to discover, invoke and interoperate with the services. Using OWL-S, the ontology structure is divided into four separate sections, each dealing with a different aspect of the service [112]:

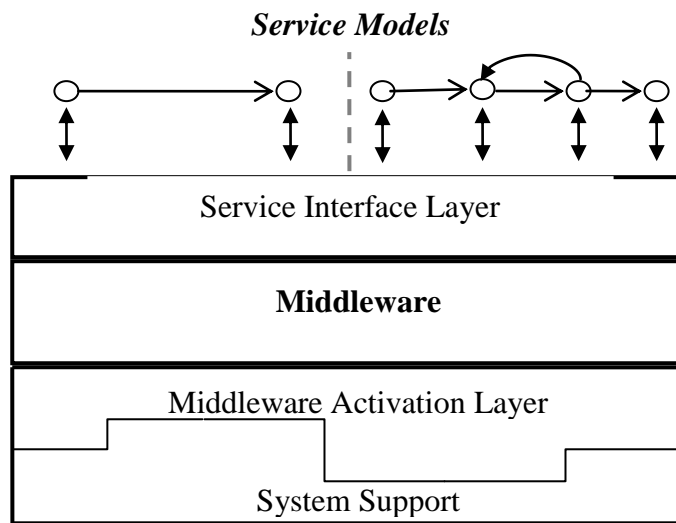


Figure 18: Two service models

- *Service Profile* - this advertises the abilities of the service (i.e., what it can do), in such a way as to enable a service-seeking agent to determine if the service meets its requirements. This provides a concise high-level description of the service to a registry, which is maintained by the InfoStation Center, and periodically disseminated copies of all service profiles to the various local InfoStations throughout the campus. The service profile outlines the limitations of the service, the quality of the service and of course the requirements placed upon the requesting agent in order to utilize the service successfully. In DeLC architecture, the InfoStation provides a user's Personal Assistant with access to a registry of services. Service profiles are comprised to three different types of information:
 - *Description* of the service and the service provider.
 - *Functional attributes* that provide supporting information about the service.
 - *Functional behavior* of the service which provides a capability description of the service, specification of what the service provides.
- *Process Model* - this gives a detailed description of the operation of the service and tells a requesting agent how and when to interact with a service (read/write messages).

Essentially it outlines how to initiate the service, and what happens when a service carries out its purpose. The process model is utilized for the invocation of the service, for planning a composition of complex actions, and for interoperation and monitoring of service functionality.

- *Grounding* - this provides details of how an agent can access and interact with a service. Specified within the grounding, are details pertaining to message formatting, transport mechanisms, protocols, addressing, etc. When combined with the details outlined within the process model, all the information necessary for a client to utilize a particular service is presented. While both the Service Profile and the Process Model provide abstract representations of the service and its processes, the Grounding provides a concrete specification of the various elements required for interaction with the service. The Web Services Description Language (WSDL) [143] is utilized within OWL-S to facilitate the initial grounding mechanism. Figure 19 illustrates the convergence of these two specification languages.

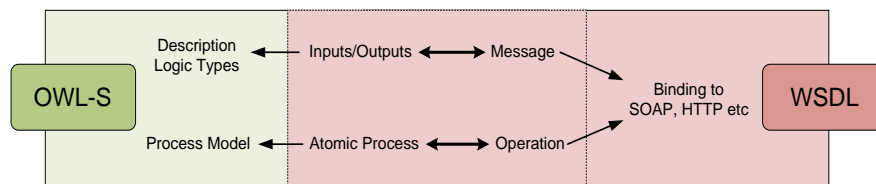


Figure 19: The mapping between OWL-S and WSDL

- *Service* - this simply binds the other elements together into a single entity which can be published and invoked.

Agents utilize the information contained within the Service Profile [120] to ascertain whether or not a service meets its requirements, and adheres to certain constraints such as security, quality of service, etc. While the Service Profile provides all the information needed for an agent to discover a service, the Process Model provides the information necessary for the

agent to use the service. The Process Model allows the agent to perform a more in-depth analysis of the service and its capabilities, and determine if it can be utilized. As well as this, it enables the agent to compose new service descriptions through the composition and interoperation of previous existing services, to perform specific tasks. The Process Model also allows agents to monitor the execution of tasks performed by a service (or a set of services), and to coordinate the entities involved in the service execution. The Service Grounding details how agents can communicate with, interoperate with, and invoke a service. The relationships between the various service components are modelled using properties such as presents (Service to Profile), describedBy (Service to Process Model) and supports (Service to Grounding), Figure 20

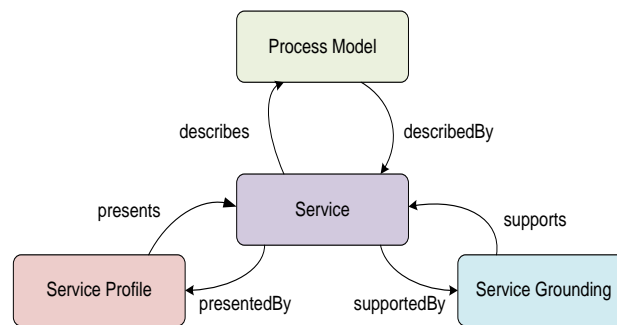


Figure 20: The OWL-S Service Ontology

When all these separate parts are combined, they form an ontology/description that allows agents to discover, invoke, compose and monitor services. Utilizing the OWL-S protocol offers a good opportunity for the realisation this flexible software architecture, offering a suitable environment for the support of education services in DeLC.

One of the fundamental futures of DeLC is the facility for service context-sensitivity and personalisation. In order to support this functionality we turn to the creation of device-, user- and service profiles. For the implementation of the former two profiles, which are integral components in the service adaptation procedure, we have opted to use the uniform platform-

independent Composite Capability/Preference Profile (CC/PP) format [144]. This format is based on the Resource Description Framework (RDF), one of the key specifications of the Semantic Web, and is recommended by the World Wide Web Consortium (W3C) [145]. When adapting service content for a specific user, the information required for the adaptation can come from different sources - the network, the accessing device or indeed the user's own preferences/context. The InfoStation can receive these different pieces of information separately, but needs to merge the information into one model before doing content adaptation. Based on the Semantic Web and RDF, CC/PP simplifies this data integration through the use of extensible and non-centralised vocabularies.

Functionality, modeling and support of various eLearning services are presented in a round of publications, such as [125, 126, 127, 131, 132, 133, 134]. In this section is demonstrated *mTest* service, which provides a means for educators to rapidly evaluate learners assimilated knowledge and provide valuable feedback to learners concerning their progress [121, 122]. The mTest facility enables the educator to more effectively shape the learning experience of the learners, ensuring the learner remains engaged in the correct material. The sequence diagram Figure 21, depicts sample interactions between entities involved within the provision of the mTest service through this enhanced architecture. As has been mentioned previously, the Scanner agent continuously searches for mobile devices/Personal Assistant agents within the service area of the InfoStation. The Personal Assistant, having already gathered the requisite information from the user to complete the user profile, passes the capability and preference information (CPI) on to the Scanner agent within the contents of an ACL message. The Scanner agent takes the information relating to the Personal Assistant, directs the profile information to the Profile Processor agent, and directs a list of mobile devices and applicable services to the Connection Advisor agent. The latter in turn filters this information providing the information necessary for the Connection Initiator agent to establish a communication

session with the Personal Assistant onboard the mobile device. The Connection Initiator agent, having already established communication with the Personal Assistant, generates a Connection agent, which is tasked with maintaining and managing all communications between the InfoStation and the particular Personal Assistant. The Profile Processor agent, having received the profile information from the Scanner agent, gathers the relevant capability and preference information from the received XML content. This information is cached locally within the InfoStations profile repository. This agent works in conjunction with the Scenario Manager agent in order to monitor for any changes to the operating environment of the services (i.e., change in capabilities caused by changing one mobile device with another). At this stage, a communication session has been established between the InfoStation and the user mobile device, and the InfoStation is aware of the context/constraints any presented services must be adapted to.

Having successfully established communication between the InfoStation and the Personal Assistant, the user is presented with a list of all available services matching his/her profile. From this list the user selects a particular service, in this case the mTest service, and may then choose a particular assessment. The Personal Assistant on the user mobile device forwards the users service request on to the InfoStation, specifying the user choices. The Connection agent passes this message on to the Query Manager agent. It determines to which service a particular service request must be directed. The mTest service receives the user service request and compiles a list of applicable assessments, taking into account the personal context of the user (based on profile information). This service content is passed on to the Content Adaptation agent, which performs the role of an adaptation engine. It performs the requisite

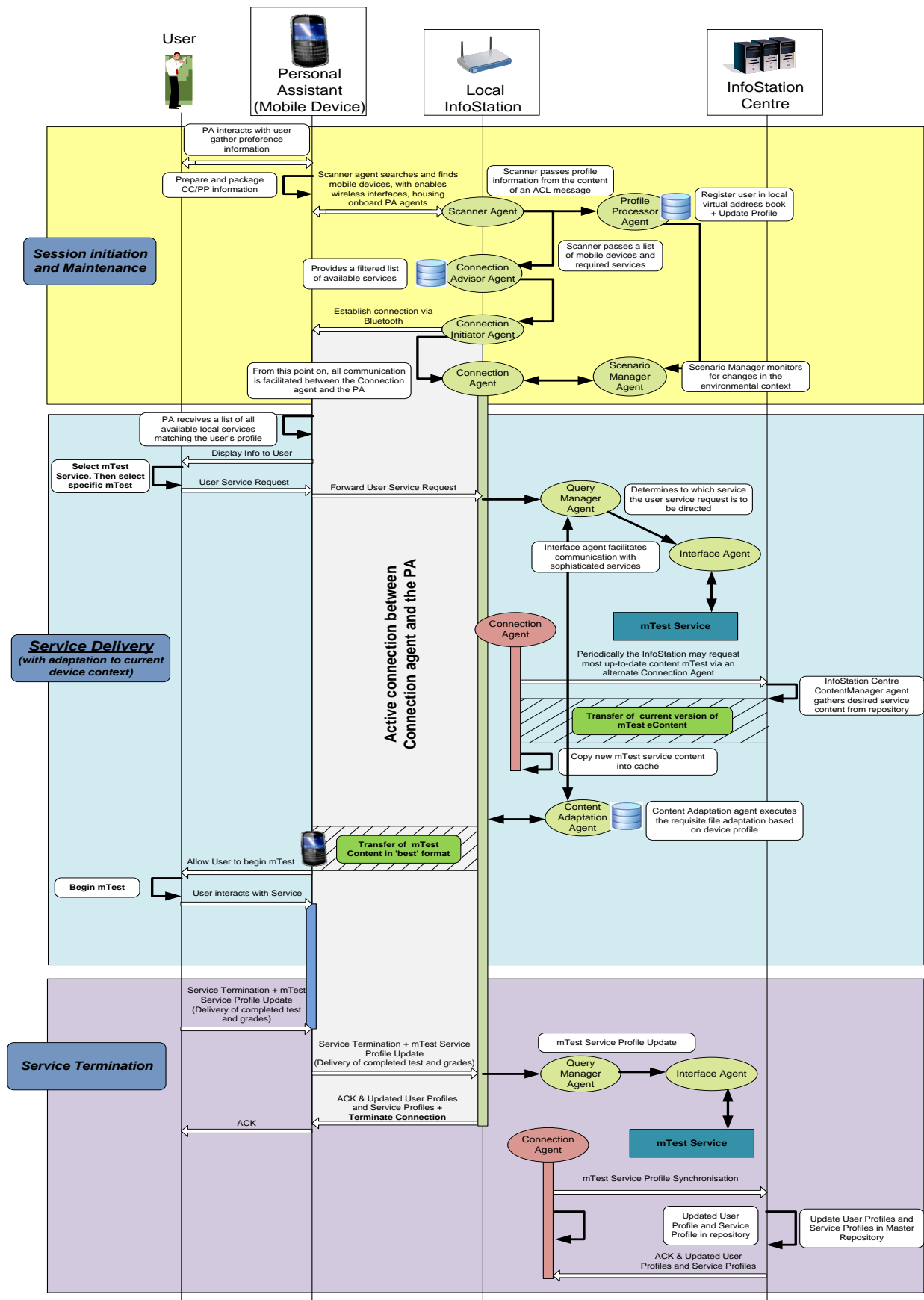


Figure 21: mTest Service Provision

personalisation and adaptation of the service content, based on the capabilities of the access device and the preferences/context of the user. This contextualisation ensures the service is presented in its most efficient format by identifying the most relevant context information such as the screen size, audio and video capabilities etc. Once the service content has been adapted to suit the operating environment on board the mobile device, the content is packaged (MIDlet packed within a JAR file) for transfer to the user's mobile device. Depending on network constraints such as the traffic volume at a particular time and of course the communication capabilities of the target device, the Connection agent may initiate the file transfer via Bluetooth (or WiFi). The Personal Assistant presents a basic GUI to the user, enquiring as to whether or not to save the incoming file. The incoming MIDlet (JAR file) is saved to onboard memory within the mobile device, which in turn permits the user to access the service content any number of times, even whilst outside the range of the InfoStation. Within this sequence diagram, the facility to update service content is also highlighted. Periodically, a Connection agent (one specifically generated to maintain communications between the InfoStation and the InfoStation Center) requests the most up-to-date versions of particular service content. This facility is essential to the success of the mLearning architecture. Any updates can be propagated from the InfoStation Center, ensuring that each InfoStation throughout the network has access to the latest versions of service content. As the learner progresses through the test, his/her user profile is maintained to reflect this progress. Furthermore we consider the possibility for the learner to do the test whilst on the move and out of range of an InfoStation. As the content is stored locally on the mobile device, the Personal Assistant can facilitate the user with continued utilisation of the service while at the same time maintaining the user profile. Thus the learner may complete the test whilst outside the radio range of any InfoStation, with the user profile reflecting the learner's progression through the material. Once the Personal Assistant eventually does enter back within range of

an InfoStation, the Personal Assistant will forward on a user profile update which reflects the progress of the learner through the test content, whilst out of range of the InfoStation network. These updates are disseminated through to the InfoStation Centre so as to ensure all information across the system is up-to-date. Once the user has completed the mTest, the Personal Assistant displays the results of the assessment to the user, providing valuable feedback on their own progress and performance.

4.8 Resource Deficit Iteration

On one hand, in the original paradigm, the InfoStations operate as mediators between the user mobile devices and a central server on which a variety of applications are installed. In DeLC architecture, this communications paradigm was enriched by extending the role of the InfoStations and deploying services that could be locally activated directly on them. With this distributed deployment of services and other information resources on all available InfoStations, we aim to achieve a good load balancing and better efficiency. However, the distributed deployment of information resources shows some downsides related to the system overhead associated with the interconnection of information resources during the execution of a service request. This overhead is particularly heavy when the interconnected resources are deployed on different InfoStations. Thus the task of overhead minimization is especially important in the distributed approach. The amount of overhead depends on the manner of deploying the information resources.

On the other hand, in some cases the user requests for particular services cannot be satisfied fully by the local InfoStation due to resource deficit (e.g., when information needed to satisfy the service request is unavailable in the database of this InfoStation). In these cases, the service provision must be globalized in a manner involving other InfoStations (through the InfoStation Center, ISC). The need for globalization depends on the manner in which the

resources are deployed on network nodes during the system initialization. Taking into account the fact that each globalization requires an additional overhead, the problem is to find such a deployment, which minimizes the number of globalizations.

In order to solve these problems (in the iteration *Resource Deficit*) an abstract model of the resource deployment in DeLC is developed (detailed description in [114, 116]). For the reason of complexity we decline to search a global optimum. Instead we are going to specify an approach which: seek suitable heuristics for specifying a real subset of possible deployments, where (with high probability) we assume that a satisfying solution exists and seek an optimum within the new set (local optimum or suboptimum). In this way the approach prescribes the following four steps (Figure 22):

- RG-problem presentation (RG-model generation) – the problem has been presented as an abstract graph network.
- RG-model transformation – the model has to be transform into a more abstract graph (called Abstract Deployment Graph) in order to be prepared for solving by an evolution strategy.
- Optimal Deployment Graph (ODG) generation – by help of an evolution strategy an optimal graph has been generated as solution of the optimization problem.
- ODG interpretation – the problem solution (ODG) is presented as a real deployment map of the information resources.

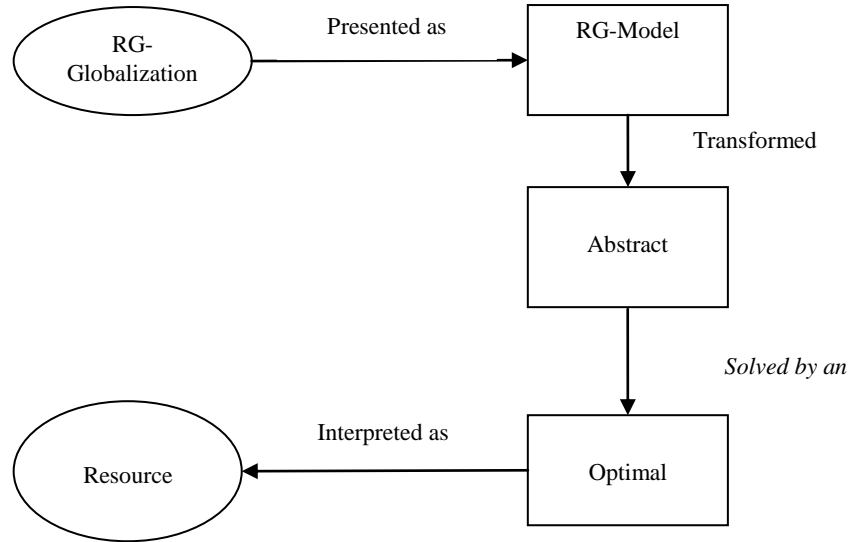


Figure 22: Approach diagram

4.9 Simulation Environment

A simulation environment for middleware testing is developed [111]. By means of this environment the middleware's behavior can be analyzed in the framework of separate experiments. The simulation system is being developed as a wrapper of the tested middleware, in which the following activities are carried out:

- Preparation and initialization of the experiment.
- Activation of the experiment execution.
- Visualization of the middleware's behavior during the experiment execution.
- Automatic generation of protocols, which may be used for the assessment of the middleware's behavior.

The simulation environment includes two main sub-environments (Figure 23):

- *Experiment Organizer (EO)* - specialized graphical environment used for the preparation of experiments and presentation of results. Experiments are described as a

specification; the results are recorded and are available for subsequent analysis. Beside this, some results may be visualized. A generator of protocols has been developed in EO too.

- *Experiment Runner (ER)* - an agent-oriented wrapper of the tested middleware, which controls the experiment execution/running in accordance with the given specification. The interim results obtained during a particular experiment are transmitted to EO, where the experiment execution can be visualized.

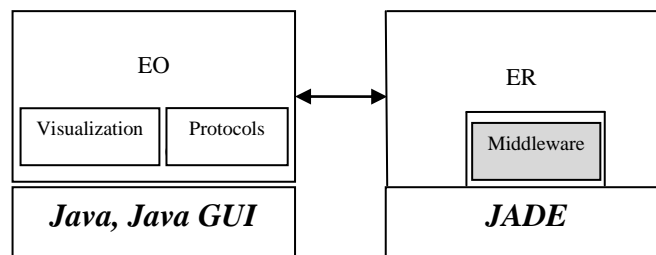


Figure 23: The logical structure of the simulation environment

The simulation environment can present all events occurring in the middleware in the form of a log file, e.g. the actual moments when agents/containers/behaviors are created or removed, communications messages exchanged between agents, changes in the state of agents/containers/behaviors, etc. The experiments, carried out with the help of this environment, allow research and analysis of the middleware's behavior at four levels as described in the following subsections.

- *Scenario level* - at this level the global behaviour of the middleware during the execution of the scenario is traced. Our particular interest is focused on the behaviour of the resident agents and the process of generation of operational (temporary) agents, which depend on the changes occurring in the environment. These changes are specified in the description of the experiment and are simulated during its execution. As a result of the experiment, the following features of the middleware can be analyzed: tracking the movement of users and the corresponding reaction of the

middleware, statistics about generated and removed operational agents, starting and completing a communication, data transfer between devices, agents' review.

- *Container level* - at this level, for each simulated component (InfoStation Center, InfoStation, or mobile device) the simulation environment shows the active containers living in the middleware. The following information that characterizes a container could be extracted: the address, the identifier (identifying the container in the JADE platform [150]), the logical name, the port, the protocol used for communication, etc. The main container also contains two functional JADE agents: an AMS (Agent Management System) and a DF (Directory Facilitator). AMS is a mandatory component of the platform responsible for the overall management of its operation, such as the creation and deletion of agents, overseeing the migration of agents to and from the platform, etc. DF is an optional component providing yellow-pages services to other agents.
- *Agent level* - at this level the simulation environment shows what agents live in the containers of each InfoStation. The following information characterizing an agent could be extracted: the name of the class whose instance is this particular agent; the local/global name for JADE; the state of the agent; the container in which it is located, etc. Beside this, different characteristics of agents can be visualized, such as the location of the agent (i.e. on a mobile device, InfoStation, or InfoStation Center), creation of an agent (the simulation environment catches the exact creation time and visualizes it - location, container, name, state), removal of an agent, etc. This level collects also information that allows analysis of interactions between agents. The simulation environment may intercept messages, exchanged between agents. Information, which may be visualized for each message, includes: the sender, the recipient, and the contents of the message encoded in ACL.

- *Behaviour level* - the purpose of this level is to provide opportunities for analysis and assessment of local behaviour of individual agents. The agents maintain their own libraries of different “behaviours”, which could change depending on the environment. The simulation environment obtains the following information during the process of operation of the observed agents: behaviour identifier, class identifier whose instance is this behaviour, behaviour type (simple or complex) and other behaviours contained in the current behaviour (for complex behaviours only). At this level the simulation environment can visualize the following behaviour characteristics: affiliation to an particular agent, creation, addition, or removal, etc.

ER has an agent-oriented architecture (Figure 24), implemented by means of JADE, which allows smooth integration of the middleware. ER consists of the following agents:

- *Simulation Controller (SC)* - the desired configuration of the InfoStation (IS) network is simulated with the help of this agent. It is the central coordinator of the experiment, which is currently executed. This agent issues commands for the creation or removal of simulated InfoStations and mobile platforms, defines waiting intervals and the sequence of interaction between simulated mobile and IS platforms. This agent is also an interpreter to a specialized script language that describes the desired experiment. Currently we are studying the possibilities for using the Interval Temporal Logics and its interpreter Tempura [151] for describing the experiments.
- *SIS Handler (SH)* – this agent controls a separate Simulated InfoStation (SIS) platform. Depending on the purpose of the experiment and on the commands received from the Simulation Controller, it generates different commands addressed to the managed SIS and received by the corresponding Spy agent. The SIS Handler takes care of the creation and removal of the specific SIS, by setting the relevant parameters

needed for the operation of the JADE platform and the desirable initialization of the Spy agent.

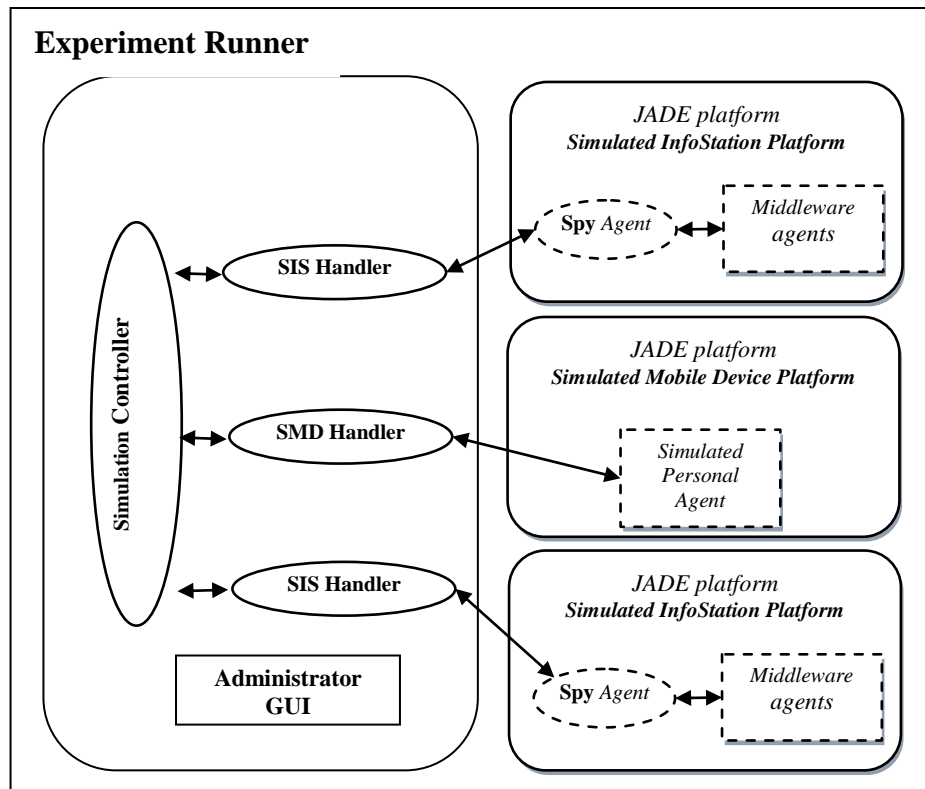


Figure 24: The architecture of the Experiment Runner (ER)

- *Spy Agent* - the role of this agent is to execute the commands issued by the Simulation Controller agent on the SIS platform. In addition, it provides feedback to the simulation environment on the events that have occurred in the SIS platform.
- *SMD Handler* – this agent controls the simulated mobile device in a similar way as the SIS Handler controls the SIS. He interprets the commands received from the Simulation Controller agent, processes them and submits corresponding commands to the simulated personal agent. The SMD Handler takes care of the creation and removal of the Simulated Mobile Device (SMD) platform and its desired initialization.
- *Simulated Personal Agent* – this agent plays the role of a user's personal agent; it simulates the user actions based on the commands it receives from the SMD Handler.

It also provides feedback by reporting on the events occurring in the simulated mobile device platform and on the exchanged messages. Another task of its own is to connect to the various SIS and to issue requests for the execution of services.

The ER's agents are deployed on the following platforms:

- The *Simulation Environment Manager* is a JADE platform, on which the Simulation Controller and the Handler agents live. It serves as a central console for the control of experiments. Work with the environment and visualization of the processes is carried out through a specialized graphical user interface (GUI) used by the administrator of experiments. By using this interface, the administrator of the simulation is able to issue commands to the controller, to start and stop experiments, to monitor the current status of the environment and the front conditions in the form of a log file.
- The *Simulated InfoStation Platform* is a JADE platform on which the unmodified agents of the standard middleware are located. The only addition is the presence of the Spy agent, which provides a feedback to the controller. The SISs provide the same conditions for the operation of middleware as the real InfoStations, with only one difference that all of them are deployed on the same computer.
- The *Simulated Mobile Device Platform* is a JADE platform, on which the simulated personal assistant is operating.

4.10 Summary

The Mobile eLearning Node is presented in this chapter. The DeLC approach supports the development of three-layer architecture within various iterations. The software system architecture is distributed, where the components are deployed across the different tiers, i.e.

on mobile devices, InfoStations, and InfoStation Center. The Scenarios Level, the Middleware, and the Education Services Level are presented in more detail. A simulation environment for testing developed as a wrapper of the middleware is presented too. The Mobile eLearning Node has been fully implemented by help of the development environment JADE.

Chapter 5

DeLC Education Portal

Objectives:

- To present the service-oriented architecture of the Education portal of DeLC.
- To discuss the personalization scheme implemented in the portal.
- To present the eContent Development Environment Selbo2.

5.1 Introduction

This chapter gives a detailed treatment of education portal DeLC. The architecture of this component is completely service-oriented and it adopts a client-server architecture. It interfaces with end user and the rest of the system. In the education portal, there are incorporated *education services* and *system services*, called engines. In the last two years, the portal is used in actual education at the Faculty of Mathematics and Informatics, University of Plovdiv.

The DeLC Education Portal is described within 9 published papers (6 in Bulgarian) of which 2 papers are included in the chapter.

5.2 Personalization

In order to support an interactive and personalized delivery of services and content, there is designed a uniform approach for the representation and processing of various types of information objects in the portal [158]. For each type of object we have established a classification scheme and profile type. Thus, an object can be uniquely identified according to its location in the classification scheme and its individual profile. A profile is presented as a set of ordered pairs $\mathbf{P} = \{(a_i, v_i) \mid a_i - \text{attribute}, v_i - \text{value}\}, i = 1, \dots, n$, that is open and can be extended with new attributes. It is possible to specify relations between/under the different objects.

The following basic types of information objects are supported in the portal:

- Users.
- Electronic content.
- Services.

- Events.

5.2.1 Users

A detailed classification scheme for potential users is implemented using the capabilities of the portal framework LIFERAY [161]. In the classification the following types of user object are defined (Figure 25):

- *Users* - users can be collected in multiple ways where they can be members of organization hierarchies, of arbitrary user groups, of communities, which draw together common interests, etc. They can have roles, which describe their functions in the system.
- *User Groups* - User Groups are simple, arbitrary collections of users, created by administrators. They can be members of communities or roles.

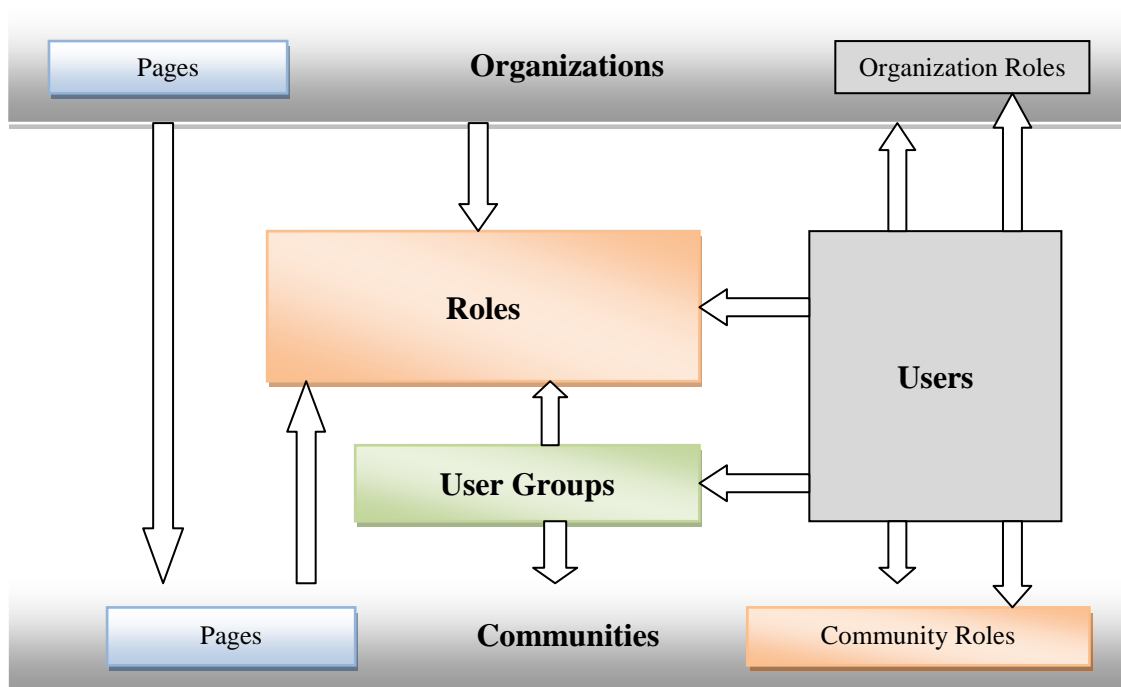


Figure 25: Users' classification scheme

- *Roles* - Roles are used to define permissions across their scope: across the portal, across an organization, or across a community. Users, User Groups, Communities, or Organizations can be members of a role.
- *Organizations* - Organizations are hierarchical collections of Users. There is also a special type of Organization called a location, which can define where users are specifically located.
- *Communities* - Communities are collections of Users who have a common interest. There are three types of Communities: Open, Restricted, and Hidden.

Users can be collected into User Groups and Users can belong to Organizations. Organizations can be grouped into hierarchies. Users, Groups, and Organizations can belong to Communities that have a common interest.

5.2.2 Electronic content

In the portal, a two-level classification scheme of electronic content is supported. The **first level** specifies various types of containers for electronic content, implemented in the form of **digital libraries**. Currently, the following digital libraries are implemented:

- *Lecture Courses Digital Library* – a repository for keeping of teaching material as lecture courses; the preferred form is SCORM electronic packages.
- *Questionnaire Digital Library* – various types of questions that can be used for creating of electronic tests.
- *Test Digital Library* – in this library, templates of electronic tests are saved.

- *Course Projects Digital Library* –packages containing description, documentation and results of course projects are saved in this library.
- *Diploma Theses Digital Library* - packages containing description, documentation and results of diploma theses are saved in this library.

In the **second level**, suitable formatting and structuring schemes are provided that allow a flexible preparation of the teaching material so that:

- *Atomic building blocks* can be identified.
- *Complex structures* of the content should be built under certain rules from the simpler ones.
- Different types of knowledge can be explicitly presented and modeled – for example, the knowledge, incorporated in an education process should be presented in three models - *domain model*, *pedagogical model* and *user model*.

One possible representation, satisfying these requirements, is the standard SCORM 2004 [162]. The reference model of SCORM is a set of interrelated technical standards, specifications and guidelines developed to meet the highest requirements for learning content and systems governing the content. In practice, SCORM uses context-independent learning objects SCOs (Sharable Content Objects), which are built from simple building blocks, called Assets. To describe the characteristics of SCOs, we use metadata. The created training resources are stored in a special online store called Content Repository. The SCORM model includes:

- *Content Aggregation Model (CAM)* - defines the components of the SCORM data model, how these components can be grouped in a comprehensive package and shared

between different systems. In this model there are multiple meta-data elements that provide identification, as well as storing, finding and using the information objects more quickly and easily.

- *Run Time Environment (RTE)* - can be considered as a set of requirements for LMS to manage the run-time environment, in which learning resides. LMS takes care to ensure the proper sequence of actions performed by the student, depending on their behavior and the current training scenario.
- *Sequencing and Navigation (SN)* - defines the methods for possible behavior through a sequence of discrete actions. This model is established on the IMS SS specification [163] based on the fundamental concepts of learning activities and learning objects. The model describes the sequence of actions in the learning process, depending on the occurred events and behavioral responses of students through a series of rules under which LMS transmits the initiative between training objects.

Currently, the SCORM Engine has been integrated as an *external service* for the interpretation and visualization of the SCORM content in the education portal. The generation of this content is done by the Selbo 2 environment (described in 5.4.) or the Reload editor [164] (the Bulgarian version of this editor has been implemented in the DeLC project (Figure 26)).

5.2.3 Events

A model for event management, enabling the users to see and create events and also be notified for them in advance, is implemented in the DeLC portal [159]. An event reflects an important moment for the users, such as a lecture, an examination, a test, national holiday, birthday, etc. One event is characterized by attributes, such as a name, start and end date and time, details, and information if it is a recurring one, as well as rules for its recurrence. The

events can be yearly, monthly and weekly recurring. The events are configured by the administrator and the end users.

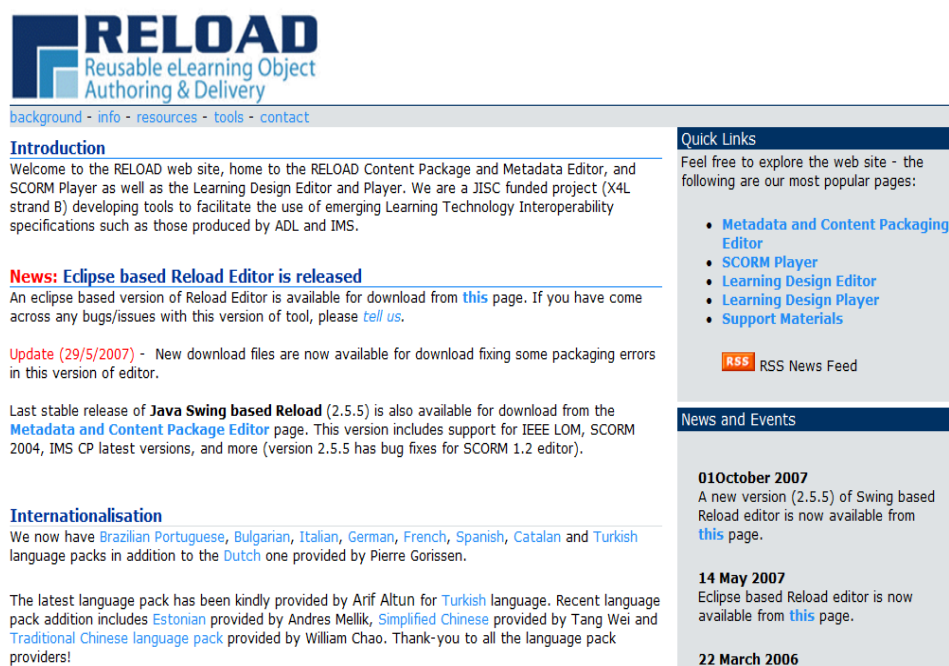


Figure 26: Reload Editor

5.2.4 Services

In accordance with the service-oriented architecture, the portal's functionality is provided as separate electronic services. A multi-level classification scheme of the services is proposed in order to improve the flexibility and the efficiency of the management. The first level is described in more detail in Chapter 3 of this thesis. Here, a more detailed description of the aspects *Functionality* and *Target Group* is given.

According the *Functionality* aspect the provided services are divided into two major groups:

- *Education services.*
- *System services.*

The system services, known as *Engines*, are transparent for the users and their basic purpose is to assist in the processing of the education services. Using the information, contained in the meta-objects (i.e., classification schemes, profiles), they can effectively support the activation, execution and finalization of the education services.

The education services are grouped as [155, 156, 160]:

- *Pre-Services* - this group consists of services for the preparation, organization and planning of the teaching process; e.g., educational process information, curricula generation, curriculum schedules.
- *eLearning Services* – services directly supporting the learning process; e-lectures, e-test, e-consultation, etc. [157].
- *Post-Services* – services for the recording and documentation of the teaching process; e.g., examination protocols, e-student books, teacher's e-diary, e-archive.

According to the *Target Group*, the education services are divided into the next categories:

- *Students' Services* – they are intended for the training and supervision of trainees. These include services for accessing and reading pre-prepared electronic lectures, for sitting for tests (mandatory or optional) in studied disciplines, for consultations with teachers, reviews of a student's book, as well as some additional services not directly related to teaching such as an intelligent personal calendar.
- *Teachers' Services* – they are intended for the teaching staff and allow the addition of lectures in the digital library, creating a library of test questions, creating, reviewing and evaluating tests and an intelligent personal calendar.
- *Administrators' Services* – they are intended for the management, configuration and planning of the teaching process and for related users, rights and services. These

include services for the creation, activation and deactivation of users and roles, the addition and removal of services and rights of access, curriculum management, academic calendar, early warning systems and a calendar of events (eg. upcoming exams, sessions, and important dates).

Currently, the services can be personalized by help of relations specified under services, users and events. Some examples for personalized services are personalized eTesting, personal calendars and schedules, personal student' book [157, 159]. The services are defined and implemented to realize the functional requirements.

5.3 Portal Architecture

The architecture of the educational portal is service-oriented consisting of the following layers [152]:

- *User interface.*
- *E-Services.*
- *Digital libraries.*

The user interface (Figure 27) supports the connection between the users and the portal. Through it, the users can register in the system and create their own personalized educational environment. The user interface visualizes and provides access for the user to services, depending on their role, assigned during the registration.

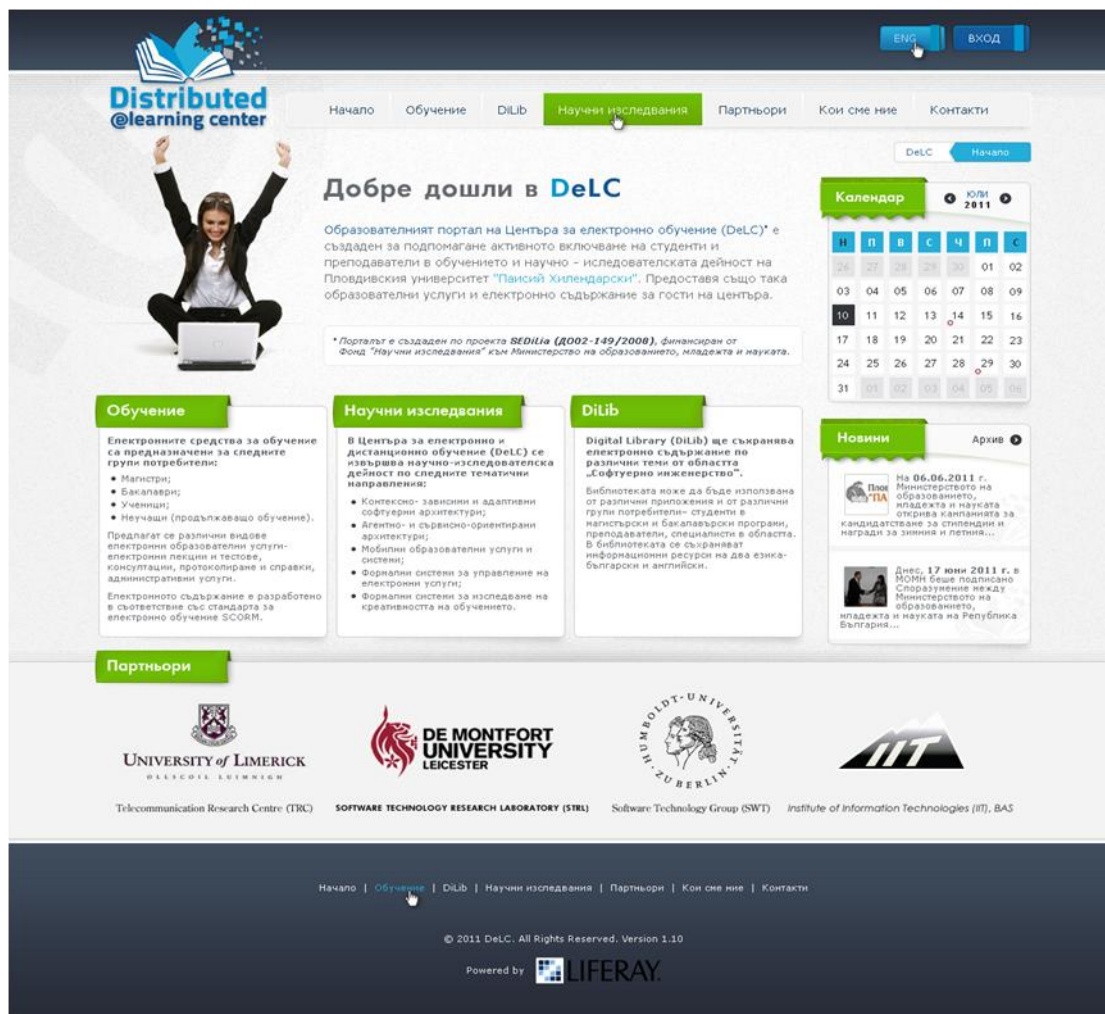


Figure 27: DeLC Portal Home Page

Currently, the next engines are implemented in the portal (Figure 28):

- *SCORM Engine* - SCORM Engine is implemented in the portal architecture for delivering an interpreter of the electronic content, developed in accordance with the SCORM 2004 standard. Currently, the SCORM Engine is an external component integrated in the portal. A new SCORM Engine is under development (s. Chapter 7).
- *Test Engine* - The Test Engine assists in performing electronic testing using the portal. It processes basically the meta-objects, which describe the questions and the patterns of the tests.

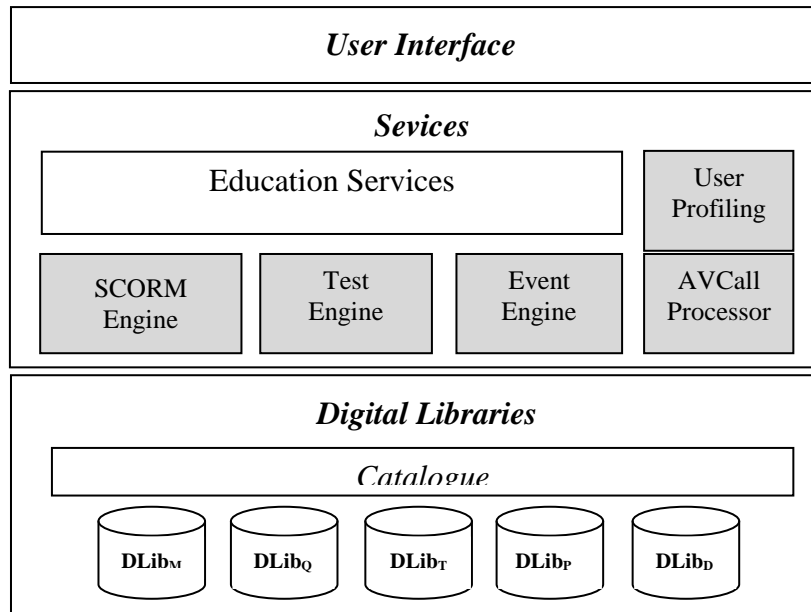


Figure 28: Portal Architecture

- *Event Engine* - The Event Engine supports a model for event management, enabling the users to see and create events and also be notified for them in advance. The events in the system reflect important moments for the users, such as a lecture, examination, test, national holiday, birthday, etc. One event is characterized by attributes, such as a name, start and end date and time, details, and information if it is a recurring one, as well as rules for its recurrence. The Event Engine supports yearly, monthly and weekly recurring.
- *User Profiling* - The User Profiling implements the user model of the portal. The profiles could be classified by roles, user groups, communities, and organizations. The standard user profile consists of three main groups of attributes:
 - *Standard attributes* - necessary for user identification through username, password, e-mail, and others.
 - *Extended attributes* - addresses, phone numbers, Internet pages, IM, social networks contacts, and others.

- *DeLC custom attributes* - other user identifications. Thus, for example, for users with role *student* these can be faculty number, subject, faculty, and course.

The portal gives an opportunity for extending the user profile with some additional attributes. The users' profiles contain the whole information needed for personalization of the provided by DeLC portal services, educational content and user interface. The profile is created automatically during the first user's log in, through a call to the university's database, filling in the standard and custom attributes. The integration with the university database and with other external components is supported by the Integration Engine. Extended attributes are filled by the user. During each next user's log-in in the portal the information in their profile is synchronized, as eventual updates in the university's database are automatically migrated in the user's profile, for example passage in the upper course or changing the subject.

- *AV Call Processor* – an engine which supports the integration between the portal and the Agent Village Node (described in Chapter 6.).

The third layer contains electronic content saved in the digital libraries. The engines have a direct access to the libraries. The needed content can be provided to the education services through the engines. The digital libraries content can be navigated by help of a general catalogue.

5.4 eContent Development Environment

Creating SCORM compliant content by using specialized editors is difficult for many reasons. One of them is that in order to be able to produce non-trivial materials, authors must know the standard in more detail. In order to facilitate content creators in DeLC, an own environment, known as *Selbo2*, was developed [153, 154]. The environment allows for creating new

content, editing available content, as well as generating educational units out of preexisting standardized elements. Major advantage is that the authors work in terms of the discipline, for which the materials are being created. The architecture of the environment is developed so that adaptation for different disciplines is easily achieved.

Selbo2 is a prototype environment which can be applied in schools, universities, distance learning or lifelong learning initiatives. The functionality of the environment is divided between three architectural layers – one client and two server layers, as shown on Figure 29. The first layer (the GUI) consists of client components, which provides the users with graphical user interface for accessing the services of the environment. The second layer (Middleware) consists of server (or business logic) components, which realize the functionality of the environment. The third layer (eLearning resources), also situated in the server part, represents the electronic learning content.

The GUI layer is the visual representation of components of the environment. Content creators interact with the system by this layer. Three types of graphical interfaces are situated (editors, assistants and viewers) which provide the user with interactive tools to manipulate electronic content and its resources. Two types of editors are implemented in the environment:

- *Model (or meta-data) editors* are used by content creators to edit the meta-data for the three models (domain, user and pedagogical). This meta-data can be objects, relations, rules, restrictions, etc. They are used to classify and aggregate knowledge, to navigate electronic content, to present user profiles and pedagogical goals, etc.
- *Content editors* are used to generate and edit electronic educational resources that will comprise the electronic lesson.

Furthermore, two additional types of components are incorporated in the GUI layer:

- *Assistants* are designed to help the content creator in designing the electronic lectures.
- *Viewers* provide only means to visualize resources, but not to edit them.

Examples of such viewers are viewer for predefined resources and ontology visualizer. Content creators are not supposed to modify the predefined resources and the ontologies, because they are created by specialists in the selected domain and provide well balanced set of reusable material and metadata description of notions, terms and concepts with explicitly defined semantics. *SCORM previewer* allows the teacher to preview the generated SCORM lesson. Although it can be regarded as viewer, it is outlined as separate component, because of its significant importance in the lesson creation process. It is invaluable tool, which enables the author to achieve almost WYSWYG style of work with the environment.

The middleware layer consists of server components implementing the main functionality of the environment. The server components are transparent to the user. Interaction with them is performed only through their corresponding graphical user interface component in the client part of the system. *Content editors* are responsible for the creation and manipulation of the individual resources that comprise educational unit. Resource is any multimedia object that can be presented to the learner. Examples of such objects are text file, HTML file, and picture. Selbo2 recognizes some widely spread multimedia formats as resources. Content editors can be put in two distinct groups:

- *First group* consists of editors of standard formats, which are independent of the domain of the educational unit, such as HTML editor, text editor, simple image editor.
- *Second group* consist of specialized editors, specific to a given domain, such as UML editor for Software Engineering, chemical formulae editor for Chemistry, etc.

Currently, Selbo2 is intended for creating materials for Software Engineering. For this reason we are focusing on standard HTML and text editors and specialized editor for creating UML diagrams for Software Engineering. Important role in this architecture play editors of interactive content, which are used to create training materials, and most notably, tests.

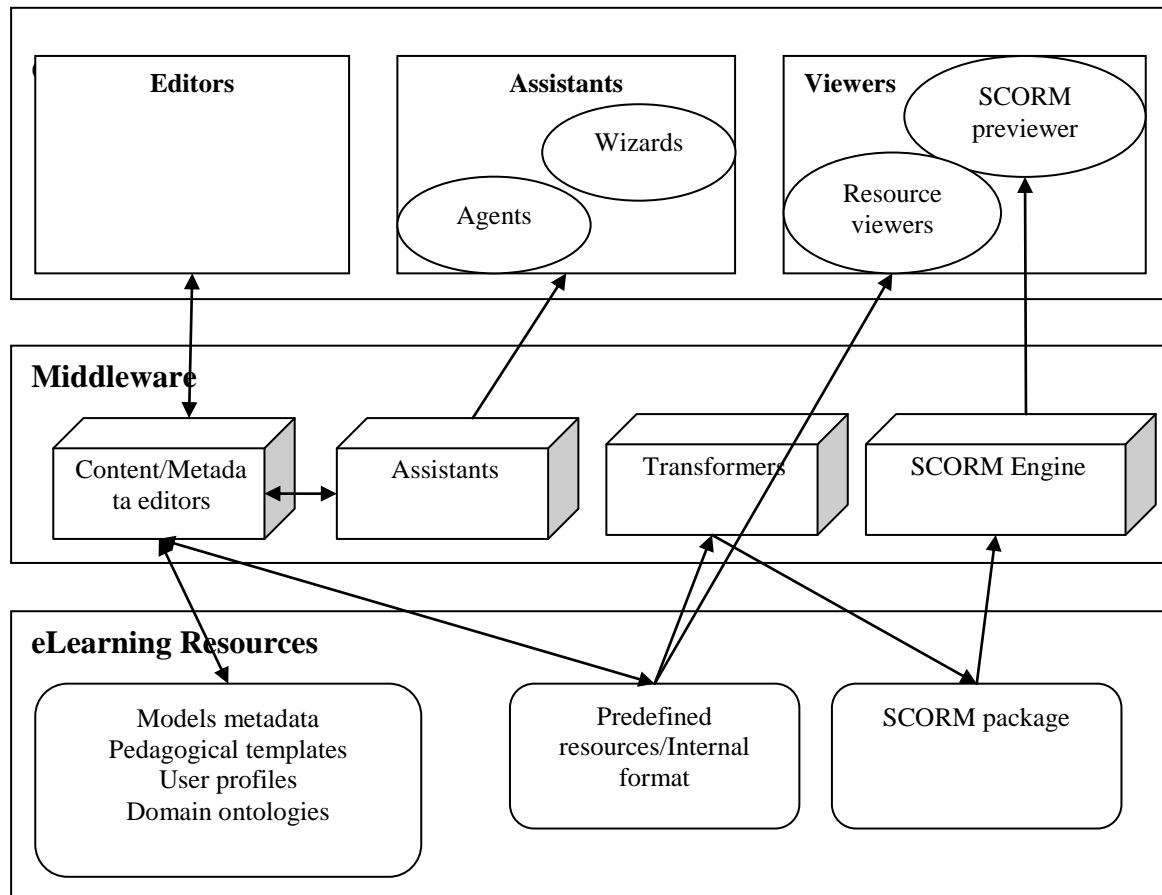


Figure 29: Selbo 2 Architecture

Content editors are implemented as intelligent components, i.e., as combination of GUI *visible editor* and supporting transparent *intelligent agent*.

Resources are the actual pieces that form specific teaching unit. Content creator may choose to create these resources from scratch, or to use as a base some predefined resources. The authors may choose to use the built-in content editors to edit the resources, or use external

applications. Tests are special form of resources. They are interactive units, which provide feedback from the learner to the learning management system. The system uses this feedback information to evaluate the sequencing and navigation rules, defined in the resource organizer. Based on that evaluation, the LMS can choose the most suitable content for the learner. Tests are crucial didactical instrument for control and evaluation of learner's knowledge. Content editors, and more precisely domain specific content editors, maintain the domain model.

Resource organizer is a special type of editor. Instead of modifying resource, that will be used in the lesson, it groups separate (atomic) educational resources into structure, representing single educational unit. Requirements of pedagogical model are defined over this structure, as it stores the rules of sequencing and navigation among individual resources. In Selbo2 the resource organizer is presented as a content tree.

Agents and wizards, referred to collectively as *assistants*, are software entities, which automate the creation of electronic resources. In this architecture assistants play very important role, because they interoperate with all of the components in the system to aid and ease content developer during content creation. Selbo2 uses two kinds of assistants: wizards and agents. *Wizards* are passive assistants; they are called explicitly by the user, and are not active during the whole process of development. Their role is mainly to initialize other components, so they are used in creation of new objects like educational units or different resources. *Agents*, in contrast to wizards, are constantly active assistants. They work autonomously, interoperating with other system components and agents. As a result of this interoperation and agent's goals, they can make decisions and undertake actions, related to developer's work. Selbo2 uses JADE [165] as agent platform. For this reason, Selbo's notion of an agent is restricted by JADE's implementation. For our purposes an agent is entity with

independent control mechanism, capable of exhibiting different behaviors, able to communicate with other agents and GUI components in the environment. Assistants are used to ease user's (content creator's) interaction with the environment; therefore they maintain the user model.

Models govern the behavior of other components in the system. Models are not distinct components of the system. They are rather set of specialized knowledge of the studied discipline (domain model), knowledge of the used pedagogical approach (pedagogical model) and knowledge characterizing the individuality of the students and teachers (user model). This knowledge is used as guidelines in the design and subsequent operation of editors and agents.

Transformers are used to transform the internal structure of the educational unit and its resources into popular standards for electronic education (and vice versa). Selbo2 is oriented primarily in supporting SCORM 2004 standard. The author of the electronic content can use existing electronic material (possibly taken from a repository) as input for the environment, or can create this material from scratch, using the supplied editors. The generated output (through the converters) is SCORM compliant package. The package contains all of the resources of the content, along with the rules for sequencing and navigation amongst these resources, which are defined in the content tree.

SCORM engine is the backend of the SCORM previewer. It is the runtime, which parses the generated SCORM educational unit and interprets its content and educational sequencing and navigational rules.

The data structures layer is formed by the passive elements of the system. It is the data, which editors use and produce. *Pedagogical templates* intend to ease creation of educational unit's structure. Templates define initial layout of the educational unit (content tree). They contain

predefined pedagogical goals and complete set of SCORM sequencing and navigational rules for satisfying these goals. Templates are used only to define the initial layout of the content tree when educational unit is created and are not exported as part of the resulting SCORM package. The author is free to alter the sequencing and navigation rules, or not use templates at all. Set of templates are being developed to be included in the Selbo 2 distribution, but we foresee possibilities for templates to be additionally added to the system, for example to be downloaded from learning management system's portal. Templates, along with Resource organizer, maintain the pedagogical model in the system, because they define the rules that the learner has to follow, in order to achieve the desired pedagogical goal.

User profiles is collective term describing sets of data that contains the preferences of both content creators and students. From the content creator's perspective the user profile contains its personal preferences in working with the environment. Such preferences may include position of editors on screen, favorite editor for specific resources, most recently used resources and ontologies, and main discipline domain. Agents may use this information in order to optimize the system to suit better the teacher's habits of work.

From the student's perspective, the user profiles are set of data describing the intended users of the generated e-learning materials. This data may reflect students' background in the subject, level of interest, learning intensity, available time, etc. Agents can use this information to assist and advise authors during lesson creation process. User profiles, along with component that interpret them, maintaining the user model in the system.

Domain ontologies are specially created ontologies, containing definitions, terms, conceptions and examples, that can be used easily by content developers. Ontology is created specific to a given domain and contains meta-level formal description of semantic connections between its elements. This eases creation of intelligent agents, which can help developer by performing

more meaningful searches. The prototype ontology contains only textual information in the form of definitions from the domain of Software engineering. These definitions can be copied and the pasted in textual content editor. We foresee ontologies that contain not only textual, but binary data like images or specialized formats, which can be added directly as resources in the content tree. Ontology with predefined resources will be included in Selbo 2, but others may be added subsequently, e.g. downloaded from learning management system's portal. Predefined resources, as they are domain specific in their nature, maintain the domain model.

Predefined resources contain ready-made materials, which content developers can use in their educational units. These resources may be available locally, or accessed over network. They can have different granularity: from simple definitions to whole lecture courses, and different structure: from plain text files to ontologies. Examples of predefined resources include HTML pages with explanation of terms, image files containing visual description of a process, sound files with pronunciation of words in foreign language. Resources are multimedia content, which can be added to the electronic lesson with no (or little) need of editing.

SCORM package is the output of the system. It is the electronic educational material that is to be presented to the students. It is generated by the transformers, which convert it from internal representation to selected output format. Currently, Selbo2 supports SCORM 2004 compatibility.

Figure 30 shows the user interface of Selbo2. Content tree editor is at the left side of the image, the ontology viewer is at the right side, and centered is the HTML editor, manipulating Advanced Distributed Learning main page [166]. At the top of the picture is the mascot of the environment - the elephant Agent Selbo.

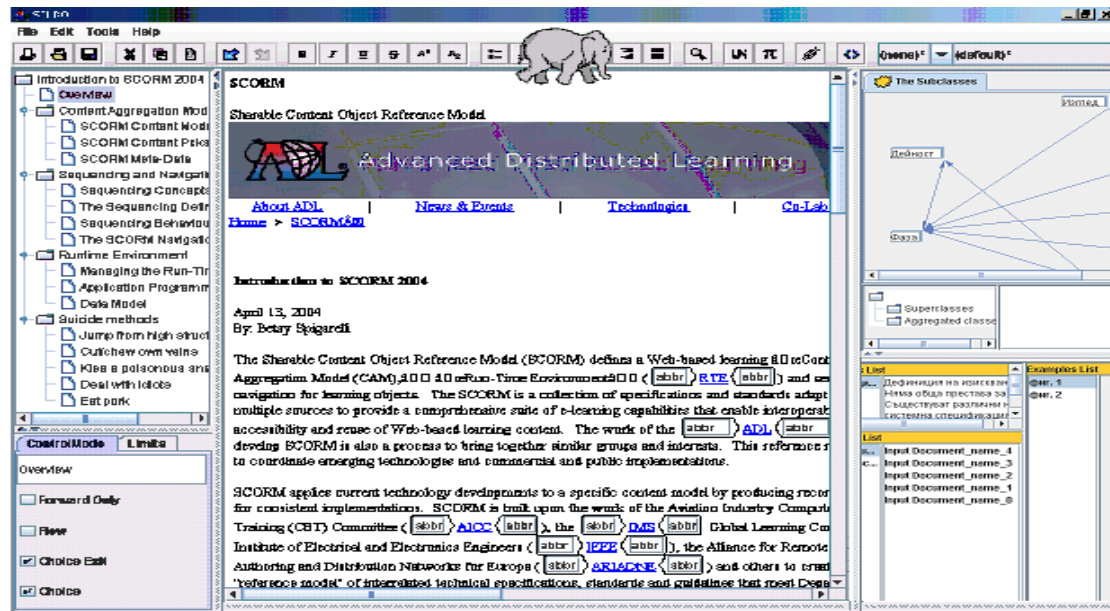


Figure 30: Selbo 2 user interface

Selbo2 is implemented using Java programming language, version 6.0. It uses JADE library to provide agent container for the intelligent agents in the system. All of the agents are managed by a single container, which turns Selbo2 into multi-agent system. It uses Protégé ontology editor to provide means for querying and manipulating domain ontologies. Kafenio is used as a HTML editor library for the HTML resources. Java Plugin Framework (JPF) is used as a plug-in manager. Implementing intelligent components as component of the graphical interface of the environment coupled with specialized agent decreases the complexity of application development. We believe that this approach is more productive than other possible solutions, i.e. creating a sole “super-agent” that collaborates with all of the visual components or development of the agents themselves as visual editors. It also allows us to develop Selbo2 as multi-agent system, featuring several layers of agents, ranging from

agents handling specific editors to agents, governing pedagogical goals over the whole environment.

5.5 Summary

The Education Portal is presented in this chapter. The personalization supported in diverse levels is discussed. The service-oriented architecture of the Educational Portal incorporates the following layers: *User interface*, *E-Services*, *Digital libraries*. Four engines support the selection and the execution of the eLearning services. Teaching material is provided in conformity with SCORM 2004 standard. Own environment, known as *Selbo2*, was developed that allows for creating new content, editing available content, as well as generating educational units out of preexisting standardized elements. The Education Portal has been implemented in the LifeRay framework. The services have been implemented and integrated in the framework in form of Java portlets.

Chapter 6

Education Cluster

Objectives:

- To present the hybrid architecture of the Education Cluster of DeLC
- To present the reactive and proactive service provision in DeLC.
- To present the purely agent-oriented architecture of the Agent Village node.
- To describe four intelligent assistants operating in the Agent Village.
- To present the Refactoring eLearning Environment incorporated in the Agent Village node.
- To describe the interaction between the service-oriented education portal and agent-oriented Agent Village.

6.1 Introduction

This chapter deals with Education Cluster which integrates two nodes – Education Portal and Agent Village. Agent Village node has a poorly agent-oriented architecture where three separate agents and an agent-oriented environment are incorporated. The Agent Village forms a cluster together with the education portal.

The DeLC Education Cluster is described within 4 published papers of which 2 papers are included in the chapter.

6.2 Reactive and Proactive Service Provision in DeLC

In order to provide more effective and personalized user support, we need to enhance the flexibility, reactivity and pro-activeness of the portal including intelligent components into the architecture. The pro-activity improves the usability and friendliness of the system to the users. Pro-activity means that the software can operate "on behalf" of the user" and "activate itself" when it "estimates" that its intervention is necessary. Two approaches are available:

- Direct integration of intelligent components in the currently existing portal architecture.
- Building an education cluster.

The second approach is preferable because it match with DeLC philosophy for building of more complex structures.

The education cluster, presented in this chapter, incorporates two nodes:

- The existing portal – provides the eLearning services.

- Agent Village (AV) - a new agent-oriented node where the "assistants" of the portal services will "live in".

In order to create the architecture of the education cluster three basic problems have to be solved:

- The architecture of the AV node.
- The interaction between the service-oriented portal and agent-oriented AV.
- Intelligent assistance of the portal services.

6.3 Agent Village Node

The Agent Village is developed as purely agent-oriented node with the following multi-layered architecture (Figure 31):

- *System infrastructure* - is built from two different environments (JADE and Eclipse).
- *Assistants* – intelligent agents integrated in the infrastructure. Currently, two kinds of assistants are implemented in the node:
 - *eTesting Assistants* – these agents give assistance to the electronic testing of students performed in the DeLC portal. The eTesting Assistants operate completely over the JADE platform. These assistants can be accessed only through the services located in the DeLC portal.
 - *Refactoring eLearning Environment (ReLE)* – agent-oriented environment assisting eLearning in the field of software technologies. ReLE uses the support of both system environments – JADE and Eclipse. Furthermore, it provides a GUI to the students.

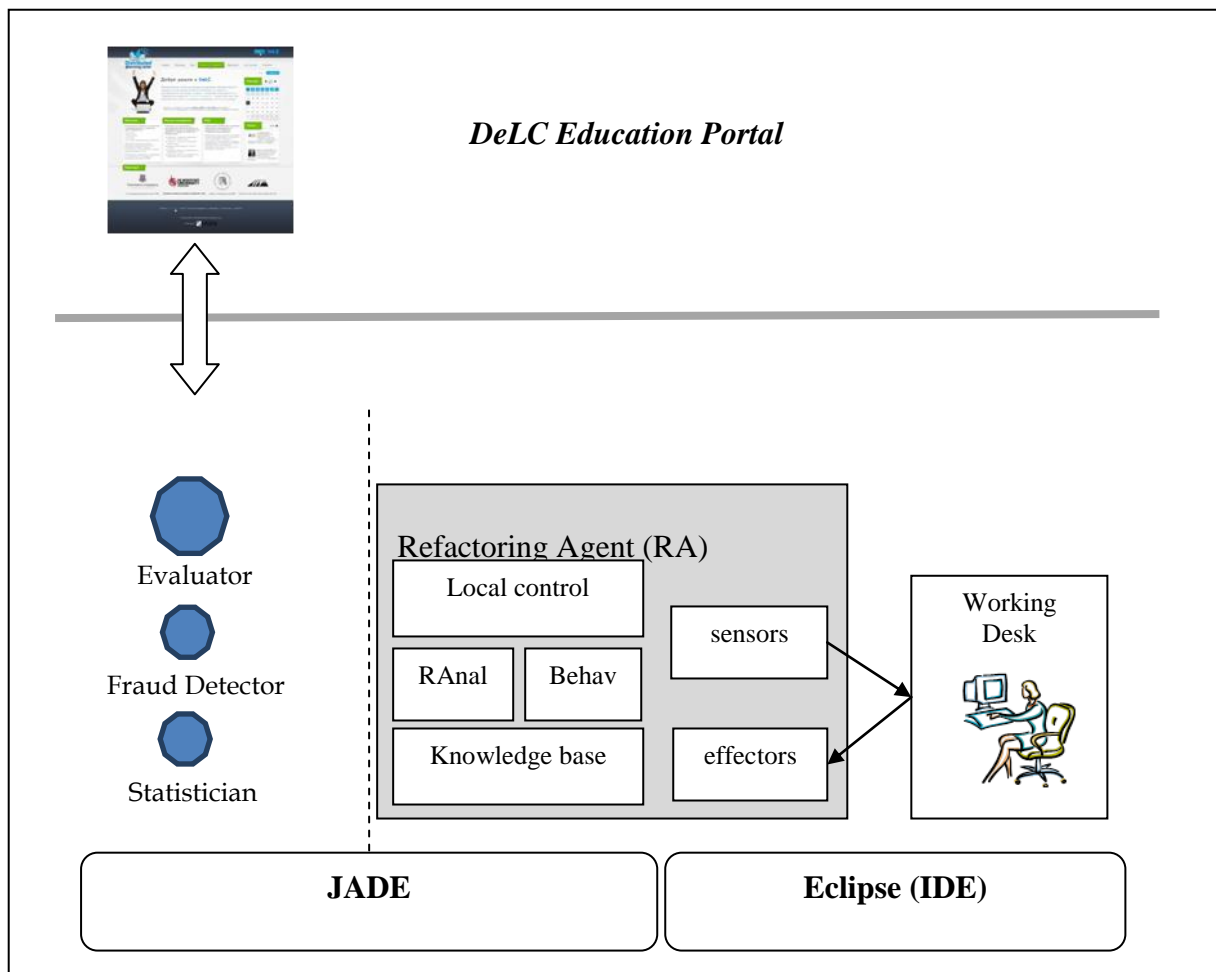


Figure 31: Agent Village Node

6.3.1 System Infrastructure

JADE (Java Agent DEvelopment Framework) [171] is an open source agent-oriented platform compatible with FIPA (Foundation for Intelligent Physical Agents) specification [172]. The platform is composed of containers that manage Java processes providing the JADE run-time and all the services needed for hosting and executing agents [173]. The containers are distributed over the Agent Village node. According FIPA specification, two special agents are automatically supported:

- *The Agent Management System (AMS)* – the agent supervises the platform. In addition it implements the white pages service of the platform.

- *The Directory Facilitator (DF)* – this agent implements a yellow pages service.

The communication within the Agent Village is supported by a Message Transport Service (MTS) provided with the JADE platform.

According the DeLC concept, the assistants have to be proactive as well. In order to achieve this end, we need agents with cognitive abilities, namely autonomy, proactivity and social ability. A widely used way of designing and implementing agents with mental states is the BDI model proposed in [174]. The model states generation of agent's action supported by mental attitudes as beliefs, desires and intentions. In [175] the BDI architecture is presented. In the last years, various agent platforms were proposed that implement the BDI architecture. The platforms JACK [176], Jason [177] and Jadex [178] are based on the Java language. Unfortunately, the standard JADE platform doesn't support a direct implementation of the BDI architecture. Due to this limitation of JADE, in the last year a BDI layer on top of JADE, known as BDI4JADE, was implemented [179, 180]. Recently, the system infrastructure of Agent Village is extended with the BDI4JADE library.

Eclipse [181] is a multi-language software development environment consisting of an Integrated development Environment (IDE) and an extensible plug-in module. Eclipse is free and open source software. In the system infrastructure of the Agent Village a development environment including the Eclipse JDT (Java Development Toolkit) for Java is integrated.

6.3.2 eTesting Assistants

Currently, four assistant are developing which “inhabit” the Agent Village [167, 169]:

- *Evaluator Assistant (EA)* - provides expert assistance to the teacher in assessment of the electronic tests. In the Test Engine, a system service is built for automated assessment of “choice like” questions. In the portal, questions of the “free text” type

are assessed by the examiner and the ratings are entered manually in the service to prepare the final assessment of the test. In the education cluster, however, the Test Engine calls the assistant (an intelligent agent), which makes an “external” assessment of the “free text” type questions.

- *FraudDetector* – tries to recognize any attempts to cheat in the answers given by the students. Such attempts would be to guess the keywords or copy/paste results from Internet search engines. This assistant cooperates with the Evaluator agent and if its receptors detect a probability of a cheating attempt, it informs the Evaluator agent, which for its part send forward a warning message to the examiner.
- *Statistician* - stores information about all processed answers with a full history of the details from all calculating methods used by the Evaluator agent. This assistant needs a feedback how many points are finally given by the teacher for each answer. Thus, it accumulates a knowledge base for each teacher and is able to decide which of the methods best suits the assessment style of the particular teacher. Upon returning the results of the Evaluator assistant, the agent determines which results from each method will be presented as main result, and the results of the other methods will be presented as an alternative. Another feature of this agent is to generate actual statistics on the performance of each of the calculating methods, as the “weakest” of them will go out of service until new and better performing methods are added to the Evaluator agent.

6.3.3 Refactoring eLearning Environment

An eLearning environment for assisting the acquisition of the special technique of refactoring, called Refactoring eLearning Environment (ReLE), is an integral part of the Agent Village node [168, 170]. Although the refactoring process could be realized by hand, the possibility of applying automatic tools is of great importance.

The ReLE architecture consists of two components (Figure 32):

- *Front-end (FE)* – the environment, which is used by the students for the development, compilation and testing of the source code.
- *Back-end (BE)* – the Refactoring Agent (RA), which is an intelligent agent assisting the students during the code development.

The Refactoring Agent is an autonomous software application that continuously analyses and assesses the code that is developed in FE. Consequently, from the RA point of view, FE is its environment. The Refactoring Agent communicates with its environment by means of its sensors and effectors. Via the sensors the RA accesses the complete source code. This implies not only the files being edited, but also the completed ones that have not been opened in the FE for editing. In this way the agent could make a profound analysis and give an adequate assessment for the required changes on the basis of all the code rather than the part that is currently being modified. The sensors also provide some basic metric information to the agent, which is used for initial filtering of the possible refactoring methods that can be further evaluated. The possible metrics are LOC (Line of Code) per class/method, number of methods/attributes per class, and so on. The role of the effectors is to trigger different events that assist the students during the accomplishment of their tasks in FE, where they are working. Such events could be:

- Underlying particular parts of the code by highlighting them with an appropriate colour.
- Displaying messages in dialog windows, balloon messages, etc.
- Emitting sound-signals, vocal messages.
- “Incarnating” the agent in the form of animation to exalt the effect.

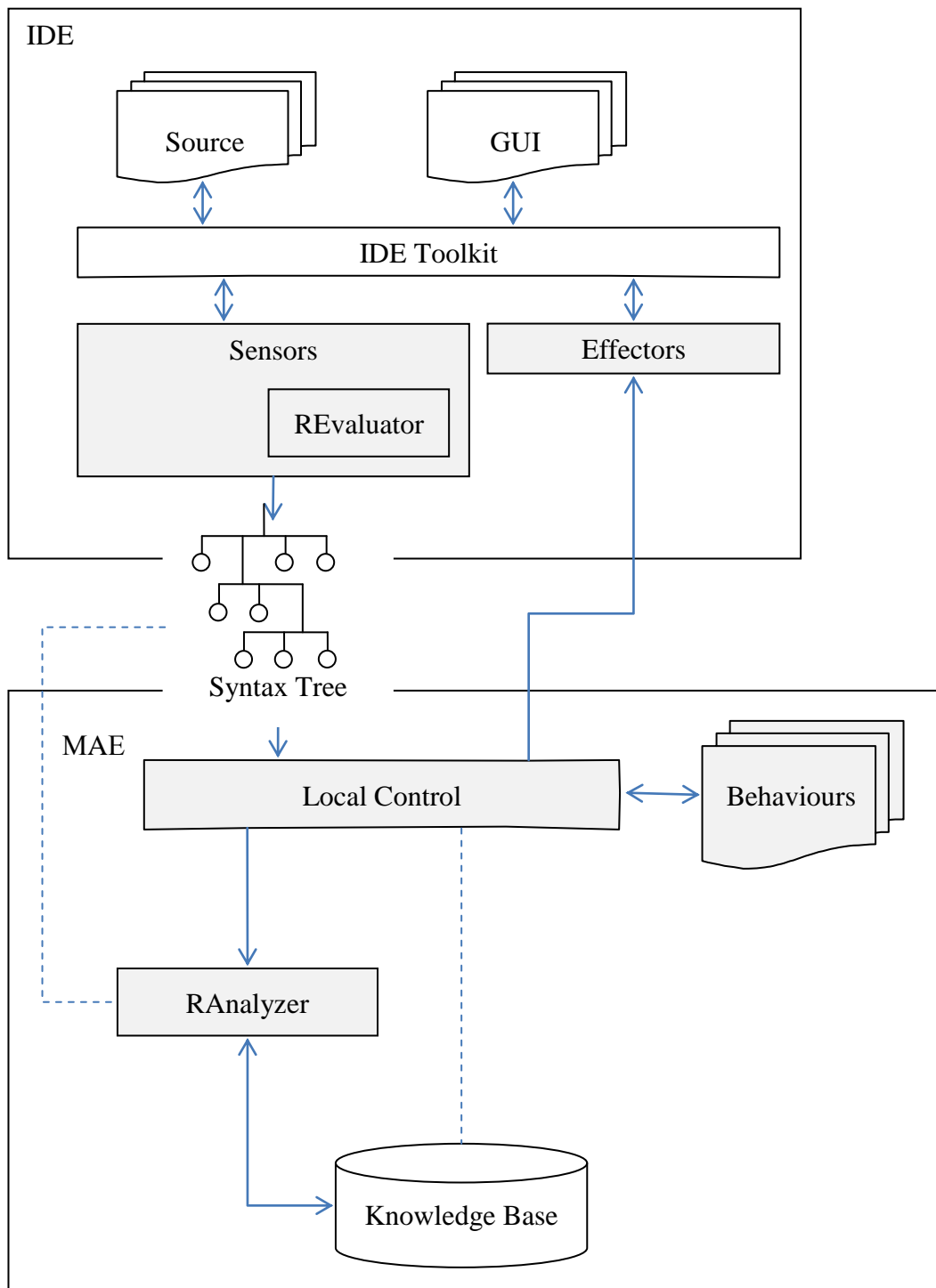


Figure 32: ReLE Architecture

The collaboration of the sensors and effectors is coordinated by the Local Control of the agent, which is based both on the information, incoming from the sensors, and the refactoring rules, stored in the Refactoring Knowledge Base (RKB) of the agent. The analysis of the

source code, written by the students in FE, is made by the RAnalyzer. Before the RAnalyzer starts its work, the RParser parses the source code and creates a tree structure from it. This tree structure can be analysed by the RAnalyzer. The RKB consists of a set of rules together with a set of classes, which build a consistent knowledge base. Each rule describes in a common form the conditions, which allow a particular refactoring method to enter the “short list”, based upon some metrics. In this way, the rules are used by the RAnalyzer in order to make the initial filtering of the refactoring methods, which should be evaluated at the next step.

The kernel of ReLE is an intelligent agent assisting the students in the process of refactoring. Its main task consists in checking the code, which is being developed by the students in FE, and appropriately displaying instructions for improving its quality, whenever necessary. Depending on the refactoring method, which should be applied, the agent could react in three different ways:

- *Automatic Refactoring* - to apply the method automatically after receiving a confirmation from the user.
- *Refactoring Proposal* - to display detailed instructions, explaining to the user where and how the particular refactoring method should be applied.
- *Refactoring Questionnaire* - to ask the user additional questions in order to clarify the conditions and define the appropriate refactoring method.

6.4 Nodes Interaction in the Education Cluster

Agent Village node is implemented as an agent-oriented server, by help of JADE environment. The connection between the educational portal and the AV node is made through the middle layer of the portal architecture, where the electronic services are located.

Depending on the direction of the asked assistance we distinguish *reactive* and *proactive* behaviour of the architecture [184, 185].

In the reactive behaviour, the interaction between the two nodes is initiated by the portal. This is necessary in the cases when a user request is processed and a service needs an "expert" assistance. The problem is that, in their nature, the services are passive and static software modules, intended mainly for the convenient realization and integration of some business functionality. Therefore, they must "transfer" the responsibility for the activation and support of the connection to an active component of the architecture, as agents do. To do this, a specialised engine, known as *AVCallProcessor*, is developed in the portal. *AVCallprocessor* implements an *interaction protocol* (Figure 33) that could be described briefly as follows:

- The central point is that the assistants' environment in Agent Village is considered as an active component.
- In the case of reactive behaviour, the environment "masks" itself as a service – in this way, it can be identified by the *AVCallprocessor* (*AVCallprocessor* is a service; it can't directly identify the agents located in the AV node).
- *AVCallprocessor* sends a request to the agent's environment.
- The environment (as an active component) transforms the request into an ACL-message.
- The generated ACL-message is forwarded to the needed assistant.

The reactive behaviour of the architecture could be implemented using:

- *Synchronous model* - this model is analogous to calling subroutines in programming languages. In this model the service sends a message to AV and waits for the result from the corresponding agent before continuing its execution.

- *Asynchronous model* - in the asynchronous model the interaction is accomplished through some kind of a mechanism for sending and receiving messages. Currently, this model is implemented in the education cluster.

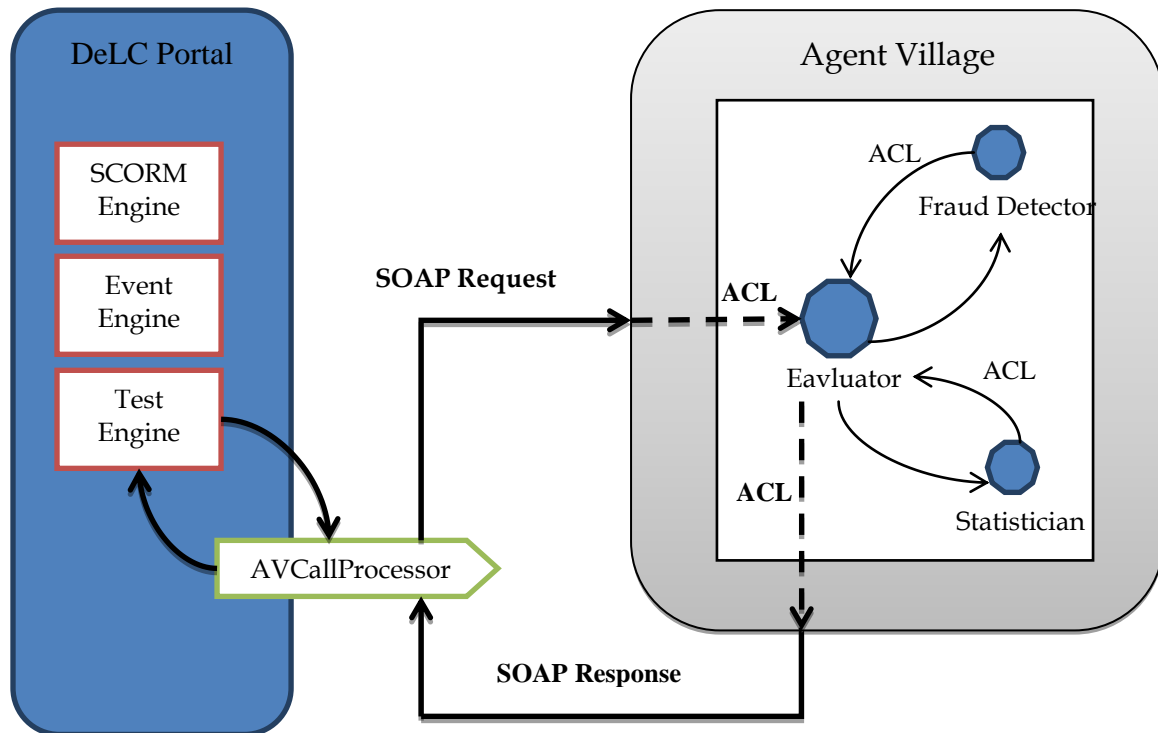


Figure 33: Cluster nodes interaction

In the proactive behaviour (agents work "on behalf of the user"; in this case the user is represented by the portal services), an agent from the AV can determine that in its environment "something is happening" that would be interesting to the services that are assisted by that agent. The agent activates and it can perform certain actions to satisfy the preferences (wishes) of the services. The agent can inform the services (and through the educational portal the user) for its activity.

The difficulties, associated with the management of the pro-activity of our architecture, result from the fact that the portal is designed for reaction of the user's requests. Therefore, the pro-

activity can be managed only asynchronously where the assistants' environment (in Agent Village node) initiates the interaction with the AVCallProcessor (in the education portal) implementing (in opposite direction) the protocol described above. In the proactive behaviour AVCallProcessor operates as a "mailbox" that is periodically checked for incoming messages from the Agent Village node.

6.5 Summary

The Education Cluster is presented in this chapter. Summarizing the chapter, it is important to stress that according to the education cluster architecture, the reactivity and the pro-activity are possible only if the environment of the agents (Agent Village) remains **not more passive**. In order to be identified, the agents need a wrapper (the environment), which "masks" them as a web service [182] to the portal. In such a way, the portal send the request to this service (masked environment), which in its turn transform the request into an ACL message, understandable for the agents. In a similar manner the active environment transform ACL messages into SOAP responses [183] which can be process from the portal services. Currently, three eTesting Assistants and the Refactoring eLearning Environment (ReLE) have been implemented and integrated in the Agent Village.

Chapter 7

Conclusion and Future Work

Objectives:

- Summarise the work in this thesis
- Give a comparative study with other systems
- Give the future work that follows from this thesis

This chapter concludes the thesis by recapitulation of our underlying vision for this work, summarising of the significance of major results and providing an overview of directions for future work.

7.1 Research Summary

This thesis proposed solutions to some shortcomings to current eLearning architectures, as observed in Chapter 2. The work could be summarised as follows:

- An overview of the DeLC architecture, which supports context-aware and adaptable provision of eLearning services and electronic content (Chapter 3). The architecture is fully distributed and integrates service-oriented development with agent technology. Central to this architecture is that a node is our unit of computation (known as E-Learning node) which can have a purely service-oriented architecture, agent architecture or a mixed architecture. In the following chapters, the architectures of the particular eLearning nodes are described.
- The concept of *Mobile eLearning Node* is introduced in Chapter 4. The node uses a three-level communication network, called InfoStations network, supporting mobile service provision. The services, displayed on this node, are to be aware of its context, gather required learning material and adapted to the learner request. This is supported by a multi-layered hybrid (service- and agent-oriented) architecture whose kernel is a middleware. For testing of the middleware a simulation environment has been developed. In addition, the DeLC development approach is proposed.
- Currently, DeLC comprises a second eLearning node implemented as Education Portal (Chapter 5). The architecture of this node is completely service-oriented and it adopts a client-server architecture. In the Education Portal, there are incorporated

education services and *system services*, called engines. The electronic content is kept in Digital Libraries. Besides, in order to facilitate content creators in DeLC, the environment *Selbo2*, was developed. The environment allows for creating new content, editing available content, as well as generating educational units out of preexisting standardized elements. In the last two years, the portal is used in actual education at the Faculty of Mathematics and Informatics, University of Plovdiv.

- The third eLearning node (known as Agent Village), presented in this thesis, exhibits a purely agent-oriented architecture. The purpose of this node is to provide intelligent assistance to the services deployed on the Education Portal. Currently, two kinds of assistants are implemented in the node - *eTesting Assistants* and *Refactoring eLearning Environment (ReLE)*.
- In order to demonstrate the capability of DeLC, a more complex architecture has been developed (known as Education Cluster) and presented in this thesis (Chapter 6). The Education Cluster incorporates two eLearning nodes, namely the Education Portal and the Agent Village. eLearning services and intelligent agents interact in the cluster.

7.2 Related Work

There is a huge amount of starting points that could be investigated in order to improve the current eLearning systems. Recently, ubiquitous and pervasive perspectives of eLearning are delighted at big research interest. Context-awareness and adaptability are often in the focus of ubiquitous and pervasive education scenarios. Towards these scenarios, some basic aspects of the solution of the main research question, proposed in this thesis, could be evaluated comparing with systems presented in the specialized literature. Because of big variety of aspects that could be paid attention to, the examination is focused on architectural point of view.

Articulated arguments about paradigm shifting towards combining of service-oriented and agent-based systems and architectures can be found in [204]. In the last years, this kind of architectures is in focus of intensive research; in the field of eLearning as well. GlobalEdu is a ubiquitous environment that expands the ISAM architecture to eLearning applications [196]. The ISAM project [195] considers that “the computer” is the whole network. GlobalEdu is composed by two kinds of components, namely a Pedagogical Agent and Educational Services (ES) and Support Services (SS). ES supports educational process in the ubiquitous environment where they implement the management of profiles, content and context. SS assists the ES. Furthermore, a context-awareness model has been supported by the architecture [197]. The context management supports personal context, social context and organizational context according the classification given in [192]. The services can be accessed only through the Pedagogical Agent. The FX-Agents project [199] aims to develop new technologies based upon the combination of Web services and techniques from artificial intelligence, especially software agents. The following four primary functions were indicated that need to be automated in building a network of service agents: discovery, service integration, process integration, process control. According the FX-Agents approach, a deployed and discovery technology is proposed based on Business Server Directory (BSD). BSD provide distributed search using logic as “glue” and can be used as a Web service browser [200]. ARISTOTLE architecture [201] uses wireless technologies to support ubiquitous, remote asynchronous mLearning and cooperation among students, faculty, and staff. The services provided in this architecture rely only on users’ devices, making the platform usable even when no other equipment is available. Architecture for an enhanced eLearning system developed within the AdeLE project is presented in [202]. The server component of the architecture is divided into three subsystems: a learning management subsystem, an adaptation subsystem, and a user modeling subsystem. For the purpose of developing a platform-independent and flexible

eLearning system, the design of these subsystems follows a pure service-oriented approach. In [203], system architecture combines synchronous, asynchronous and collaborative services into an integrated platform for meeting the needs of vocational Web-based training. Intelligent agents assist the users only during the synchronous sessions. The OERD is an open platform three-layered architecture that integrates users and learning resources via a communication and management services [194]. The architecture supports: distributed electronic content; exchangeable educational materials such as ebooks, recoded lectures, presentations, lecture notes, case studies, and quizzes, etc.; personalization in a distributed environment; integration of services to provide personalized access to resources in an e-learning network; transparent integration of heterogeneous resource repositories. In [193], a system, called CLUE, is described that consists of the three subsystems for supporting ubiquitous language learning.

Provision of multimedia electronic content from digital libraries would increase immensely the effectiveness of eLearning systems. Digital libraries are an important source for the provision teaching material [186]. However, digital libraries and their standards are developing independently on eLearning applications and their standards and in this way an integration effort is needed. An integrated architecture, known as ASIDE, supporting interoperability between digital libraries and eLearning systems is proposed in [187]. ASIDE, built on the top of digital libraries, supports a service-oriented approach for dealing with reusable learning objects conformed to the IMS Digital Repositories Interoperability Specification [188]. An important component of the architecture is the middleware responsible for the assembly of personalized learning experiences.

In comparison with the investigated solutions the DeLC allows the following benefits to be derived for implementation of eLearning applications:

- Our solution provides a loosely coupled hybrid architecture that give more freedom to build various configurations of the eLearning nodes. There is possible to implement nodes with “pure” architectures (agent-oriented or service-oriented) or with “mixed” architectures (agent- and service- oriented or service- and agent-oriented).
- Context-awareness and adaptation could be made on architectural level. In our architecture, context-aware agents are supported that are responsible for discovering of changes in the environment and generating of new agents in order to adapt the architecture to identified changes.
- The context has been considered and presented in various levels. At the middleware level, the context-awareness is closed related to the four basic scenarios while the domain model, pedagogical model and student model build the context for the provision of educational services. Furthermore, the features of various mobile user devices are presented as profiles according CC/PP specification [144].
- The agents’ environment doesn’t remain passive yet. Service-agent interaction benefits from active behavior of the environment in the Education Cluster.
- The propose architecture stimulate usage of more flexible models for implementation of educational services, such as presented in [141].
- The services in the Education Portal access and use electronic content saved in digital libraries.

7.3 Evaluation

Evaluation of the portal

The eLearning environment DeLC was designed and implemented in the period 2009 - 2011.

Its first run/deployment was in masters' program in software engineering at the Faculty of Mathematics and Informatics of the University in Plovdiv in the academic year 2010/2011. 56 students used it for downloading teaching material, preparing home works, registration and planning of consultations and meetings with the staff. The exams in all subjects were conducted by the help of the environment.

The second run/deployment of the environment was in the academic year 2011/2012. 81 students in the MSc program and about 400 students in the BSc programs in Informatics and Information technologies used the environment. The exams in all subjects in the MSc program were conducted by DeLC. In the BSc program the exams in the following subjects were conducted by the environment: Programming in Java, Introduction in Software Engineering, Introduction in Data Bases, Artificial Intelligence and English for Informatics.

DeLC is deployed on a server located in the Faculty of Mathematics and Informatics. The students can interact with the system through PCs or mobile devices (e.g. laptops). To work with the system the students need passwords. The portal communicates with the university information system (located on a separate server) to verify the identity of the student and if authentication is successful then a unique password has been generated which can be used for the work with the portal.

The tests, generated by the system during exams, are individual for each student. Finished tests are marked automatically. Furthermore, the system generates exam reports and write the marks in electronic student books.

About 3000 tests are solved with the portal in the 2011/2012 academic year. 93% of the tests were completed successfully. The failures are identified in 6% of the remaining tests. The failure isn't identified only in 1 % of the cases yet. This compares favourably to the

traditional methods as it has gained as more efficiency and reduction of staff loading by average of 80 %. The manual creation of 3000 individual tests will be massive task on the staff concerned, in addition to it being error prone. The reason for 1 % failures of the system to automatically generate tests have not be identified. In the forthcoming version of the system this issue will be resolved.

In the University of Plovdiv, the completion of MSc program is possible in two ways: state examination and thesis which is the preferred way. Before using the environment only half of students were able to develop theses due to lack of time. Last year using the environment, 90% of the students completed theses.

Due to flexibility of the DeLC architecture:

- Resources (e.g. time, teaching rooms) can be utilized effectively – for example examinations in different subjects can held at the same time.
- The reusability of the teaching material is increased.
- The overload of the staff is the same, but staff is more flexible and in this way we don't need any more staff.
- The staff is exempt from various routines and can deal with more creative activities.

Evaluation of the cluster

The education cluster is deployed on an agent-oriented server to provide educational services such as teaching material providing, e-testing and so on. The cluster was deployed in academic year 2011/2012. A detailed evaluation of the cluster is fully given in [169], [186] and [191].

7.4 Future Directions

The architecture for context-aware and adaptable provision of educational services and electronic content as presented in this thesis could be improved in all three eLearning applications: the Mobile eLearning Node, the Education Portal and the Education Cluster.

According to the DeLC approach, in the mobile eLearning node, we would like to continue with the implementing the next two *scenario-level-to-middleware-level* iterations, namely ‘Scenario-based Management’ and ‘Adaptation’. In order to support a complete scenario-based management, the middleware has to be enhanced with new agents that are able to identify and locate distributed and time-depended events (e.g., entering/leaving the scope of an InfoStation, changing the user devices). Analysing the identified events, the scenario-based management has to commit the current scenario execution or to initiate a scenario change. Thus we are going to develop a formal model which can be supported by *jTempura* which is implemented in Java during a reengineering process of *Tempura*.

The ‘*Adaptation*’ iteration is concerned with problems related to strengthening the architecture to support adaptability. The existing architecture will be extended with intelligent components providing three models that influence the learning process: the user model, the domain model, and the pedagogical model. We are going to implement these models using agent-oriented approach. In this way, the adaptation potential of the architecture could be enhanced because of being active components (implemented as agents) each model decides for the launching of some adaptation activities own. Furthermore, the interaction among the models and the integration in the middleware or in the Agent City should be facilitated.

The domain model has been implemented in accordance with the SCORM 2004 standard. Currently, the SCORM Engine is an external component integrated in the Education Portal. A

new SCORM Engine with agent-oriented architecture is under development. The educational material would be provided in form of *Intelligent Books (IntelBos)* that should be interactive, proactive and adaptable knowledge repositories. Furthermore, they should deliver the teaching content in a personalized manner; e.g., for each student an IntelBo would create and monitor a personal curriculum in collaboration with pedagogical assistant and with student model assistant.

The pedagogical model will incorporate various *pedagogical assistants* implemented as intelligent agents. The pedagogical assistants, analyzing the current situation, could generate a pedagogical goal that would be suggested to a suitable IntelBo in order to plan the next education session. Thus, a pedagogical-goal-driven education could be managed.

SCORM was proposed as a standard for packaging, delivering and sequencing of sharable learning objects. SCORM doesn't address some important eLearning needs such as assessments, content authorization, collaborative forums, outcomes reporting, and accessibility [205]. In the last years, new standards have been proposed as an extension of SCORM providing more flexibility; e.g., Common Cartridge (CC) [206], Learning Tools Interoperability [207], and Learning Information Services [208]. For this reason a Test Engine was implemented in the Education Portal. We have the intention of developing a new version of the *TestEngine* that will operate as framework supporting a CC compatible questionnaire as plug-ins.

The next future direction is the *New GUI of Selbo 2*. Currently, the functionality of the environment is divided between three architectural layers: one client and two server layers. *Selbo2* supports a plug-in style of extension with new components. In order to improve the user friendliness the client layer will be replaced by various plug-ins which are under construction.

A significant future work, related to the refinement of the Mobile eLearning node, is formalization of the middleware architecture. Some preliminary work has been done in order to specify the InfoStation-based middleware as an ambient system [189, 190, 191, 198].

References

- [1] J. Lu, Da Ruan, G. Zhang, E-Service Intelligence: An Introduction, Studies in Computational Intelligence (SCI) 37, 1-33, 2007.
- [2] J. Lu, D. Ruan, G. Zhang, H-J. Zimmermann, Editorial: A Special Issue on E-Service Intelligence, International Journal of Intelligent Services, Vol 22, 2007, 397-400.
- [3] L. A. Zadeh, Fuzzy sets, Information and Control, 8, 1965, 338-353.
- [4] L. A. Zadeh, From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions, IEEE Transactions on Circuits and Systems, 45, 1999, 105-119.
- [5] L. A. Zadeh, Web Intelligence and Fuzzy Logic – The Concept of Web IQ (WIQ), keynote in Web Intelligence, 2003.
- [6] I. Anagnostopoulos, C. Anagnostopoulos, V. Loumos, E. Kayafas, Classifying Web pages employing a probabilistic neural network, proceedings of IEEE, Vol 151, 2004, 139-150.
- [7] F. Crestani, Comparing neural and probabilistic relevance feedback in an interactive information retrieval system, proceedings of IEEE International Conference on Neural Networks, Orlando, FL, 1994, 3426-3430.
- [8] A.E.Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer, Natural Computing Series, 2007, ISBN: 978-3-540-40184-1.
- [9] N. J. Nilsson, Introduction to Machine Learning, an early draft of a proposed textbook, Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA-94305, 2005.
- [10] J. Han, M. Kamber, Data Mining. Concepts and Techniques, Morgan Kaufmann Publ., 2001.

- [11] D. W. Patterson, Introduction to Artificial Intelligence and Expert Systems, Prentice-Hall, 2007.
- [12] M. Bonett, Personalization of Web Services: Opportunities and Challenges, 2006. <http://www.ariadne.ac.uk/issue28/personalization/> (retrieved December, 2011).
- [13] D. E. Riecken, Personalized views of personalization, Communication of the ACM, Vol. 43. No. 8, 27-28.
- [14] R. R. Yager, targeted E-Commerce marketing Using Fuzzy Intelligent Agents, IEEE Intelligent Systems, Vol. 15, pp.42-45.
- [15] M. Delgado, F. Herrera, et al., A Communication Model Based on the 2-tuple Fuzzy Linguistic Representation for a Distributed Intelligent Agent System on Internet, Soft Computing, 6,2002, 320-328.
- [16] M. Delgado, F. Herrera, et al., Combining Linguistic Information in a Distributed Intelligent Agent Model for Information Gathering on the Internet, Computing with Words, John Wiley & Son, 2001, 251-276.
- [17] E. Herrera-Viedma, F. Herrera, et al., Incorporating Filtering Techniques in a Fuzzy Multi-Agent Model for Gathering of Information on the Web, Fuzzy Sets and Systems, 148, 2004, 61-83.
- [18] R. Kowalczyk, V. Bui, FeNAs: A Fuzzy E-Negotiation Agents Systems , Proceedings of the Conference on Computational Intelligence for Financial Engineering, 2000, NY, 26-29.
- [19] D. Dubois, H. Fargier, H. Prade, Propagation and Satisfaction of Flexible Constraints, Fuzzy Sets, Neural Networks And Soft Computing, 1994, 166-187.
- [20] H. Ouchiya, R. Huang, et al., An Experience-Based Evolutionary Negotiation Model, Proceedings of the Fifth International Conference on Computational Intelligence and Multi-Media Applications, 2003, 212-217.

- [21] A. Toth and P. Toth, "Virtual learning aspects of curriculum development in technical teacher training," in INES'06. Proceedings. International Conference on Intelligent Engineering Systems, London, 2006, pp. 308–313.
- [22] O. Conlan, Multi-Model: Metadata Driven Approach to Personalised e-Learning Services. Doctor of philosophy, University of Dublin, Trinity College, Department of Computer Science, Trinity College, Dublin, 2005.
- [23] A. Chavan and S. Pavri, Open-source learning management with moodle, linux journal, p. 7817, December 2004. p. 7817.
- [24] S. Yuuichi, K. Toshihiro, Y. Seisuke, and N. Hiroshi, Web-based rapid authoring tool for lms quiz creation," in: ITHET'06, IEEE 7th International Conference on Information Technology Based Higher Education and Training, 1-4244-0406-1, Kumamoto University, Ultimo, Australia, 2006, pp. 617–620.
- [25] G. Tortora, M. Sebillo, G. Vitiello, and P. D'Ambrosio, "A multilevel learning management system," in SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering, ACM, New York, NY, USA, 2002, pp. 541–547.
- [26] M. Dougiamas, Moodle, www.MOODLE.org (retrieved December 2011).
- [27] J. Bruce and N. Curson, Uea virtual learning environment, UEA, Product evaluation report, Learning Technology Group, 2001, <http://www.uea.ac.uk/ltg/blackboard/VLEreport.pdf> (retrieved December, 2011).
- [28] A. Dong and H. Li, Ontology-based information integration in virtual learning environment," in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 0-7695-2415-X, Dept. of Comput. Sci., North Dakota State Univ., Fargo, ND, USA, 19-22 Sept 2005, pp. 762– 765.
- [29] S.Graham, S. Simeonov, T. Boubez, G. D. D. Davis, Y. Nakamura, and R.

Neyama, Building Web Services_with Java: Making Sense of XML, SOAP, WSDL and UDDI, 0672321815, Pearson Education, 1st ed., December 2001.

- [30] A. Dong and H. Li., Multimedia access platform for virtual learning environment, in IEEE International Conference on Electro Information Technology, 0-7803-9232-9, North Dakota State Univ., Fargo, ND, USA, 2005.
- [31] S. Dunn, A project for city university centre for hci design, <http://www.saradunn.net/VLEproject/> (retrieved December 2011).
- [32] M. Berry, An investigation of the effectiveness of moodle in primary education, Deputy Head, St. Ives School, Haslemere, 2005. <http://Moodlemoot.org/mod/resource/view.php?id=19> (retrieved December 2011).
- [33] J. Massy, “Study of the e-learning suppliers “market” in europe,” Master ’s thesis, Danish Technological Institute, Heriot-Watt University, 2004.
- [34] M. Zenha-Rela and R. Carvalho, Work in progress: Self evaluation through monitored peer review using the moodle platform, in Frontiers in Education Conference, 36th Annual, 0190-5848, San Diego, CA, Dept. Eng. Informatica, Coimbra Univ, IEEE, 2006, pp. 26–27.
- [35] C. Cheng and J. Yen, Virtual learning environment (vle): a web-based collaborative learning system, In Proceedings of the Thirty-First Hawaii International Conference on System Sciences, Vol. 1, 0-8186-8255-8, Kohala Coast, HI, USA, Univ. of Hong Kong, IEEE, 6-9 Jan 1998, pp. 480–491.
- [36] A. Kumar, R. Pakala, R. Ragade, and J. Wong, “The virtual learning environment system,” in Frontiers in Education Conference, 1998. FIE ’98. 28th Annual, vol. 2 of 0-7803-4762-5, (Tempe, AZ, USA), pp. 711–716, Louisville Univ, IEEE, 4-7 Nov 1998.
- [37] S. Britain and O. Liber, A framework for pedagogical evaluation of virtual

learning environments, JTAP, JISC Technology Applications Programme, University of Wales-Bangor, 1999, <http://www.leeds.ac.uk/educol/documents/00001237.htm>, (retrieved December, 2011).

- [38] J. Lombardi and M. P. McCahill, “Enabling social dimensions of learning through a persistent, unified, massively multi-user, and self-organizing virtual environment,” in C5 '04: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing, (Washington, DC, USA), pp. 166–172, IEEE Computer Society, 2004.
- [39] M. P. McCahill and J. Lombardi, “Design for an extensible croquet-based framework to deliver a persistent, unified, massively multi-user, and self-organizing virtual environment,” in C5 '04: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing, (Washington, DC, USA), pp. 71–77, IEEE Computer Society, 2004.
- [40] S. Graf and B. List, An evaluation of open source e-learning platforms stressing adaptation issues, in ICALT '05: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, (Washington, DC, USA), pp. 163–165, IEEE Computer Society, 2005.
- [41] M. Dougiamas and P. Taylor, Interpretive analysis of an internet-based course constructed using a new courseware tool called moodle, in HERDSA 2002 conference, (Perth, Australia), Curtin University of Technology, 20.
- [42] R. O’Leary and A. Ramsden, “Virtual learning environments.” University of Bristol, 2002. The Handbook for Economics Lecturers.
- [43] Y. Perrie, Virtual learning environment, The Pharmaceutical Journal, vol. 270, pp. 794–795, 2003.
- [44] C. Martin, N. Morris-Cotterill, and M. Smith, Open source software for the

- education market, eduforge, March 2004, <http://eduforge.org> (retrieved December, 2011).
- [45] EduTools, Course management systems, EduTools, 2006, <http://www.edutools.info/static.jsp?pj8&pageHOME> (retrieved December, 2011).
- [46] L. Jolla and H. Motulsky, Graphpad prism, GraphPad Software, Inc., 2008, <http://www.graphpad.com/prism/Prism.htm> (retrieved December, 2011).
- [47] H. Wharekura-tini and kahautu Aotearoa, Technical evaluation of selected learning management systems, Master's thesis, Catalyst IT Limited, Open Polytechnic of New Zealand, 2004.
- [48] N. Sclater, "Moodle : Transforming learning transforming institutions," in MOODLE Regional User Group Conference, 1904811299, The Open University, Packt Publishing, 17th March 2006.
- [49] V. Alvarez, E-learning survey, Department of Computer Science at the University of Oviedo, 2008, <http://www.di.uniovi.es/victoralvarez/survey/> (retrieved December, 2011).
- [50] S. Wilson, Future of VLE - The Visual Version, Scott Wilson's Website, 2005 (retrieved December, 2011).
- [51] [29] E. McMahon, "Future use and development of the 'learnwise' virtual learning environment." BECTA,2001, <http://ferl.becta.org.uk/display.cfm?page=13&resID=1598> (retrieved December, 2011).
- [52] I. Ganchev, D. Meere, S. Stojanov, M. Ó hAodha, M. O'Droma, On InfoStation-Based Mobile Services Support for Library Information Systems, in: Proc. of the 8th IEEE International Conference on Advanced Learning Technologies (IEEE ICALT'08), 2008, pp. 679-681.
- [53] I. Ganchev, S. Stojanov, M. O'Droma, D. Meere, An InfoStation-Based

University Campus System Supporting Intelligent Mobile Services, in: Journal of Computers, Vol. 2, No. 3, Academy Publisher, May, pp. 21-33.

- [54] I.Ganchev, S. Stojanov, M. O'Droma, D. Meere, Communications Scenarios for InfoStation-Based Adaptable Provision of M-Learning Services, in: Proceedings of Second International Conference "Modern (e-) Learning", July, 2007, Varna, pp. 98-104.
- [55] M. Weller, Vle 2.0 and future directions in learning environments, in: Proceedings of The First International LAMS Conference 2006: Designing the Future of Learning, Institute of Educational Technology, Milton Keynes, UK, The Open University, 2006, pp. 99–106.
- [56] J. Malins and I. Pirie, Developing a virtual learning environment for art & design: A constructivist approach, Gray's School of Art, The Robert Gordon University, Aberdeen, 2002.
- [57] C. Chao Su, An open source portal for educators, in: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT '05), Vol. 9, 2005, pp. 961 – 962.
- [58] I.Ganchev, M. O'Droma, D. Meere, M. Ó hAodha, S. Stojanov, "M-learning and m-teaching architectures and the integration of evolving multi-campus educational support e-services." In "M-libraries." Gill Needham and Mohamed Ally, editors. September 2008; 352pp ISBN: 978-1-85604-648-0, pp.159-172.
- [59] Ganchev I., M. O'Droma, D. Meere, S. Stojanov, M. Ó hAodha. 2007. "Integrating the Educational Support Architecture in an E-Services Paradigm: The M-Learning Approach". In Proc. of the 1st International M-Libraries Conference "Information access anywhere, anytime", 19 Pp., 13-14 November, The Open University, Milton Keynes, UK.

- [60] Ganchev I., S. Stoyanov, M. O'Droma. 2005. "Mobile Distributed e-Learning Center". In Proc. of the 5th IEEE International Conference on Advanced Learning Technologies (IEEE ICALT'05), 2 pp., July 5-8, Kaohsiung, Taiwan.
- [61] Ganchev I., S. Stojanov, M. O'Droma, D. Meere. 2007. "Adaptable InfoStation-based mLecture Service Provision within a University Campus". In Proc. of the 7th IEEE International Conference on Advanced Learning Technologies (IEEE ICALT'07), Pp. 165-169, 18-20 July, Niigata, Japan. ISBN 0-7695-2916-X
- [62] Ganchev,I, S. Stojanov, M. O'Droma, D.Meere, An Infostation-Based Multi-Agent System for the Provision of Intelligent Mobile Services in a University Campus Area, 3rd International IEEE "Intelligent Systems", University of Westminster London, UK, September 4th – 6th 2006, pp. 426-432.
- [63] I.Ganchev, M. O'Droma, S. Stojanov, M. hAodha, M-Learning and M-Teaching Architectures and the Integration of E-Services for Educational Support, P. Isaias, P. Kommers, I. Sanchez (eds.), Proceedings of Mobile Learning 2006, iadis, 14-16 July, Dublin, Ireland, pp. 19-25.
- [64] Ganchev, I., S. Stoyanov, M. O'Droma, D.Meere, An Infostation-Based University Campus System for the Provision of mLearning Services, Proceedings of the Sixth International Conference on Advanced Learning Technologies (IEEE ICALT'06), Amsterdam, 2006 IEEE, pp. 195-199.
- [65] Ganchev I., S. Stoyanov, M. O'Droma. 2005. "Consumer-oriented DeLC Service Architecture". In Proc. of the 3rd International Conference on Education and Information Systems, Technologies and Applications (EISTA 2005), vol. 2, Pp.213-218, 14-17 July, Orlando, Florida, USA. ISBN 980-6560-34-5.

- [66] I.Ganchev, S.Stojanov, M.O'Droma, D.Meere, Infostation-Based Adaptable Provision of M-Learning Services: Main Scenarios, International Journal "Information Technologies and Knowledge" Vol.2 / 2008, Pp. 475-482.
- [67] J. Davies, R. Studer, P. Warren, Semantic Web Technologies: Trends and Research in Ontology-based Systems, John Wiley & Sons, 2006.
- [68] EULER Project, <http://www.emis.de/projects/EULER> (retrieved December, 2011).
- [69] Alexandria Digital Library (ADL), <http://www.alexandria.ucsb.edu/>, (retrieved December, 2011).
- [70] PERSIVAL Project, <http://persival.cs.columbia.edu/> (retrieved December, 2011).
- [71] VARIATIONS Project, <http://www.dlib.indiana.edu/variations/> (retrieved December, 2011).
- [72] Perseus Digital Library project, <http://www.perseus.tufts.edu/hopper/> (retrieved December, 2011).
- [73] Google Scholar, <http://scholar.google.com/> (retrieved December, 2011).
- [74] C. L. Borgmann, What can Studies of e-Learning teach us about Collaboration in e-Research? Some Findings from Digital Library Studies, Computer Supported Cooperative Work, 2006, pp. 359-383.
- [75] Y. Mu, X. Wang, Rapid Access to Digital Library Resources: Application of Knowledge Maps in E-Learning Environment in Graduate Education, In Proceedings of 2009 4th International Conference on Computer Science & Education, 2009, pp. 1769-1771.
- [76] M. P. McCahill and J. Lombardi, "Design for an extensible croquet-based framework to deliver a persistent, unified, massively multi-user, and self-organizing

virtual environment,” in C5 '04: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing, (Washington, DC, USA), pp. 71–77, IEEE Computer Society, 2004.

- [77] S. Wilson. Future of VLE - The Visual Version, Scott Wilson's Website, 2005. retrieved April 15, 2007 from <http://www.cetis.ac.uk/members/scott/blogview?entry=20050125170206>.
- [78] H. Zedan, A. Al-Ajlan, S. Stoyanov, K. Bothe, I. Ganchev, M. O'Droma, SEDELia: Software Engineering Digital Library WorkPackage 1: Methodology and Principles of eLearning Systems, STRL, April 19, 2009
- [79] M. Arrigo, M. Gentile, D. Taibi, JELD, the Java Environment for Learning Design, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 9 – 10.
- [80] J2EE, <http://java.sun.com/j2ee/overview.html>, (retrieved March, 2012).
- [81] M. Arrigo, et. Al., mCLT: An Application for Collaborative Learning on a Mobile Telephone, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 11 – 13.
- [82] Brasher, J. Taylor, Development of a research plan for use of ambient technology to test mobile learning theories, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 33 – 37.
- [83] N. Capuano, et al., A system for adaptive platform-independent mobile learning, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 53 – 56.

- [84] Ferscha, C. Holzmann, S. Oppl, Team awareness in personalized learning environments, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 67 – 72.
- [85] Jäppinen, et al., Adaptive mobile learning systems – the essential issues from the design perspective, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 109 – 111.
- [86] F. Cacace, et al., The impact of innovation in medical and nursing training: a Hospital Information System for Students (HISS) accessible through mobile devices, , In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 47 – 52.
- [87] E. Malliou, et al., The MOTFAL project: mobile technologies for ad hoc learning, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 119 – 122.
- [88] T. Monahan, et al., Collaborative m-learning using agents and virtual reality, In: J. Attewell and C. Savill-Smith (eds.), Conference “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 135 – 137.
- [89] C. O’Nuallain, A. Brennan, How can one effectively assess students working in a collaborative mobile environment on an individual basis?, In: J. Attewell and C. Savill-Smith (eds.), “Mobile learning anytime everywhere”, A Book of Papers from MLEARN 2004, pp. 149 – 152.
- [90] P. Lonsdale, et al., A context-awareness architecture for facilitating mobile learning, In: J. Attewell and C. Savill-Smith (eds.), “Learning with mobile devices: research and development”, pp. 79 – 85.

- [91] M. Sharples, Evaluation of a mobile learning organiser and concept-mapping tools, In: J. Attewell and C. Savill-Smith (eds.), "Learning with mobile devices: research and development", pp. 139-144.
- [92] Mobile learning is for everyone, www.m-learning.org, (retrieved March, 2012).
- [93] Ganchev I., S. Stojanov, M. O'Droma, D. Meere, Development of InfoStation-based and Context-aware mLearning System Architectures, In: Advanced Learning. Raquel Hijón-Neira, editor. In-Teh. ISBN: 978-953-307-010-0, 2009 Pp. 115-139.
- [94] Ganchev, M. O'Droma, D. Meere, S.Stoyanov, M. Ó hAodha, On InfoStation-Based Mobile Services Support for Library Information Systems, The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08), 1-5 July 2008, Santander, Spain, Pp. 679-681.
- [95] I.Ganchev, S. Stojanov, D. Meere, M.O'Droma, InfoStation-based mLearning System Architectures: Some Development Aspects, The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08), 1-5 July 2008, Santander, Spain, Pp. 504-505.
- [96] Meere D., I. Ganchev, M. O'Droma, S. Stoyanov, S. Hayes, P. Noonan, Mobile Phones as Tools to Enhance the Traditional Educational Experience, Proc. of the IADIS International Conference on E- Learning 2011 (EL 2011), pp. 5, 22-26 July 2011, Rome, Italy.
- [97] Ganchev I., S. Stojanov, M. O'Droma, D. Meere, An InfoStation-Based University Campus System Supporting Intelligent Mobile Services, Journal of Computers (JCP, ISSN1796-203X), Vol. 2, No. 3, Academy Publisher, May 2007, pp. 21-33.

- [98] S.Stoyanov, I.Popchev, Evolutionary Development of an Infrastructure Supporting the Transition from CBT to e-Learning, Cyberbetics and Information Technologies (CIT), 2/2006, Bulgarian Academy of Sciences, pp. 101-114.
- [99] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, From CBT to e-Learning, Journal "Information Technologies and Control", No. 4/2005, Year III, Pp. 2-10, ISSN 1312-2622
- [100] I. Ganchev, S. Stojanov, M. O'Droma, D. Meere, Adaptable InfoStation-based mLecture Service Provision within a University Campus, In Proc. of the 7th IEEE International Conference on Advanced Learning Technologies (IEEE ICALT'07), 18-20 July, Niigata, Japan, 2007, ISBN 0-7695-2916-X, pp. 165-169.
- [101] I. Ganchev, M. O'Droma, D. Meere, S. Stojanov, M. Ó hAodha, Integrating the Educational Support Architecture in an E-Services Paradigm: The M-Learning Approach". In Proc. of the 1st International M-Libraries Conference "Information access anywhere, anytime", 19 pp., 13-14 November, The Open University, Milton Keynes, UK, 2007.
- [102] I. Ganchev, S. Stojanov, M. O'Droma, D.Meere, An Infostation-Based Multi-Agent System for the Provision of Intelligent Mobile Services in a University Campus Area, 3rd International IEEE "Intelligent Systems", University of Westminster London, UK, September 4th – 6th 2006, pp. 426-432.
- [103] I.Ganchev, M. O'Droma, S. Stojanov, M. hAodha, M-Learning and M-Teaching Architectures and the Integration of E-Services for Educational Support, P. Isaias, P. Kommers, I. Sanchez (eds.), Proceedings of Mobile Learning 2006, iadis, 14-16 July, Dublin, Ireland, pp. 19-25.
- [104] I. Ganchev, S. Stoyanov, M. O'Droma, D.Meere, An Infostation-Based University Campus System for the Provision of mLearning Services, Proceedings of

the Sixth International Conference on Advanced Learning Technologies (IEEE ICALT'06), Amsterdam, 2006 IEEE, pp. 195-199.

- [105] I. Ganchev, S. Stoyanov, M. O'Droma, Consumer-oriented DeLC Service Architecture, In Proc. of the 3rd International Conference on Education and Information Systems, Technologies and Applications (EISTA 2005), Vol. 2, 2005, 14-17 July, Orlando, Florida, USA. ISBN 980-6560-34-5, pp.213-218.
- [106] S. Stoyanov, I. Ganchev, M. O'Droma, E. Doychev, G. Cholakov, Agent-oriented Distributed eLearning Center, Proceedings of the 2th Balcan Conference in Informatics BCI2005, 17-19 November 2005, Ohrid, Macedonia, pp. 238-245.
- [107] I. Ganchev, S. Stoyanov, M. O'Droma, I. Popchev. Enhancement of DeLC for the Provision of Intelligent Mobile Services, In Proc. of the 2nd International IEEE Conference on Intelligent Systems (IS'2004), Vol. 1, 22-24 June, Varna, Bulgaria, 2004, pp. 359-364.
- [108] R. Frenkiel and T. Imielinski, Infostations: The joy of 'many-time, many-where' communications, WINLAB Technical Report,1996.
- [109] O. Rachneva, Distributed eTesting Cluster, PhD Thesis, University of Plovdiv, 2006 (in Bulgarian).
- [110] T. Glushkova, An Adaptive eLearning Environment for Secondary School, PhD Thesis, University of Plovdiv, 2011 (in Bulgarian).
- [111] S. Stoyanov, I. Ganchev, D. Mitev, V. Valkanov, M. O'Droma, Service-oriented and Agent-based Architecture Supporting Adaptable, Scenario-based and Context-aware Provision of Mobile e-Learning Services, International Journal of Computer Information Systems and Industrial Management Applications, Volume 3, Dynamic Publishers, Inc., USA, 2011, ISSN 2150-7988, pp. 771-779.

- [112] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, A. Elci, M. T. Kone, M. A. Orgun (Eds.): "Semantic Agent Systems: Foundations and Applications", Book Series: Studies in Computational Intelligence, Vol. 344, Springer Verlag, 2011, ISBN: 978-3-642-18307-2.
- [113] Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma, An Approach for the Development of a Context-Aware and Adaptive eLearning Middleware, V. Sgurev et al. (Eds.): Intelligent Systems: From Theory to Practice, SCI 299, Springer-Verlag, Berlin-Heidelberg, 2010, ISBN: 978-3-642-13427-2. DOI: 10.1007/978-3-642-13428-9_26. Pp. 519-535.
- [114] S.Stoyanov, I.Ganchev, I.Popchev, I.Dimitrov, Request Globalization in an InfoStation Network, Compt. Rend. Acad. Bulg. Sci., 63, No 6, 2010, Pp.901-908.
- [115] I.Ganchev, S.Stojanov, M.O'Droma, D.Meere, Infostation-Based Adaptable Provision of M-Learning Services: Main Scenarios, International Journal "Information Technologies and Knowledge" Vol.2 / 2008, Pp. 475-482.
- [116] I. Dimitrov, S. Stoyanov, I. Popchev, Optimal Resource Deployment in an InfoStation-Based Network, International Conference REMIA 2010, University of Plovdiv, 2010, pp. 247-253. (to be included – 5 pages)
- [117] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, An Approach for the Development of InfoStation-Based eLearning Architectures, Compt. Rend. Acad. Bulg. Sci., 61, No 9, 2008, Pp.1189-1198.
- [118] S. Stoyanov, V. Valkanova, I. Ganchev, M. O'Droma, An Approach and Architecture Supporting Context-Aware Provision of mLearning Services, IEEE

Second International Conference on Mobile, Hybrid, and On-line Learning, 10-16 February 2010, St. Maarten, pp.11-16.

- [119] S. Stoyanov, I.Ganchev, V.Valkanova, M.O'Droma, Scenario-Oriented and Context-Aware M-Learning System Architecture, IADIS International Conference Mobile Learning 2010, Porto, Portugal, pp. 250-254.
- [120] I.Ganchev, D. Meere, M. O'Droma, S. Stoyanov, Approaches to the Development of an InfoStation-based mLearning System, Proceedings of the 2009 International Conference on E-Learning, E-Business, Enterprise Information Systems, & E-Government (EEE'09), WORLDCOMP'09, July 13-16, 2009, Las Vegas Nevada, USA, pp. 10-15.
- [121] Ganchev I., S. Stojanov, M. O'Droma, D. Meere, Development of InfoStation-based and Context-aware mLearning System Architectures, In: Advanced Learning. Raquel Hijón-Neira, editor. In-Teh. ISBN: 978-953-307-010-0, 2009 Pp. 115-139.
- [122] D. Meere, I. Ganchev, M. O'Dróna, S. Stojanov, V. Valkanova, An Enhanced Context-Sensitive InfoStation-Based mLearning Architecture, 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST '10), 13-16 April 2010, Dubai, UAE.
- [123] S.Stoyanov, I.Ganchev, M.O'Droma, H.Zedan, V.Valkanova, Agent-Oriented Middleware for Mobile eLearning Services, Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC 2009), Seattle Washington USA, 20-24 July, 2009, pp. 62-66.
- [124] S. Stoyanov, I. Ganchev, I.Popchev, M.O'Droma, V. Valkanova, Agent-Oriented Middleware for InfoStation-based mLearning Intelligent Systems, 5th IEEE

International Conference on Intelligent Systems IS'10, 07.07. – 09.07.2010, London,
IEEE Catalog Number: CFP10802-CDR, ISBN:978-1-4244-5164-7, Library of
Congress:2009934065, pp. 91-95.

- [125] D. Meere, I. Ganchev, M. O'Droma, M. O'hAodha, S. Stojanov, "Evolution of
Modern Library Services: The Progression into the Mobile Domain", In M-Libraries
2: A virtual library in everyone's pocket, M. Ally and G. Needham, Eds.: Facet
Publishing, May 2010. ISBN: 978-1-85604-696-1. Pp. 61-72.
- [126] Meere D., I. Ganchev, S. Stojanov, M. O'Droma. 2009. "Adaptation for
Assimilation: The Role of Adaptable M-Learning Services in the Modern Educational
Paradigm," in International Journal "Information Technologies and Knowledge", Vol.
3, Pp. 101-110. ISSN: 1313-0455.
- [127] D. Meere, I. Ganchev, M. O'Droma, S. Stojanov, V. Valkanova, Adaptation
for Assimilation: Shaping Context-Sensitive M-Learning Services within a Multi-
Agent Environment, 2010 Sixth Advanced International Conference on
Telecommunications, 9-15 May 2010, Barcelona, Spain, pp. 74-79.
- [128] I.Ganchev, S. Stoyanov, V. Valkanova, M. O'Droma, Service-oriented and
Agent-based Architecture Supporting Adaptable Context-Aware Provision of Mobile
e-Learning Services, IADIS e-Learning 2010 Conference, Freiburg, Germany, 26 - 29
July 2010, 97-104.
- [129] Ganchev,I., S. Stoyanov, M. O'Droma, V. Valkanova, D. Meere, Pervasive
InfoStation-based mLearning System, ICSNC 2010: The Fifth International
Conference on Systems and Networks Communications, August 22-27, 2010 - Nice,
France, 320-325.

- [130] Ganchev,I., S.Stoyanov, M. O'Droma,V.Valkanova, Context-Aware mLearning Service Execution in an InfoStations Environment, 2009 Fourth International Conference on Internet and Web Applications and Services (ICIW 2009), 25-28 June 2009, Venice/Mestre, Italy, pp. 632-637.
- [131] Ganchev I., M. O'Droma, D. Meere, M. Ó hAodha, S. Stojanov. 2009. "Evolution of Modern Library Services: The Progression into the Mobile Domain". Presented at the 2nd International M-Libraries Conference "Information on the move", Pp. 7, 23-24 June, Vancouver BC, Canada.
- [132] D. Meere, I. Ganchev, S. Stojanov, M. O'Droma, Adaptation for Assimilation: The Role of Adaptable M-Learning Services in the Modern Educational Paradigm, International Book Series "Information Science and Computing", Number 14, 2009, pp. 101-110.
- [133] Ganchev,I., M. O'Droma, D. Meere, S.Stoyanov, M. Ó hAodha, On InfoStation-Based Mobile Services Support for Library Information Systems, The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08), 1-5 July 2008, Santander, Spain, Pp. 679-681.
- [134] I.Ganchev, S. Stojanov, D. Meere, M.O'Droma, InfoStation-based mLearning System Architectures: Some Development Aspects, The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08), 1-5 July 2008, Santander, Spain, Pp. 504-505.
- [135] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, Service-oriented and Agent-based Approach for the Development of InfoStation eLearning Intelligent System

Architectures, IS 2008 – IEEE International Conference on Intelligent Systems Varna, Bulgaria, September 6-8, 2008, pp. 6-20 – 6-25.

- [136] S. Stoyanov, I. Ganchev, M. O'Droma, D. Mitev, I. Minov, Multi-Agent Architecture for Context-Aware mLearning Provision via InfoStations, The 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, pp.549-552, October 28-31, 2008, Paris.
- [137] S. Stoyanov, Context-Aware and Adaptable Architecture CA³, International Conference REMIA 2010, University of Plovdiv, 2010, pp. 97-104.
- [138] Model Driven Architecture, <http://www.omg.org/mda/> (to date)
- [139] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process, Addison-Wesley, 1999
- [140] A. K. Dey and G. D. Abowd, Towards a better understanding of context and context-awareness, in: Workshop on the What, Who, Where, When and How of Context-Awareness, New York, 2000.
- [141] M. Solanki, A. Cau, H. Zedan, Augmenting semantic web service descriptions with compositional specification, In: Proceedings of the 13th international conference on World Wide Web, pp. 544-552, ACM Press, 2004.
- [142] OWL-S: Semantic Markup for Web Services. [Online]. Available - <http://www.w3.org/Submission/OWL-S/> (to date).
- [143] E. Christensen, *et al.*, Web Services Description Language (WSDL) 1.1 [Online], <http://www.w3.org/TR/2001/NOTE> (retrieved November, 2011).
- [144] W3C, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0, World Wide Web Consortium (W3C) (retrieved November, 2011).
- [145] World Wide Web Consortium (W3C), <http://www.w3.org/> (retrieved November, 2011).

- [146] FIPA Specifications, <http://www.fipa.org/repository/aclspecs.html> (retrieved March, 2012).
- [147] UDDI Version 3.0.2, http://www.uddi.org/pubs/uddi_v3.htm (retrieved November, 2011).
- [148] SOAP Version 1.2, <http://www.w3.org/TR/soap/> (retrieved November, 2011).
- [149] W3C, Document Object Model (DOM), <http://www.w3.org/DOM/> (retrieved March, 2012).
- [150] JADE, <http://jade.tilab.com/> (retrieved March, 2012).
- [151] B. Moszkowski, Executing Temporal Logic Programs, Cambridge University Press, Cambridge, England, 1986.
- [152] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, V. Valkanov, A. Stoyanova-Doycheva, V. Valkanova , I. Minov, DeLC Educational Portal, Cybernetics and Information Technologies (CIT), Vol.10, No 3., Bulgarian Academy of Sciences, 2010, pp. 49-69.
- [153] D.Mitev, S. Stoyanov, I. Popchev, Selbo2 – An Environment for Creating Electronic Content in Software Engineering, Cybernetics and Information Technologies (CIT), Vol.9, No 3., Bulgarian Academy of Sciences, 2009, pp. 96-105.
- [154] S.Stoyanov, D.Mitev, I.Minov, T.Glushkova, eLearning Development Environment for Software Engineering Selbo 2, 19th International Conference on Database and Expert Systems Application (DEXA 2008), 1-5 September, Turin, Italy, 2008, Pp. 100-104.
- [155] S. Stoyanov, V. Valkanova, Intelligent Students’ Assistants, International Conference „Key Competences in The Education – Strategies and Practices”, 22-24 September, 2011, Stara Zagora, Bulgaria, ISBN 978-954-691-071-4 (in Bulgarian).

- [156] S. Stoyanov, I. Popchev, M Sandalski, G. Cholakov, An Intelligent eLearning Environment, International Education Conference, Free University of Bourgas, 24-25 June, 2011, Bourgas, Bulgaria, pp. 251-258 (in Bulgarian).
- [157] V. Valkanova, I. Popchev, M. Sandalski, S. Stoyanov, Education Portal for the Secondary School, International Education Conference, Free University of Bourgas, 24-25 June, 2011, Bourgas, Bulgaria, pp. 287-294 (in Bulgarian).
- [158] S. Stoyanov, M Sandalski, I. Popchev, G. Cholakov, E. Doychev, Personalised and Proactive provision of eLearning Services in DeLC Education Portal, 4th National Conference „The Education in the Information Society”, 26-27 May, Plovdiv, Bulgaria, pp. 119-128 (in Bulgarian).
- [159] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, G. Cholakov, DeLC Education Portal Architecture, 4th National Conference „The Education in the Information Society”, 26-27 May, Plovdiv, Bulgaria, pp. 129-138 (in Bulgarian).
- [160] S. Stoyanov, M. Sandalski, V. Valkanova, A. Stoyanova-Doycheva, E. Doychev, An Environment for Provision of eLearning Services, International Conference „E- Distance- ... 21 Century Education”, 6-8 April, 2011, Sofia, Bulgaria, pp. 207-214 (in Bulgarian).
- [161] LIFERAY, <http://www.liferay.com/> (retrieved March, 2012).
- [162] SCORM 2004 4th Edition Specification, <http://www.adlnet.gov/resources> (retrieved March, 2012).
- [163] IMS Simple Sequencing, <http://www.imsglobal.org/simplesequencing/> (retrieved March, 2012).
- [164] Reusable eLearning Object Authoring & Delivery, <http://www.reload.ac.uk/> (retrieved March, 2012).

- [167] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE, A White Paper, exp – Volume 3, no 3., September 2003, pp. 6-19.
- [168] Advanced Distributed Learning, <http://www.adlnet.gov/> (retrieved March, 2012).
- [169] S. Stoyanov, G. Cholakov, V. Valkanova, M. Sandalski, Personalized, Reactive and Proactive Providing of e-Learning Services, EdiLib Conference , Hawaii, 14-17 October, 2011.
- [170] S.Stoyanov, A. Stoyanova-Doycheva, I. Popchev, M. Sandalski, ReLE - A Refactoring Supporting Tool, Compt. Rend. Acad. Bulg. Sci., Tome 64, № 7, 2011, pp. 1017-1026.
- [171] S.Stoyanov, V. Valkanova, G. Cholakov, M. Sandalski, Education Portal for Reactive and Proactive Service Provision, COGNITIVE - IADIS International Conference, 25-30 September, 2011, Rome
- [172] M. Sandalski, A. Stoyanova-Doycheva, I. Popchev, S.Stoyanov, Development of a Refactoring Learning Environment, Cybernetics and Information Technologies (CIT), Vol.11, No ..., Bulgarian Academy of Sciences, 2011, pp. ...
- [173] Java Agent Development Framework, <http://jade.tilab.com/> (retrieved December 2011)
- [174] The Foundation of Intelligent Physical Agents, <http://www.fipa.org/> (retrieved December 2011)
- [175] F. Bellifemine, G. Caire, D. Greenwood, Developing Multi-Agent Systems with JADE, John Wiley & Sons, Ltd, 2007.
- [176] Georgeff, M., Pell, B., Pollack, M., Tambe, M. & Wooldridge, M., The belief-desireintention model of agency, in J. Müller, M. P. Singh & A. S. Rao, eds, 'Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)', Vol. 1555, Springer-Verlag: Heidelberg, Germany,1999, pp. 1-10.

- [177] Rao, A. S. & Georgeff, M. P., BDI-agents: from theory to practice, in: Proceedings of the First Intl. Conference on Multiagent Systems, 1995, San Francisco.
- [178] Howden, N., Rönquist, R., Hodgson, A. & Lucas, A., Jack intelligent agents.: Summary of an agent infrastructure, in 'The Fifth International Conference on Autonomous Agents', Montreal, 2001, Canada.
- [179] Bordini, R. H., Wooldridge, M. & Hübner, J. F., Programming Multi-Agent Systems, in AgentSpeak using Jason (Wiley Series in Agent Technology), 2007, John Wiley & Sons.
- [180] Pokahr, A. & Braubach, L., Jadex user guide, Technical Report 0.96, University of Hamburg, 2007, Hamburg, Germany.
- [181] Pokahr, A., Braubach, L. & Lamersdorf, W., Jadex: Implementing a BDIInfrastructure for JADE Agents, EXP - in search of innovation, 3(3), 2003, 76-85.
- [182] Oliveira de Nunes, C. J. Pereira de Lucena, BDI4JADE: a BDI layer on top of JADE, Monografias em Ciência da Computação, No. 15/10 ISSN: 0103-9741, Editor: Prof. Carlos José Pereira de Lucena, November, 2010.
- [183] Eclipse, <http://www.eclipse.org/> (to date).
- [184] Web Services Addressing, <http://www.w3.org/TR/ws-addr-core/> (retrieved December 2011).
- [185] SOAP Messaging Framework, <http://www.w3.org/TR/soap12-part1/> (retrieved December 2011).
- [186] S. Stoyanov, E. Doychev, V. Valkanova, G. Cholakov, Education Cluster for Intelligent Provision of eLearning Services, DBKDA 2012 : The Fourth International Conference on Advances in Databases, Knowledge, and Data Applications, 29 February – 5 March 2012, Reunion, IARIA, 2012. ISBN: 978-1-61208-185-4, pp. 45-50.

- [187] S. Stoyanov, E. Doychev, A. Stoyanova-Doycheva, V. Valkanova, V. Valkanov, Education Cluster Supporting eTesting and eLearning in Software Engineering, 2nd Annual International Conference on Web Technologies & Internet Applications (WebTech 2012), 7- 8 May 2012, Bali, Indonesia (to print).
- [188] N. McLean, The Ecology of Repository Services: A Cosmic View, Keynote Address, 8th European Digital Library Conference (ECDL), 2004 UK.
- [189] P. Arapi, N. Moumoutzis, S. Christodoulakis, ASIDE: An Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications, Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06), 2006.
- [190] IMS, Digital Libraries Repository Specification, retrieved February 20, 2012 from http://www.imsproject.org/digitalrepositories/driv1p0/imsdri_infov1p0.html.
- [191] S. Stoyanov, H. Zedan, E. Doychev, V. Valkanov, I. Popchev, G. Cholakov, M. Sandalski, Intelligent Distributed eLearning Architecture, In "Intelligent Systems", INTECH Publ., 2012.
- [192] M. H. Al-Sammarraie, Policy-based Approach For Context-aware Systems, PhD Thesis, Software Technology Research Laboratory, De Montfort University Leicester – United Kingdom, July 2011.
- [193] M. Al-Sammarraie, F. Siewe, H. Zedan, Formalising Policies of a mLearning System using CCA, Proceedings of the 4th ACM International Workshop of Context-Awareness Systems for Self-Managing Systems, Copenhagen, Denmark, Sept 26-29 2010.
- [194] A. Schmidt, Potential and Challenges of Context-Awareness for Learning Solutions, Proceedings of 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems (ABIS 05), Saarbruecken, 2005.

- [195] H. Ogata, Y. Yano, How Ubiquitous Computing Can Support Language Learning, Proceedings of the 1st Conference on Knowledge Economy and Development of Science and Technology (KEST 2003), Honjo City, Japan, 2003, pp. 1-6.
- [196] B. Khasawneh, S. A. El-Seoud, Framework for on-demand e-learning resources allocation and distribution: OERAD, International Journal of Computing & Information Sciences Vol. 4, No. 2, August 2006, (On-Line).
- [197] I. Augustin, et al., ISAM, Joing Context-Awareness and Mobility to Building Pervasive Applications, In: I. Mahgoub, M. Ylias (eds.) Mobile Computing Handbook, New York, 2004, pp. 73 – 94.
- [198] D. N. F. Barbosa, et al., GlobalEdu – an Architecture to Support Learning in a Ubiquitous Computing Environment, In: [New Trends and Technologies in Computer-Aided Learning for Computer-Aided Design IFIP International Federation for Information Processing](#), 2005, Volume 192/2005, 1-10, DOI: 10.1007/0-387-30761-3_1.
- [199] C. P. Nino, et al., Context-Aware Model in a Ubiquitous Learning Environment, Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), 2007.
- [200] Al-Sammarraie, M.; Siewe, F.; Zedan, H., Formal specification of an intelligent message notification service in an infostation-based mLearning system using CCA, 2011 International Conference and Workshop on Current Trends in Information Technology (CTIT), 2011.
- [201] C. Petrie, C. Bussler, Service Agents and Virtual Enterprises: A Survey, IEEE Internet Computing, July-August 2003, pp. 2-12
- [202] C. Petrie, M. Genesereth, et al., Adding AI to Web Services, Stanford Logic Group, Stanford University.

- [203] P. Grew, E. Pagani, F. Giudici, From E-Learning to M-Learning: Architectures to Support University Teaching, C. Pahl (ed.): Architecture Solutions for E-Learning Systems, Information Science Reference, Hershey, New York, 2008, pp. 62-79.
- [204] C. Guetl, Moving Towards a Generic, Service-based Architecture for Flexible teaching and Learning Activities, C. Pahl (ed.): Architecture Solutions for E-Learning Systems, Information Science Reference, Hershey, New York, 2008, pp. 1-24.
- [205] C. Bouras, E. Giannaka, T. Tsiatsos, An Integrated Architecture for Supporting Vocational Training, C. Pahl (ed.): Architecture Solutions for E-Learning Systems, Information Science Reference, Hershey, New York, 2008, pp. 197-218.
- [206] M. N. Huhns, Software Development with Objects, Agents, and Services, University of South Carolina, Department of Computer Science and Engineering, Colombia, SC 29208, USA.
- [207] V. Gonzales-Barbone, L. Anido-Rifon, From SCORM to Common Cartridge: A step forward, Computers & Education 54 (2010), pp. 88-102.
- [208] Common Cartridge, IMS Global Learning Consortium, <http://www.imsglobal.org/commoncartridge.html>, (retrieved, March 2012).
- [209] Learning Tools Interoperability, IMS Global Learning Consortium, <http://www.imsglobal.org/toolsinteroperability2.cfm>, (retrieved, March 2012).
- [210] Learning Information Services, IMS Global Learning Consortium, <http://www.imsglobal.org/lis/>, (retrieved, March 2012).
- [211] R. Zhao, C. Zhang, An Ontology-Based Knowledge Management Approach for E-Learning System, International Conference on Management and Service Science (MASS '09), 2009, 1-4.

Appendix: Included Papers

Service-oriented and Agent-based Architecture Supporting Adaptable, Scenario-based and Context-aware Provision of Mobile e-Learning Services

Stanimir Stoyanov¹, Ivan Ganchev², Damyan Mitev¹, Vladimir Valkanov¹, Máirtín O'Droma²

¹ Department of Computer Systems, University of Plovdiv,
236 Bulgaria blvd., Plovdiv 4002, Bulgaria
stani@uni-plovdiv.bg, d.mitev@isy-dc.com, assarel@abv.bg

² Telecommunications Research Centre, University of Limerick,
Plassey, National Technological Park, Castletroy, Limerick, Ireland
{[Ivan.Ganchev](mailto:Ivan.Ganchev@ul.ie); [Mairtin.ODroma](mailto:Mairtin.ODroma@ul.ie)}@ul.ie

Abstract: This paper describes an OMG's MDA-based approach for the development of a service-oriented and agent-based middleware architecture supporting flexible and adaptable, scenario-based and context-aware provision of mobile e-Learning services within InfoStation wireless environments. Considering the system development as a process of iterations, the approach provides an extensive ability to examine different development aspects and extend the system architecture step by step. The first two iterations, namely the base middleware architecture and the scenario-based management, are described in detail. A simulation environment used for testing the architecture is also presented.

Keywords: m-Learning, adaptable software architecture, MDA-based approach, intelligent agents, mobile access, InfoStation environment.

I. Introduction

A distinguishable feature of contemporary mobile e-Learning (m-Learning) systems is the anywhere-anytime-anyhow aspect of delivery of electronic content, which is personalized and customized to suit a particular mobile user [1], [2]. In the light of this, our goal is to develop a software architecture which is able to support m-Learning services delivered within a University campus. The environment uses an InfoStation-based communication environment with distributed control. The InfoStation paradigm is an extension of the wireless Internet, where mobile clients interact directly with Web service providers (i.e. InfoStations). The users request services (by using their mobile devices) from the nearest InfoStation via available Bluetooth (IEEE 802.15 WPAN), WiFi (IEEE 802.11 WLAN), or WiMAX (IEEE 802.16 WMAN) connections. In our approach, each application utilizing the InfoStation infrastructure consists of two components: (i) standardized middleware, which

identifies and locates the changes in the environment in order to prepare the adequate delivery of requested services; and (ii) a set of electronic services (in our case e-Learning services), which are adapted and controlled by this middleware.

In the original InfoStation architecture [3], the InfoStations operate as mediators between the user mobile devices and an InfoStation Center, on which a variety of electronic services are deployed and executed. However in our architecture, the InfoStations are not only mediators but primarily service-providing nodes. The implementation of such architecture raises serious challenges. The main one among these is related to the support of distributed control, which to be able to detect changes in the environment (context-awareness) and according to these changes to offer the requested services in more flexible and efficient way (adaptability).

In the past few years different context-aware systems for different purposes were presented in the literature. An architecture similar to our own architecture is presented in [4]. This context-aware architecture is composed of clients (moving nodes), context-server and middleware (fixed nodes connected through TCP/IP to the context-server). The middleware plays important role both in identifying the clients using the Bluetooth technology and in finding a suitable executable module in accordance with the context acquired from the context-server. Another architecture described in [5] uses a Context Engine for context-aware delivery of Web services by utilizing a rule-based approach based on first-order logic for the centralized processing of context. In [6], a context broker architecture based on ontology for context representation is presented. A context-aware service provider analog to a telephone provider is given in [7]. More common context-aware architectures are presented in [8], [9]. The context in these architectures is stored and processed in a centralized way whereby the middleware is used mainly as a mediator.