

---

---

# Enterprise Integration Opportunities and Challenges

---

---

UNIVERSITY INFORMATION SYSTEMS CASE STUDY

EGYPT, 2013

EDITED BY

**HAITHAM A. EL-GHAREEB**  
*The University of Mansoura  
Egypt*

HELGHAREEB@MANS.EDU.EG  
[HTTP://WWW.HELGHAREEB.ME](http://WWW.HELGHAREEB.ME)  
2013



-= To my Family =-  
*Haitham A. El-Ghareeb, Ph.D.*

March, 2013



# Acknowledgment

First and foremost, I thank ALLAH for giving me incentive to go on and finish this work, and every other work have been finished and giving me glimpses of heavenly hope in the darkest times.

I would like to thank Professor Alaa El-Din Mohamed Riad, Dean of Faculty of Computers and Information Sciences at Mansoura University; at the time of writing these lines, for his unlimited support and encouragement, for providing me with valuable insights and encouraging me to always do better, for always giving me great and valuable ideas to enhance my work.

I would like to thank Dr.Hamy El-Minir, head of communications department of Misr High Institute of Engineering for his support and motivation.

I would like to thank Dr.Hazem M. El-Bakry for the insightful discussions, and ideas, for taking long times to discuss smallest details.

I would like to thank Dr.Ahmed El-Said Hassan for the insightful discussions, ideas, and for taking long times for discussions.

I would like to thank Asma Samir for her contribution in building Sanay3y Application, and her great work in documenting the project.

My Family, you are the people whom without I could not have continued the journey to the day I witness this book alive. Thanks for always being there to support me, every single moment, with everything you can, when I was about to lose hope and trust in me being able to continue through the darkness, trying to reach the goal through long nights of mess, you were there, through long distances you were pushing me forward by your support, encouragement, guidance, and help. Thanks to ALLAH, I am really blessed with amazing family like you. I Love You all!

*Haitham A. El-Ghareeb  
March, 2013*



# List of Figures

1.1	Current View of Zachman Framework [267] . . . . .	10
1.2	Objects encapsulate data and behavior . . . . .	15
1.3	Components encapsulate Objects . . . . .	16
1.4	Infrastructure SOA Building Blocks [159] . . . . .	19
2.1	Enterprise Architecture (EA) Dimensions . . . . .	24
2.2	Enterprise Architecture Classes Utilization . . . . .	25
2.3	Architecture Design is one aspect of System Design . . . . .	26
2.4	Systems non-functional requirements . . . . .	27
2.5	Integration Levels . . . . .	31
2.6	Integration Techniques . . . . .	32
2.7	Pipe-And-Filter Architecture . . . . .	33
2.8	Adopting Standard Data Element Definition Pitfalls . . . . .	34
2.9	Repository Architecture . . . . .	35
2.10	Adopting Databases and Data warehouses Pitfalls . . . . .	37
2.11	Layered Architecture Different Implementations . . . . .	38
2.12	Layered Architectural Model of Client-Server [246] . . . . .	39
2.13	Client - Server Architecture Different Implementations . . . . .	40
2.14	Three Tier Architecture . . . . .	41
2.15	Layered architectural model for client–server application [246] . . . . .	41
2.16	Three Tier Architecture Implementation . . . . .	42
2.17	Three Tier Architecture First Implementation . . . . .	43
2.18	Three Tier Architecture Second Implementation . . . . .	44
2.19	Three Tier Architecture Third Implementation . . . . .	45
2.20	Adopting Standard Enterprise Software Pitfalls . . . . .	45
2.21	Adopting Message Routing Pitfalls . . . . .	47
2.22	Message Bus Architecture . . . . .	49
2.23	Adopting Message CORBA Pitfalls . . . . .	49
2.24	Message Broker Architecture . . . . .	50
2.25	Broker Architecture . . . . .	51
2.26	Adopting Web services Pitfalls . . . . .	52
2.27	Integration Techniques and Software Architectures . . . . .	53
3.1	Flat Multi Agent System (MAS) Architecture . . . . .	60
3.2	Hierarchical MAS Architecture . . . . .	61
3.3	Subsumption MAS Architecture . . . . .	61
3.4	Modular MAS Architecture . . . . .	62

3.5 Basic Webservices' Architecture . . . . .	64
3.6 Service-based in-car Information System (IS) [246] . . . . .	69
4.1 Business Agility Framework Components . . . . .	81
4.2 Proposed Service Oriented Architecture (SOA) and Business Process Management System (BPMS) alignment model . . . . .	88
5.1 Learning Models . . . . .	98
5.2 Prototypical University Management Information System . . . . .	99
5.3 As an umbrella term, "e-Learning" covers many features. . . . .	103
5.4 The Main Learning Management System (LMS) Functionalities . . . . .	103
5.5 LMSs offer a long list of component functionality. . . . .	104
5.6 Detailed IBM Lotus Architecture . . . . .	108
5.7 LMS Evaluation Framework . . . . .	109
5.8 Non-ISO Addressed IS Quality Parameters . . . . .	110
5.9 University Integration Challenges . . . . .	111
5.10 Redundant Student Data . . . . .	114
6.1 Updated Taxonomy of Adaptive Hypermedia Technologies . . . . .	125
6.2 ITS Block Diagram . . . . .	135
6.3 OKI Layered Architecture . . . . .	143
7.1 Course Management System (CMS) Usecase . . . . .	147
7.2 Proposed CMS Architecture . . . . .	149
7.3 Search Process Analysis . . . . .	151
7.4 Manage Rules Process Analysis . . . . .	152
7.5 Discoverer Agent Architecture . . . . .	153
7.6 Course Ranking Process Analysis . . . . .	154
7.7 Imported Courses Tracker Process Analysis . . . . .	156
7.8 Analyzer Agent Process Analysis . . . . .	157
7.9 Search Process Design . . . . .	158
7.10 Tables of Discover/Recommend Service . . . . .	159
7.11 Tables of Check Capability Service . . . . .	159
7.12 Tables of Manage Course Service . . . . .	160
7.13 Table of Pay Service . . . . .	161
7.14 Tables of Raise Exception Search Process Service . . . . .	161
7.15 Manage Rules Process . . . . .	162
7.16 Tables of Manage Rules Process . . . . .	163
7.17 Imported Courses Tracking Required Table . . . . .	164
7.18 Tables of Ranker Software Agent . . . . .	165
8.1 Mobile Services Architecture . . . . .	168
8.2 AMS Use Case . . . . .	168
8.3 AMS Entities . . . . .	169
8.4 Proposed AMS Architecture . . . . .	170
8.5 Take Mobile Assessment Analysis . . . . .	171
8.6 Tracking Process Analysis . . . . .	173
8.7 Take Mobile Assessment Process Design . . . . .	174

8.8	Table of Insert SMS Service . . . . .	175
8.9	Tables of Manage Learner Services . . . . .	175
8.10	Tables of Manage Assessments Services . . . . .	176
8.11	Manage Assessment Items Services . . . . .	176
8.12	Tables of Manage Session Services . . . . .	177
8.13	Table of Insert Exception Service . . . . .	178
8.14	Overview of AMS Database Tables . . . . .	178
8.15	Table of Tracker Agent . . . . .	179
8.16	Service Layer and SMS Manager Class Diagram . . . . .	180
8.17	AMS Class Diagram . . . . .	181
8.18	Mobile Simulator . . . . .	181
8.19	Classical Learner – AMS Mobile Interaction . . . . .	182
9.1	Instructor Challenges between Students and Researchers . . . . .	185
9.2	e-Learning 2.0 Supportive Technologies Online Implementations . . . . .	186
9.3	e-Learning Transitions . . . . .	188
9.4	e-Learning 2.0 Transition in Information Sources . . . . .	189
9.5	e-Learning 2.0 Transition in Feedback . . . . .	189
9.6	e-Learning 2.0 Transition in Information Flow . . . . .	190
9.7	Taxonomy of Adaptive Lecture Technologies . . . . .	196
9.8	Block Diagram of TF Clicker . . . . .	200
9.9	EA Dimensions and Roles . . . . .	204
9.10	Proposed Adaptive Online Lecture Model IT Architecture . . . . .	206
10.1	Current Scenario (Isolated University Management Information System (UMIS) and LMS integrated via users) . . . . .	214
10.2	Presenting a Service Layer between UMIS, LMS, and Users . . . . .	215
10.3	Comparative Network Analysis . . . . .	218
10.4	Insert Performance Measures of the three LIS Architectures . . . . .	219
10.5	Update Performance Measures of the three LIS Architectures . . . . .	220
10.6	Select By ID Performance Measures . . . . .	221
10.7	Display All Performance Analysis . . . . .	221
10.8	Classical Web service Consumption Scenario . . . . .	225
10.9	Load Balanced Web service Consumption Scenario . . . . .	225
11.1	Mobile News Agent Architecture . . . . .	237
11.2	UML Sequence Diagram for Serializing Mobile Objects . . . . .	240
11.3	Hosts' Actions upon Mobile News Agent System (MNAS) Move method invocation . . . . .	241
11.4	Initialization Actions of Mobile News Agent . . . . .	242
11.5	Mobile News Agent's Class Diagrams . . . . .	243
11.6	Mobile News Agent – Approach 1 . . . . .	246
11.7	Mobile News Agent – Approach 2 . . . . .	247
11.8	Parallel Processing Mobile News Agent . . . . .	248
11.9	Effect of Mobile Agent Size on Tc . . . . .	252
11.10	Effect of Total Access Delay on Tc . . . . .	252
11.11	Effect of DBR incremental on TC . . . . .	252

12.1 Sanay3y Context Diagram . . . . .	259
12.2 Sanay3y Use Case . . . . .	260
12.3 Add Craftman Process Activity diagram . . . . .	261
12.4 Main Character in Sanay3y Application . . . . .	262
12.5 Sanay3y Application General Designs . . . . .	263
12.6 Sanay3y Application Layout . . . . .	263
12.7 Real Application Design . . . . .	264
12.8 Sanay3y Application in Action . . . . .	264
12.9 System and Environment Boundary . . . . .	266
12.10 Proposed Web and Mobile Integration . . . . .	267
12.11 Sanay3y Application Architecture . . . . .	267

# List of Tables

2.1	Adopting Standard Data Element Definition Pitfalls . . . . .	34
2.2	Adopting Databases and Data warehouses Pitfalls . . . . .	36
2.3	Adopting Standard Enterprise Software Pitfalls . . . . .	43
2.4	Adopting Message Routing Pitfalls . . . . .	47
2.5	Adopting Message CORBA Pitfalls . . . . .	49
2.6	Adopting Web services Pitfalls . . . . .	51
5.1	Examples of Currently Available LMSs . . . . .	107
9.1	Statistics about Popular Web 2.0 Sites Usage . . . . .	187
9.2	Five fastest growing "member destinations" in the U.S . . . . .	187
9.3	Technology conjunction by Students/Instructors . . . . .	188
9.4	Differences between In-Lecture and After-LectureActivities . . . . .	196
10.1	Advantages and Disadvantages of Parameterized Queries . . . . .	212
10.2	Advantages and Disadvantages of Stored Procedures . . . . .	213
10.3	Insert Operation Measurements Summary . . . . .	219
10.4	Update Operation Measurements Summary . . . . .	219
10.5	Select By ID Operation Measurements Summary . . . . .	220
12.1	Tabular description of the Context Diagram . . . . .	259
12.2	Tabular description of the 'Data Transfer' use case . . . . .	260



# Contents

Acknowledgment	v
Preface	xxi
<b>1 EIS and EA</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Management Support System . . . . .	1
1.2.1 Enterprise Information System . . . . .	6
1.2.2 Group Support Systems . . . . .	6
1.2.3 Knowledge Management System . . . . .	6
1.2.4 Expert Systems . . . . .	6
1.2.5 Hybrid Support Systems . . . . .	7
1.2.6 Emerging Technologies . . . . .	7
1.3 Enterprise Architecture . . . . .	7
1.3.1 TOGAF . . . . .	8
1.3.2 ArchiMate . . . . .	9
1.3.3 Zachman Framework . . . . .	9
1.4 Zachman Framework: Detailed View . . . . .	9
1.4.1 Focus of Columns . . . . .	11
1.4.2 Models of Cells . . . . .	11
1.5 SOA Selected Topics . . . . .	13
1.5.1 SOA Definition . . . . .	13
1.5.2 SOA is an Architectural Style . . . . .	14
1.5.3 Fundamental Constructs of SOA . . . . .	14
1.5.4 Service Attributes . . . . .	16
1.5.5 EAI vs. SOA vs. SOI . . . . .	18
1.5.6 Building Blocks of an SOA Infrastructure . . . . .	18
1.5.7 SOA Role in Embedded or Real-Time Systems . . . . .	19
1.5.8 SOA vs. EDA . . . . .	20
1.6 Summary . . . . .	20
1.7 Review Questions . . . . .	20
1.8 Exercises and Labs . . . . .	21
<b>2 SW Architecture and Integration</b>	<b>23</b>
2.1 Introduction . . . . .	23
2.1.1 Enterprise Architecture Dimensions . . . . .	23
2.1.2 Business Architecture . . . . .	23

2.2	SW Architecture and System Design . . . . .	25
2.3	Importance of SW Architecture . . . . .	26
2.3.1	Distributed Approach Advantages . . . . .	28
2.4	Common SW Architecture Patterns . . . . .	28
2.5	Non-Functional Requirements . . . . .	30
2.5.1	Design Issues of Distributed Systems . . . . .	30
2.5.2	Integration . . . . .	30
2.6	SW Architectures and Integration Techniques . . . . .	31
2.6.1	Data Based Integration Techniques . . . . .	32
2.6.2	SW Based Integration Techniques . . . . .	36
2.7	SOA Selected Topics . . . . .	52
2.7.1	Relationship Between EA and SOA . . . . .	52
2.7.2	Services Identification Methods . . . . .	54
2.7.3	CBA . . . . .	54
2.7.4	Agile/OOAD and SOA . . . . .	55
2.8	Summary . . . . .	55
2.9	Review Questions . . . . .	56
2.10	Exercises and Labs . . . . .	56
<b>3</b>	<b>Service Oriented Architecture</b> . . . . .	<b>57</b>
3.1	Introduction . . . . .	57
3.2	SOA . . . . .	57
3.2.1	SOA Advantages . . . . .	57
3.3	SOA Technologies . . . . .	58
3.3.1	Software Agents . . . . .	59
3.3.2	Webservices . . . . .	63
3.4	Webservices Technology . . . . .	63
3.4.1	Webservices Architecture . . . . .	63
3.4.2	Webservices Key Features . . . . .	65
3.4.3	Webservices Advantages . . . . .	65
3.4.4	Enhanced Standards . . . . .	66
3.4.5	RESTful Webservice . . . . .	67
3.5	An in-car Information System . . . . .	68
3.6	Webservices as SOA enabler . . . . .	68
3.7	Enterprise Service Bus . . . . .	70
3.7.1	ESB . . . . .	70
3.7.2	ESB vs. Integration Technology . . . . .	71
3.7.3	ESB Functionality . . . . .	71
3.7.4	BizTalk Server . . . . .	72
3.8	SOA Selected Topics . . . . .	73
3.8.1	SOA service vs. Webservice . . . . .	73
3.8.2	SOA vs. DCE and CORBA . . . . .	74
3.8.3	SOA vs. CBA . . . . .	74
3.8.4	WSDL vs. REST . . . . .	75
3.8.5	Fundamental Types of Services . . . . .	76
3.8.6	REST . . . . .	76

3.8.7	Web services vs. REST . . . . .	77
3.9	Summary . . . . .	77
3.10	Review Questions . . . . .	77
3.11	Exercises and Labs . . . . .	78
<b>4</b>	<b>BPM and SOA</b>	<b>79</b>
4.1	Introduction . . . . .	79
4.2	Business Agility . . . . .	80
4.2.1	Business Agility Principles . . . . .	80
4.2.2	Business Agility Framework . . . . .	81
4.3	Business Process Agility . . . . .	82
4.3.1	TQM . . . . .	82
4.3.2	BPR . . . . .	82
4.3.3	WfMS and BPR . . . . .	82
4.3.4	BPM . . . . .	83
4.3.5	BPMS . . . . .	83
4.4	Technical Agility . . . . .	84
4.4.1	SW Agents as SOA implementation . . . . .	84
4.4.2	Webservices as SOA enabler . . . . .	86
4.5	Utilizing SOA for BPMS . . . . .	87
4.5.1	Business Layer . . . . .	87
4.5.2	Business Services Layer . . . . .	89
4.5.3	Application Services Layer . . . . .	90
4.6	SOA Selected Topics . . . . .	91
4.6.1	Business Stackholders and SOA . . . . .	91
4.6.2	SOA ROI . . . . .	92
4.6.3	Measuring SOA Effectiveness . . . . .	92
4.6.4	SOA Governance Effectiveness . . . . .	93
4.6.5	Design and Run time SOA Governance . . . . .	93
4.7	Summary . . . . .	94
4.8	Review Questions . . . . .	94
4.9	Exercises and Labs . . . . .	95
<b>5</b>	<b>Integrating UMIS and LMS</b>	<b>97</b>
5.1	Introduction . . . . .	97
5.2	Learning Models . . . . .	98
5.3	UMIS . . . . .	99
5.3.1	Student IS . . . . .	99
5.3.2	Library IS . . . . .	100
5.3.3	Faculty IS . . . . .	100
5.3.4	Finance System . . . . .	100
5.3.5	UMIS Role . . . . .	101
5.4	Prototypical e-Learning . . . . .	101
5.5	LMS and VLE . . . . .	103
5.5.1	Extended LMS . . . . .	105
5.6	UMIS OR LMS . . . . .	106
5.7	IBM Lotus . . . . .	107

5.8	Evaluation of Current LMSs . . . . .	107
5.8.1	Managerial Evaluation . . . . .	109
5.8.2	Pedagogical Evaluation . . . . .	109
5.8.3	IS Evaluation . . . . .	110
5.9	Selected Topics . . . . .	113
5.9.1	SOA and Service Modeling . . . . .	113
5.9.2	Applications and Composite Services . . . . .	114
5.9.3	BPM and Applications . . . . .	114
5.10	Summary . . . . .	115
5.11	Review Questions . . . . .	115
5.12	Exercises and Labs . . . . .	115
<b>6</b>	<b>LMS from Monolithic to Services</b>	<b>117</b>
6.1	Introduction . . . . .	117
6.2	Problem Definition . . . . .	118
6.3	Proposed Solution . . . . .	120
6.4	Adaptive e-Learning Systems . . . . .	120
6.4.1	Adaptive e-Learning Approaches . . . . .	121
6.4.2	AEHS Overview . . . . .	122
6.4.3	Adaptive e-Learning Applications' Examples . . . . .	128
6.5	Intelligent e-Learning Systems . . . . .	128
6.5.1	Evolution of AI in e-Learning . . . . .	129
6.5.2	AI Methods and Techniques . . . . .	129
6.6	SOA in e-Learning Systems . . . . .	136
6.6.1	Webservices as SOA enabler . . . . .	136
6.6.2	SOA and e-Learning Systems . . . . .	137
6.7	Summary . . . . .	144
6.8	Selected Topics . . . . .	145
6.8.1	Deploying Services in the Cloud . . . . .	145
6.8.2	Cloud Computing as SOA Infrastructure . . . . .	145
6.9	Review Questions . . . . .	146
6.10	Exercises and Labs . . . . .	146
<b>7</b>	<b>Course Management Systems</b>	<b>147</b>
7.1	Introduction . . . . .	147
7.2	CMS Analysis . . . . .	147
7.3	Proposed CMS Architecture . . . . .	148
7.3.1	Presentation Layer . . . . .	148
7.3.2	Service Layer . . . . .	148
7.4	Processes Analysis . . . . .	150
7.4.1	Search Process Analysis . . . . .	150
7.4.2	Manage Rules Process Analysis . . . . .	150
7.5	SW Agents Analysis . . . . .	150
7.5.1	Discoverer Agent Analysis . . . . .	150
7.5.2	Ranker Agent Analysis . . . . .	153
7.5.3	Tracker Agent Analysis . . . . .	155
7.5.4	Analyzer Agent Analysis . . . . .	155

7.6	Design of CMS . . . . .	155
7.6.1	Search Process Design . . . . .	155
7.6.2	Manage Rules Process Design . . . . .	161
7.6.3	Software Agents Design . . . . .	163
7.7	Summary . . . . .	164
7.8	Review Questions . . . . .	165
7.9	Exercises and Labs . . . . .	165
<b>8</b>	<b>Mobile Devices in LMS</b>	<b>167</b>
8.1	Introduction . . . . .	167
8.2	AMS Analysis . . . . .	167
8.2.1	Proposed AMS Architecture . . . . .	168
8.2.2	Take Assessment Process Analysis . . . . .	169
8.2.3	Tracker Agent . . . . .	172
8.3	AMS Design . . . . .	172
8.3.1	Take Mobile Assessment Process Design . . . . .	172
8.3.2	Tracker Agent . . . . .	179
8.3.3	AMS Classes . . . . .	179
8.3.4	Mobile Simulator Interface Design . . . . .	179
8.4	Learner interaction with LMS . . . . .	179
8.5	Summary . . . . .	179
8.6	Review Questions . . . . .	182
8.7	Exercises and Labs . . . . .	182
<b>9</b>	<b>Web 2.0 in LMS</b>	<b>183</b>
9.1	Introduction . . . . .	183
9.2	Web 2.0 Technologies . . . . .	184
9.3	e-Learning 2.0 via Web 2.0 . . . . .	188
9.4	Web 2.0 in Learning Institutions . . . . .	190
9.5	Examples of Web 2.0 in Learning . . . . .	191
9.5.1	Summary and Comparison . . . . .	193
9.6	Web 2.0 for Learning Institutions . . . . .	195
9.7	In-Lecture Web 2.0 . . . . .	197
9.7.1	Web Application . . . . .	197
9.7.2	Real-Time Web . . . . .	198
9.7.3	Mobile Applications . . . . .	199
9.8	After Lecture Web 2.0 . . . . .	201
9.8.1	Collaboration Tools . . . . .	201
9.8.2	Social Networks . . . . .	203
9.9	Future of Attended Classes . . . . .	204
9.10	Future Research Directions . . . . .	206
9.11	SOA Selected Topics . . . . .	207
9.11.1	WOA and Web 2.0 Effect on SOA . . . . .	207
9.12	Summary . . . . .	208
9.13	Review Questions . . . . .	208
9.14	Exercises and Labs . . . . .	209

<b>10 Evaluation of SOA in e-Learning</b>	<b>211</b>
10.1 Introduction . . . . .	211
10.2 Current Situation and Solution . . . . .	212
10.2.1 Current IT Infrastructure . . . . .	213
10.2.2 Proposed IT Modifications . . . . .	214
10.3 Evaluation of Solution . . . . .	214
10.3.1 IS Evaluation . . . . .	216
10.3.2 Pedagogical Evaluation . . . . .	226
10.4 SOA Selected Topics . . . . .	227
10.4.1 SOA Governance . . . . .	227
10.4.2 Need of SOA Governance . . . . .	228
10.4.3 Proper Service Granularity . . . . .	228
10.5 Summary . . . . .	228
10.6 Review Questions . . . . .	229
10.7 Exercises and Labs . . . . .	229
<b>11 Mobile Information Agents</b>	<b>231</b>
11.1 Introduction . . . . .	231
11.1.1 Importance of Mobile Information Agents . . . . .	232
11.1.2 Mobile Information Agents in Distributed Systems . . . . .	233
11.1.3 Problem Definition . . . . .	235
11.2 Mobile News Agent System . . . . .	235
11.2.1 Development Framework . . . . .	235
11.2.2 MNAS Components . . . . .	236
11.2.3 Advantages and Disadvantages of MNAS . . . . .	238
11.2.4 Technologies Used in MNAS . . . . .	239
11.3 Analysis and Design of MNAS . . . . .	240
11.4 MNAS Methods . . . . .	243
11.4.1 Methods performed by Mobile News Agent . . . . .	243
11.4.2 Methods performed by Hosts . . . . .	244
11.5 Implementation Approaches of MNAS . . . . .	245
11.5.1 Approach 1 'Single Processing – Holding Data' . . . . .	245
11.5.2 Approach 2 'Single Processing – Delivering Data' . . . . .	245
11.5.3 Approach 3 'Distributed Processing' . . . . .	246
11.6 Performance Measurements . . . . .	248
11.7 Comments on Results . . . . .	250
11.8 SOA Selected Topics . . . . .	251
11.8.1 Information Services . . . . .	251
11.8.2 Information Services Classification . . . . .	253
11.9 Summary . . . . .	253
11.10 Review Questions . . . . .	254
11.11 Exercises and Labs . . . . .	255
<b>12 Mobiles in Community Services</b>	<b>257</b>
12.1 Introduction . . . . .	257
12.1.1 Need for The Project . . . . .	257
12.1.2 Scope . . . . .	258

12.1.3 Aim . . . . .	258
12.1.4 Objectives . . . . .	258
12.1.5 Integrating Web and Mobile . . . . .	258
12.2 Mobile Components . . . . .	259
12.2.1 User Requirements . . . . .	259
12.2.2 User Interaction with Sanay3y . . . . .	260
12.3 Interface Design . . . . .	262
12.4 Integrating Web and Mobile . . . . .	262
12.4.1 Integration . . . . .	265
12.4.2 Sanay3y Application Architecture . . . . .	266
12.5 Summary . . . . .	268
<b>A Resources</b>	<b>271</b>
A.1 Book Resources . . . . .	271
A.2 Learning Management Systems . . . . .	271
<b>B Acronyms and Glossary</b>	<b>273</b>
Acronyms . . . . .	273
Glossary . . . . .	276
<b>Bibliography</b>	<b>300</b>



# Preface

Today, organizations use computer-based IS to better manage operations in the digital world. These organizations use information systems to provide high-quality goods and services as well as to gain or sustain competitive advantage over rivals. In addition to helping organizations to be competitive, information systems have contributed to tremendous societal changes. Chapter 1 presented at page 1 presents different roles of IS as we move into the digital world and how they have helped fuel decision making. We then highlight what information systems are, how Decision Support System (DSS) have evolved to become a vital part of modern organizations, what are the main technologies behind DSS, and why this understanding is necessary to become an effective manager in the digital world. Chapter concludes by discussing EA, its components, and different examples of *defacto* standards.

Technical agility refers to the ability to quickly change the type and flow of information within an enterprise. Technical agility parameters are affected by EA. IT advance has not yet satisfied business requirements due to improper software architectures. SOA addresses technical agility requirements by presenting compositability, modularity, and loose coupling concepts as services that wrap underlying IT infrastructure, databases, and legacy systems and present them via standard interface. There is a need to stabilize IT infrastructure rather than developing new ones and SOA enables this stabilization. Enterprises should balance IT to become better positioned and more agile. Services are the building Blocks of an agile enterprise.

Chapter 2 presented at page 23 discusses the importance of EA in meeting systems non-functional requirements and examines different software architecture paradigms, driving and restraining forces for each one in meeting systems integration as one of the most desirable requirement. Integration has different techniques, and different software architecture paradigms can address different techniques. This chapter presents a proposed mapping of different software architecture paradigms to different integration techniques while highlighting driving forces that encourages software architects to utilize it and restraining forces that discourages architects from utilizing them.

SOA as a design pattern that presents systems as collection of reusable services that can be exposed and consumed on the Internet with standard interfaces has many advantages that can be achieved on technical, managerial, and implementation aspects of the system. Chapter 3 presented at page 57 discusses advantages of SOA, different SOA technologies, focusing on Webservices as main SOA enablers. Chapter discusses Enterprise Service Bus (ESB) and highlighting Microsoft BizTalk Server as one of the major ESB implementations. Chapter concludes with answers of important SOA questions.

Business agility is a new paradigm that is a solution for maintaining enterprise competitive advantage. Business improvement approaches, such as Total Quality Management, Business Process Reengineering, and Workflow Management Systems attempted to satisfy business agility concepts and requirements, but suffered from fatal limitations like lack of concepts definition and measurement, lack of practical implementation, and failure to support ongoing change. The three business improvement approaches realized the need for Business Process Management (BPM) that is the key to business agility. BPMS are ISs that enable implementation of BPM. Current IT infrastructure and ISs architecture do not satisfy BPMS objectives. SOA as a design pattern addresses technical agility that satisfies BPMS objectives in order to achieve business agility.

Chapter 4 presented at page 79 presents a coupling model of BPMS and SOA in order to satisfy process and technical agility aspects of business agility. Proposed model utilizes standards available for mapping BPM concepts via BPMS into SOA, and consists of three layers: Business, Business Services, and Application Services. Business layer enables business executives to handle business processes as information, BPMS resides in this layer. Business services layer is the layer that maps BPM concepts and requirements addressed by BPMS into SOA based IT infrastructure and Enterprise Information System (EIS)s. Application Services layer holds the core services ready to be consumed by different BPMS and shared among enterprises.

Integrating UMIS and LMS can be achieved effectively, efficiently, and with minor modifications of both systems via SOA utilization. SOA can also be useful in integrating long decades and efforts of adaptive and intelligent features of e-Learning in today's ISs. Both UMIS and LMS; as the two major components of learning ISs can make use of adaptive and intelligent features became available over decades. Adaptive and intelligent features in ISs are many, and they can be utilized in new learning systems. Presenting adaptive and intelligent features as services with standard interfaces will allow different e-Learning systems to adopt them, so they will be reusable and newly introduced ISs will not have to redo the work again, besides, wrapping adaptive and intelligent features with standard interfaces will present a separation of interests that help adaptive and intelligent features researchers and developers to focus more on their target and transfer the responsibility of utilizing those features in different ISs to ISs specialists. Today's e-Learning IS architects can make use of SOA in integrating major e-Learning components together, side by side with adaptive and intelligent features to provide students with personalized learning environment.

Chapter 6 presented at page 117 reviews e-Learning different models and the challenges facing e-Learning nowadays, supposing solution to those challenges. Reviewing the e-Learning proposed systems, architectures, and frameworks starting from traditional e-Learning systems through adaptive e-Learning systems supported by intelligent e-Learning features, methods, and techniques yields e-Learning researchers to Services based Adaptive and Intelligent e-Learning Systems.

One of the critical limitations of a newly established educational institution is the lack of available well prepared courses. It is more applicable to use widely available courses that might be higher in quality than preparing new courses. Current CMSs do not exploit courses shareability. Chapter 7 presented at page 147 addresses this

shortage by presenting a CMS to highlight automated discovering and importing of courses maintained and managed by external CMSs. Proposed CMS architecture utilizes SOA design pattern to integrate different CMSs on service level. Proposed CMS consists of two layers: Presentation layer and Service Layer. Presentation layer is responsible for interacting with user either via displaying information or receiving user inputs. Service layer contains core system services. Service layer is divided into three sub layers: orchestration layer, application services layer, and agents layer. Orchestration layer holds business logic required by system processes. Application services layer contains set of stateless services that are capable of performing certain tasks. Agents layer presents the suggested required software agents to serve the overall system. Suggested agents are: Discoverer, Ranker, Tracker, and Analyzer software agents. Integration between software agents and Web services is achieved by utilizing SOA. Proposed CMS facilitates integration between different CMS in order to share resources of educational institutions.

Mobile Learning (M-Learning) is an approach to e-Learning that utilizes mobile devices. LMS should enable M-Learning. Unfortunately, M-Learning is not the same at each educational institution. Assessment is one of the learning activities that can be achieved electronically and via mobile device. Mobile assessment relies on services that are not part of the LMS. Integrating different external systems and services to be virtually part of LMS is one of integration challenges. Chapter 8 presented at page 167 presents a SOA to integrate mobile assessment in LMS. Proposed Architecture consists of two layers: Interface layer, and Service layer. Interface layer interacts with instructors and learners via portals, and with external organization services via Web services. Service layer contains core system services and has three sub layers: Orchestration, Application Services, and Agents layer. Orchestration layer is the layer that holds business logic required by system processes. Application Services layer contains set of stateless services that are capable of performing certain tasks. Agents layer presents the suggested required software agents to serve the overall system. Suggested agents are Analyzer and Tracker. Analyzer is required to analyze students assessments to detect learning deficiencies. Tracker agent tracks students to remind with missing assessments. SOA facilitated integration of software agents within LMS, and implementation of new processes to support learning.

Several types of Information and Communication Technology "ICT" have been utilized in the learning process to overcome time and place challenges. Learning delivery models are: traditional, distance, and hybrid learning. Hybrid learning attempts to maintain the best of traditional learning and provides the hopes and objectives of distance learning in a model that maintains the learning process on the right road. The widespread of Web 2.0; the Internet created by collaborative activities of different users resulted in the appearance of the acronym (e-Learning 2.0). Web 2.0 is a big resource that changed the way everyone around thinks about and accesses the Internet, and greatly will touch the coming generations; the generations that we are currently presenting education to. e-Learning 2.0 is supposed to make use of different Web 2.0 capabilities. From our perspective, e-Learning 2.0 shall be introduced into the lecture itself. This chapter presents taxonomy of Web 2.0 technologies, highlighting how, and when they can be used within learning institutions in enhancing the learning process. Web 2.0 technologies can be classified

into In-Lecture and After-Lecture technologies based on their capabilities to support required activities for each class. Chapter 9 presented at page 183 highlights different studies that show the importance and need from both instructors and students to utilize Web 2.0 technologies in learning institutions, presents different ways to use Web 2.0 technologies, and concludes with future research directions about Web 2.0 adoption in learning institutions. SOA is a design pattern that presents infrastructure architecture, information architecture, and software architecture as composable services that wrap legacy systems and can be utilized in new systems. This chapter highlights the importance of SOA in adopting Web 2.0 technologies in current lecture model.

To help universities achieve their goals, it is important to align managerial functionality side by side with educational aspects. Universities consume University Management Information Systems (UMIS) to handle managerial aspects as they do with LMSs (LMS) to achieve learning objectives. UMIS advances LMS by decades and has reached stable and mature consistency level. LMS is the newly acquired solution in Universities; compared to UMIS, and so adopting LMSs in universities can be achieved via three different deployment approaches. First approach believes in LMS ability to replace UMIS and performing its functionality. Second approach presents the idea of extending UMIS to include LMS functionality. Third approach arises from the shortages of the two proposed approaches and present integration between both as the appropriate deployment approach. SOA is a design pattern that can be used as a suitable architectural solution to align UMIS and LMS. SOA can be utilized in universities to overcome some of IS challenges like the integration between UMIS and LMS. Chapter 10 presented at page 211 presents the current situation at Mansoura University; Egypt, presents integration as the most suitable solution, and evaluates three different implementation techniques: Dynamic Query, Stored Procedure, and Web services. Evaluation concludes that though SOA enhanced many different aspects of both UMIS and LMS; and consequently university overall. It is not recommended to adopt SOA via Web services as the building unit of the system, but as the interdisciplinary interface between systems.

Mobile information agents are mobile agents that retrieve information to end users. Chapter 11 presented at page 231 presents a new architecture for MNAS (MNAS) that is based on a mobile information agents to retrieve news to users from different news servers. Several approaches for implementing MNAS are illustrated and comparative analysis is performed to identify the most suitable one. MNAS components are: Mobile News Agent that retrieves news to user, Directory Server that is the central manager server, and News Servers that hold different news categories and contents. MNAS provides the client an easy to use interface to specify preferred news categories, a client application that initiates the mobile news agent which heads directly to the directory server to retrieves list of news servers that contain news matches user preferences, and searches all news servers and collects news to finally return to client. MNAS proves that mobile information agent technology can not just be used in all information retrieval systems, because under some conditions, it might loose some of its advantages. MNAS tries to find out the most appropriate information retrieval systems conditions to implement mobile information agent technology..

# Chapter 1

## EIS and EA

### 1.1 Introduction

Today, organizations use computer-based IS to better manage operations in the digital world. These organizations use information systems to provide high-quality goods and services as well as to gain or sustain competitive advantage over rivals. In addition to helping organizations to be competitive, information systems have contributed to tremendous societal changes. An IS can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to supporting decision making, coordination, and control, information systems may also help managers and workers analyze problems, visualize complex subjects, and create new products [45]. IS contain information about significant people, places, and things within the organization or in the environment surrounding it. By information we mean data that have been shaped into a form that is meaningful and useful to human beings. Data, in contrast, are streams of raw facts representing events occurring in organizations or the physical environment before they have been organized and arranged into a form that people can understand and use.

This chapter presents different roles of IS as we move into the digital world and how they have helped fuel decision making. We then highlight what information systems are, how DSS have evolved to become a vital part of modern organizations, what are the main technologies behind DSS, and why this understanding is necessary to become an effective manager in the digital world. Chapter concludes by discussing EA, its components, and different examples of *defacto* standards.

### 1.2 Management Support System

Management Support System (MSS) is the support of management tasks by the application of technologies. Sometimes called DSSs or Business Intelligence (BI) [255]. MSS include:

- DSS [14]: is a computer-based IS that supports business or organizational decision-making activities. DSSs serve the management, operations, and planning levels of an organization and help to make decisions, which may be rapidly

changing and not easily specified in advance. DSSs can be either fully computerized, human or a combination of both.

- Business Analytics (BA) [4]: refers to the skills, technologies, applications and practices for continuous iterative exploration and investigation of past business performance to gain insight and drive business planning. BA focuses on developing new insights and understanding of business performance based on data and statistical methods. In contrast, BI traditionally focuses on using a consistent set of metrics to both measure past performance and guide business planning, which is also based on data and statistical methods. BA makes extensive use of data, statistical and quantitative analysis, explanatory and predictive modeling, and fact-based management to drive decision making. Analytic may be used as input for human decisions or may drive fully automated decisions. BI is querying, reporting, Online Analytical Processing (OLAP), and "alerts".
- Data Mining (DM)[12]: is the process that attempts to discover patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and Data Base (DB) systems. The overall goal of the DM process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves DB and data management aspects, data preprocessing, model and inference considerations, interesting metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.
- Data Warehouse (DW) [13]: is a DB used for reporting and data analysis. It is a central repository of data which is created by integrating data from multiple disparate sources. Data warehouses store current as well as historical data and are used for creating trending reports for senior management reporting such as annual and quarterly comparisons. The data stored in the warehouse are uploaded from the operational systems (such as marketing, sales etc.). The data may pass through an operational data store for additional operations before they are used in the DW for reporting.
- BI [5]: is the ability of an organization to collect, maintain, and organize data. This produces large amounts of information that can help develop new opportunities. Identifying these opportunities, and implementing an effective strategy, can provide a competitive market advantage and long-term stability. BI technologies provide historical, current and predictive views of business operations. Common functions of BI technologies are reporting, online analytical processing, analytic, DM, process mining, complex event processing, business performance management, benchmarking, text mining, predictive analytic and prescriptive analytic. The goal of modern BI deployments is to support better business decision-making.
- OLAP [36]: is an approach to answering multi-dimensional analytical (MDA) queries swiftly. OLAP is part of the broader category of BI, which also encompasses relational DB report writing and DM. Typical applications of OLAP

include business reporting for sales, marketing, management reporting, BPM, budgeting and forecasting, financial reporting and similar areas, with new applications coming up, such as agriculture. The term OLAP was created as a slight modification of the traditional DB term OLTP (Online Transaction Processing). OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives. OLAP consists of three basic analytical operations: consolidation (roll-up), drill-down, and slicing and dicing. Consolidation involves the aggregation of data that can be accumulated and computed in one or more dimensions. For example, all sales offices are rolled up to the sales department or sales division to anticipate sales trends. By contrast, the drill-down is a technique that allows users to navigate through the details. For instance, users can view the sales by individual products that make up a regions sales. Slicing and dicing is a feature whereby users can take out (slicing) a specific set of data of the OLAP cube and view (dicing) the slices from different viewpoints.

- Computer assisted Systems Engineering (CASE) [8]: is the scientific application of a set of tools and methods to a software system which results in high-quality, defect-free, and maintainable software products. It also refers to methods for the development of IS together with automated tools that can be used in the software development process.
- Group Support System (GSS) [255] Facilitates decisions made by groups within the organization. Traditional meeting can last a long time and money. Collaborative computing systems, groupware, electronic meeting can run over the Web and provide video conferencing, audio conferencing in addition to meeting tools like electronic brainstorming, voting and document sharing.
- EIS[19]: is generally any kind of computing system that is of "Enterprise class". This means typically offering high quality of service, dealing with large volumes of data and capable of supporting some large organization ("an enterprise"). EISs provide a technology platform that enables organizations to integrate and coordinate their business processes. An EIS provides a single system that is central to the organization and that ensures information can be shared across all functional levels and management hierarchies. Enterprise systems create a standard data structure and are invaluable in eliminating the problem of information fragmentation caused by multiple information systems within an organization. A typical EIS would be housed in one or more data centers, would run enterprise software, and could include applications that typically cross organizational borders such as content management systems. The word enterprise can have various connotations. Frequently the term is used only to refer to very large organizations. However, the term may be used to mean virtually anything, by virtue of it having become the latest corporate-speak buzzword.
- Enterprise Information Portal (EIP) [18]: is a framework for integrating information, people and processes across organizational boundaries. It provides a secure unified access point, often in the form of a web-based user interface,

and is designed to aggregate and personalize information through application-specific portlets. One hallmark of enterprise portals is the de-centralized content contribution and content management, which keeps the information always updated.

- Enterprise Resource Management (ERM) [20]: describes software that lets an enterprise manage user access to its network resources efficiently. ERM software generally lets a user sign on to different enterprise systems and applications using the same password. ERM software makes it easy for the enterprise to control and keep track of which systems and resources each user has access to, and provides consistent standards for creating and changing passwords. One system administrator can usually manage user access to all platforms and to the applications on these platforms that require controlled access.
- Enterprise Resource Planning (ERP) [21]: systems integrate internal and external management information across an entire organization, embracing finance/accounting, manufacturing, sales and service, Customer Relation Management (CRM), etc. ERP systems automate this activity with an integrated software application. The purpose of ERP is to facilitate the flow of information between all business functions inside the boundaries of the organization and manage the connections to outside stakeholders. ERP systems can run on a variety of computer hardware and network configurations, typically employing a DB as a repository for information.
- CRM [11]: is a widely implemented model for managing a company's interactions with customers, clients, and sales prospects. It involves using technology to organize, automate, and synchronize business processes; principally sales activities, but also those for marketing, customer service, and technical support. The overall goals are to find, attract, and win new clients, service and retain those the company already has, entice former clients to return, and reduce the costs of marketing and client service. CRM describes a company-wide business strategy including customer-interface departments as well as other departments. Measuring and valuing customer relationships is critical to implementing this strategy.
- Supply Chain Management (SCM) [41]: is the management of a network of interconnected businesses involved in the provision of product and service packages required by the end customers in a supply chain. SCM spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption. Another definition is provided by the APICS Dictionary when it defines SCM as the "design, planning, execution, control, and monitoring of supply chain activities with the objective of creating net value, building a competitive infrastructure, leveraging worldwide logistics, synchronizing supply with demand and measuring performance globally."
- Knowledge Management System (KMS) [32]: A KMS is a computerized system designed to support the creation, storage, and dissemination of information.

Such a system contains a central repository of information that is well structured and employs a variety of effective and easy to use search tools that users can use to find answers to questions quickly.

- Knowledge Management Portal (KMP) [31]: provides a platform to store information and knowledge into multiple place, exploit it and share it like never before. Helps organization to organically grow their knowledge base. In a typical organization, some 75% of internal knowledge is trapped in one's email and desktop computers. Good information should be everywhere and nowhere near.
- Expert System (ES) [22]: is a computer system that emulates the decision-making ability of a human expert. ESs are designed to solve complex problems by reasoning about knowledge, like an expert, and not by following the procedure of a developer as is the case in conventional programming. The first ESs were created in the 1970s and then proliferated in the 1980s. ESs were among the first truly successful forms of AI software. An ES has a unique structure, different from traditional programs. It is divided into two parts, one fixed, independent of the ES: the inference engine, and one variable: the knowledge base. To run an ES, the engine reasons about the knowledge base like a human. In the 80s a third part appeared: a dialog interface to communicate with users. This ability to conduct a conversation with users was later called "conversational".
- Artificial Neural Network (ANN) [2]: a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data.
- Intelligent Agent (IA)s [28]: is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals (i.e. it is rational). IAs may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an IA, as is a human being, as is a community of human beings working together towards a goal.
- E-commerce DSS [16]: Electronic commerce has the potential to improve efficiency and productivity in business activities. E-commerce today is no longer technological issue, but is also a business issue. DSS is 'an interactive information system that provides information, models and data manipulation tools to help make decisions in semi-structured and unstructured situations. e-Commerce involves a number of forms, varying level of cost and complexity, depending on business need.

### 1.2.1 Enterprise Information System

EISs evolved from Executive Information Systems combined with Web technologies. EIS view information across entire organizations. EIS provides [255]:

- Rapid access to detailed information through drill-down.
- User-friendly interfaces through portals.
- Timely and effective corporate level tracking and control.
- Quick access to detailed information behind the text, numbers, or graphics through drill-down.
- Filter, compress, and track critical data and information.
- Identifies opportunities and threats.
- There are specialized EISs include ERM, ERP, CRM, and SCM.

### 1.2.2 Group Support Systems

Facilitates decisions made by groups within the organization. Traditional meeting can last a long time and money. Collaborative computing systems, groupware, electronic meeting can run over the Web and provide video conferencing, audio conferencing in addition to meeting tools like electronic brainstorming, voting and document sharing [255].

### 1.2.3 Knowledge Management System

Knowledge management is the process of finding, classifying, ensuring the quality of, storing in a repository, maintaining, and using knowledge. Knowledge that is organized and stored in a repository for use by an organization. Knowledge can be used to solve similar or identical problems in the future. Return On Investment (ROIs) as high as a factor of 25 within one to two years [255].

### 1.2.4 Expert Systems

Technologies that apply reasoning methodologies in a specific domain. Attempts to mimic human experts problem solving. Examples include [255]:

- Artificial Intelligence Systems.
- ANNs.
- Genetic Algorithms.
- Fuzzy Logic.
- IAs.

The major benefits of ESs include [255]:

- They provide monetary savings because fewer human experts are needed.
- They provide for improved quality of decisions and/or solutions because they are more consistent, and fewer mistakes are made.
- They are compatible with various decision styles.
- They can be used as a training vehicle.
- They allow an expert to be free from repetitive, time-consuming tasks.
- Scarce expertise is preserved.
- They are able to operate in hazardous environments.

### 1.2.5 Hybrid Support Systems

Integration of different computer system tools to resolve problems. Tools perform different tasks, but support each other. Together, produce more sophisticated answers. Work together to produce smarter answers.

### 1.2.6 Emerging Technologies

- Grid computing.
- Improved GUI.
- Model-driven architectures with code reuse.
- M-based and L-based wireless computing.
- IAs.
- Genetic algorithms.
- Heuristics and new problem-solving techniques.

## 1.3 Enterprise Architecture

EA is the process of translating business vision and strategy into effective enterprise change by creating, communicating and improving the key requirements, principles and models that describe the enterprise's future state and enable its evolution. Practitioners of EA call themselves enterprise architects. An enterprise architect is a person responsible for performing this complex analysis of business structure and processes and is often called upon to draw conclusions from the information collected. By producing this understanding, architects are attempting to address the goals of EA: Effectiveness, Efficiency, Agility, and Durability [266].

EA can also be defined as a discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analyzing the execution of change toward desired business vision and outcomes. EA delivers value by presenting

business and IT leaders with signature-ready recommendations for adjusting policies and projects to achieve target business outcomes that capitalize on relevant business disruptions. EA is used to steer decision making toward the evolution of the future state architecture [139].

Margaret Rouse defines EA as a conceptual blueprint that defines the structure and operation of an organization. The intent of an EA is to determine how an organization can most effectively achieve its current and future objectives [232].

Microsoft's Michael Platt offers a view of EA as containing four points-of-view, called the business perspective, the application perspective, the information perspective, and the technology perspective. The business perspective defines the processes and standards by which the business operates on a day-to-day basis. The application perspective defines the interactions among the processes and standards used by the organization. The information perspective defines and classifies the raw data (such as document files, databases, images, presentations, and spreadsheets) that the organization requires in order to efficiently operate. The technology perspective defines the hardware, operating systems, programming, and networking solutions used by the organization.

Purported advantages of having an EA include improved decision making, improved adaptability to changing demands or market conditions, elimination of inefficient and redundant processes, optimization of the use of organizational assets, and minimization of employee turnover.

Different EA *defacto* standards include Zachman, TOGAF, and ArchiMate [142].

### 1.3.1 TOGAF

TOGAF has been adopted by more than 80% of the worlds leading enterprises as the architecture framework and development method of choice [143]. International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 42010:20071 defines architecture <sup>1</sup> as: the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution." TOGAF embraces and extends this definition. In TOGAF, architecture has two meanings depending upon the context:

1. A formal description of a system, or a detailed plan of the system at a component level to guide its implementation
2. The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time

## TOGAF

TOGAF covers the development of four related types of architecture. These four types of architecture are commonly accepted as subsets of an overall EA, all of which

---

<sup>1</sup>ISO/IEC 42010:2007, Systems and Software Engineering – Recommended Practice for Architectural Description of Software-Intensive Systems, Edition 1 (technically identical to American National Standards Institute / Institute of Electrical and Electronics Engineers (ANSI/IEEE) Std 1471-2000)

TOGAF is designed to support [173]. They are:

1. Business Architecture: The business strategy, governance, organization, and key business processes.
2. Data Architecture: The structure of an organization's logical and physical data assets and data management resources.
3. Application Architecture: A blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization.
4. Technology Architecture: The logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing, and standards.

### 1.3.2 ArchiMate

ArchiMate is an open and independent modeling language for EA, supported by different tool vendors and consulting firms. Users of ArchiMate benefit from sharing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure.

### 1.3.3 Zachman Framework

Zachman Framework will be discussed in details in the next section.

## 1.4 Zachman Framework: Detailed View

The Zachman Framework is an EA framework for EA, which provides a formal and highly structured way of viewing and defining an enterprise. It consists of a two dimensional classification matrix based on the intersection of six communication questions (What, Where, When, Why, Who and How). The Zachman Framework is not a methodology in that it does not imply any specific method or process for collecting, managing, or using the information that it describes. The Framework is named after its creator John Zachman, who first developed the concept in the 1980s at IBM. It has been updated several times since. The Zachman "Framework" is a schema for organizing architectural artifacts (in other words, design documents, specifications, and models) that takes into account both whom the artifact targets (for example, business owner and builder) and what particular issue (for example, data and functionality) is being addressed. Figure 1.1 presented in page 10 shows the current view of the Zachman Framework [267].

	<b>Why</b>	<b>How</b>	<b>What</b>	<b>Who</b>	<b>Where</b>	<b>When</b>
<b>Contextual</b>	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
<b>Conceptual</b>	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
<b>Logical</b>	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
<b>Physical</b>	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
<b>Detailed</b>	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

Figure 1.1: Current View of Zachman Framework [267]

### 1.4.1 Focus of Columns

In summary, each perspective focuses attention on the same fundamental questions, then answers those questions from that viewpoint, creating different descriptive representations (i.e., models), which translate from higher to lower perspectives. The basic model for the focus (or product abstraction) remains constant. The basic model of each column is uniquely defined, yet related across and down the matrix.[26] In addition, the six categories of EA components, and the underlying interrogatives that they answer, form the columns of the Zachman Framework and these are:

1. The data description — What
2. The function description — How
3. The Network description — Where
4. The people description — Who
5. The time description — When
6. The motivation description — Why

In Zachman's opinion, the single factor that makes his framework unique is that each element on either axis of the matrix is explicitly distinguishable from all the other elements on that axis. The representations in each cell of the matrix are not merely successive levels of increasing detail, but actually are different representations — different in context, meaning, motivation, and use. Because each of the elements on either axis is explicitly different from the others, it is possible to define precisely what belongs in each cell.

### 1.4.2 Models of Cells

The kinds of models or architectural descriptive representations are made explicit at the intersections of the rows and columns. An intersection is referred to as a cell. Because a cell is created by the intersection of a perspective and a focus, each is distinctive and unique. Since each cell is distinctive and unique, the contents of the cell are normalized and explicit per the perspectives focus.[26] The cell descriptions in the table itself uses general language for a specific set of targets. Below the focus of each cell in this particular Zachman Framework is explained:

#### **Contextual**

1. (Why) Goal List – primary high level organization goals
2. (How) Process List – list of all known processes
3. (What) Material List – list of all known organizational entities
4. (Who) Organizational Unit and Role List – list of all organization units, sub-units, and identified roles

5. (Where) Geographical Locations List – locations important to organization; can be large and small
6. (When) Event List – list of triggers and cycles important to organization

## Conceptual

1. (Why) Goal Relationship Model – identifies hierarchy of goals that support primary goals
2. (How) Process Model – provides process descriptions, input processes, output processes
3. (What) Entity Relationship Model – identifies and describes the organizational materials and their relationships
4. (Who) Organizational Unit and Role Relationship Model – identifies enterprise roles and units and the relationships between them
5. (Where) Locations Model – identifies enterprise locations and the relationships between them
6. (When) Event Model – identifies and describes events and cycles related by time

## Logical

1. (Why) Rules Diagram – identifies and describes rules that apply constraints to processes and entities without regard to physical or technical implementation
2. (How) Process Diagram – identifies and describes process transitions expressed as verb-noun phrases without regard to physical or technical implementation
3. (What) Data Model Diagram – identifies and describes entities and their relationships without regard to physical or technical implementation
4. (Who) Role Relationship Diagram – identifies and describes roles and their relations to other roles by types of deliverable without regard to physical or technical implementation
5. (Where) Locations Diagram – identifies and describes locations used to access, manipulate, and transfer entities and processes without regard to physical or technical implementation
6. (When) Event Diagram – identifies and describes events related to each other in sequence, cycles occur within and between events, without regard to physical or technical implementation

### Physical

1. (Why) Rules Specification – expressed in a formal language; consists of rule name and structured logic to specify and test rule state
2. (How) Process Function Specification – expressed in a technology specific language, hierarchical process elements are related by process calls
3. (What) Data Entity Specification – expressed in a technology specific format; each entity is defined by name, description, and attributes; shows relationships
4. (Who) Role Specification – expresses roles performing work and workflow components at the work product detailed specification level
5. (Where) Location Specification – expresses the physical infrastructure components and their connections
6. (When) Event Specification – expresses transformations of event states of interest to the enterprise

### Detailed Representation

Eventually the cells with the detailed representation give Rules detail for (Why); Process detail for (How); Data detail for (What); Role detail for (Who); Location detail for (Where); and Event detail for (When). There is a sixth row in the current Zachman framework, but it is not used for EA while the enterprise is described by rows one to six, EA uses only rows one to five, thus only five rows are shown here. Since the product development (i.e., architectural artifact) in each cell or the problem solution embodied by the cell is the answer to a question from a perspective, typically, the models or descriptions are higher-level depictions or the surface answers of the cell. The refined models or designs supporting that answer are the detailed descriptions within the cell. Decomposition (i.e., drill down to greater levels of detail) takes place within each cell. If a cell is not made explicit (defined), it is implicit (undefined). If it is implicit, the risk of making assumptions about these cells exists. If the assumptions are valid, then time and money are saved. If, however, the assumptions are invalid, it is likely to increase costs and exceed the schedule for implementation.

## 1.5 SOA Selected Topics

### 1.5.1 SOA Definition

There is no agreed upon definition for SOA.

- To the chief information officer (CIO), SOA is a journey that promises to reduce the lifetime cost of the application portfolio, maximize return on investment (ROI) in both application and technology resources, and reduce lead times in delivering solutions to the business.

- To the business executive, SOA is a set of services that can be exposed to their customers, partners, and other parts of the organization. Business capabilities, function, and business logic can be combined and recombined to serve the needs of the business now and tomorrow. Applications serve the business because they are composed of services that can be quickly modified or redeployed in new business contexts, allowing the business to quickly respond to changing customer needs, business opportunities, and market conditions.
- To the business analyst, SOA is a way of unlocking value, because business processes are no longer locked in application silos. Applications no longer operate as inhibitors to changing business needs.
- To the chief architect or enterprise architect, SOA is a means to create dynamic, highly configurable and collaborative applications built for change. SOA reduces IT complexity and rigidity. SOA becomes the solution to stop the gradual entropy that makes applications brittle and difficult to change. SOA reduces lead times and costs because reduced complexity makes modifying and testing applications easier when they are structured using services.
- To the IT architect, SOA is the architectural solution for integrating diverse systems by providing an architectural style that promotes loose coupling and reuse. Many IT architects think they have seen this style before with earlier architectural initiatives such as Distributed Computing Environment (DCE), the Distributed Computing Environment, and Common Object Request Broker (CORBA), the Common Object Request Broker Architecture.
- To the developer, SOA is a programming model or paradigm where web services and contracts becomes a dominant design for interoperability. It is a web service when it uses a Webservice Description Language (WSDL) or equivalent specification for describing the service. Web services enable organizations to communicate information, using messages, without intimate knowledge of each others IT systems.

### 1.5.2 SOA is an Architectural Style

SOA is often seen as an architectural style that has been around for years. In SOA scenario, a service consumer invokes or uses a service. The service consumer uses the service description to obtain necessary information about the provider service (e.g., account service) to be consumed. The service description provides the binding information so the consumer can connect to the service, and the description identifies the various operations (e.g., open or close account) available from the provider service. A broker can be used to find the service using a registry that houses information about the service and its location.

### 1.5.3 Fundamental Constructs of SOA

The most basic construct or building block of SOA is a service. Software engineering over the years has evolved from procedural to structured programming to object-

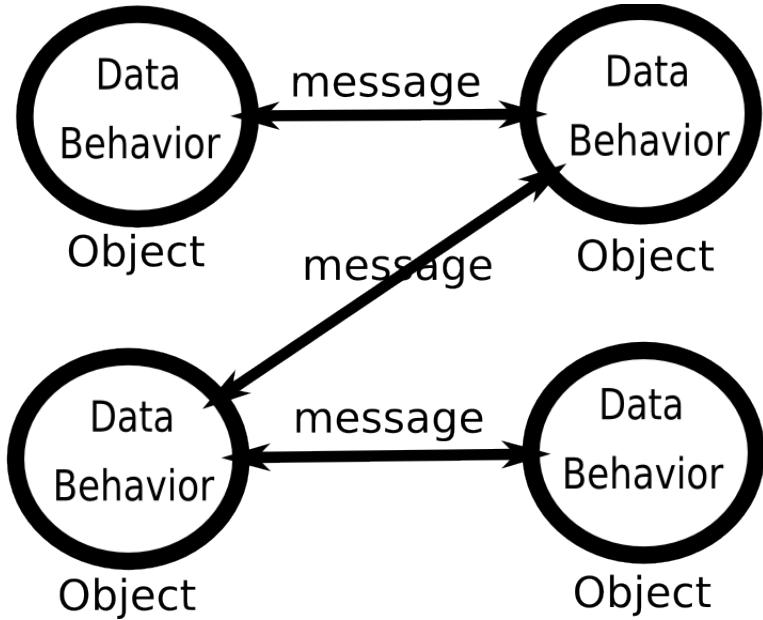


Figure 1.2: Objects encapsulate data and behavior

oriented programming to component-based development and now to service oriented. There are different levels of abstraction from objects to services. Each evolution of abstraction builds on the previous, and SOA embraces the best practices of object and component development. Evolution goes through the following three steps:

- Object is the encapsulation of data and behavior in one data type; class; as depicted in figure 1.2 presented in page 15. Objects are instances of classes that exist for a certain period of time. Objects communicate with each other via exchanged messages. Messages are instructive, not descriptive. Object based development advances software design by providing more support for hiding behavior and data through objects, however, large number of interconnected objects create dependencies that can be difficult to manage. Object based software development is called Object-Oriented Analysis and Design (OOAD) [90].
- Software Components are more sophisticated software modules than objects and require fundamental changes in systems thinking, software processes, and technology utilization. Software component is a unit of composition with contractually specified interfaces and explicit context dependencies [95]. It is a group of objects with has specified interface, working together to provide an application function, such as depicted in figure 1.3 presented in page 16. Component may refer to many different software constructs, from single application logic to an entire functional system. In all cases, a component is a software package with one or more well defined interfaces. Components overlap the properties of object orientation, such as encapsulation and polymorphism, except it reduces the property of inheritance. In component thinking, inheritance is tightly coupled and unsuitable for most forms of packaging and reuse [96].

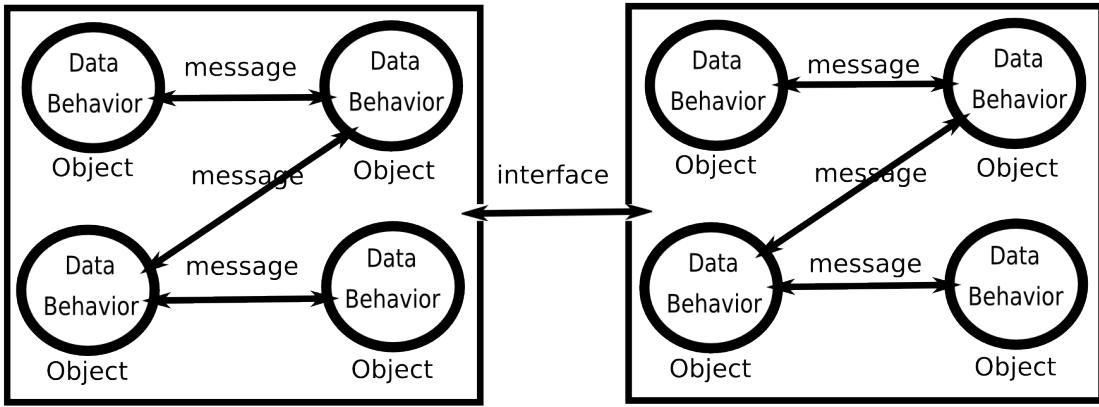


Figure 1.3: Components encapsulate Objects

Instead, components reuse the functionality by invoking other objects and components rather than inheriting from them [97]. Reusable components are good reflection of effective software design. The development of software architecture based on component specifications supports parallel and independent building of the system parts. Many platform vendors have already produced software infrastructures which support component oriented technology.

- **Service and Service Models:** Service is a component capable of performing a task [98]. Services can be categorized based on the nature of logic they encapsulate and the manner in which they are typically utilized within SOA as depicted in [99].

#### 1.5.4 Service Attributes

A service in SOA is the logical, self-contained business function. Services in SOA have the following attributes:

- **Stateless:** SOA services neither remember the last thing they were asked to do nor do they care what the next is. Services are not dependent on the context or state of other services, only on their functionality. Talking on the telephone is stateful, whereas posting a letter is stateless. The World Wide Web provides an excellent example, where each request from a user for a web page or URL results in the requested pages being served, but without the web server remembering the request later. Each request or communication is discrete and unrelated to requests that precede or follow it.
- **Discoverable:** A service must be discoverable by potential consumers of the service. After all, if a service is not known to exist, it is unlikely ever to be used. Services are published or exposed by service providers in the SOA service directory, from which they are discovered and invoked by service consumers.
- **Self-describing:** The SOA service interface describes, exposes, and provides an entry point to the service. The interface contains all the information a

service consumer needs to discover and connect to the service, without ever requiring the consumer to understand (or even see) the technical implementation details.

- **Composable:** SOA services are, by nature, composite. They can be composed from other services and, in turn, can be combined with other services to compose new business solutions.
- **Loose coupling:** Loose coupling allows the concerns of application features to be separated into independent pieces. This separation of concern provides a mechanism for one service to call another without being tightly bound to it. Separation of concerns is achieved by establishing boundaries, where a boundary is any logical or physical separation that delineates a given set of responsibilities. For example, an account service has open account, authorization, and audit features representing delineations of responsibilities and three separations of concerns.
- **Governed by policy:** Services are built by contract. Relationships between services (and between services and service domains) are governed by policies and service-level agreements (SLAs), promoting process consistency and reducing complexity. Independent location, language, and protocol: Services are designed to be location transparent and protocol/platform independent (generally speaking, accessible by any authorized user, on any platform, from any location).
- **Coarse-grained:** Services are typically coarse-grained business functions. Granularity is a statement of functional richness for a service—the more coarse-grained a service is, the richer the function offered by the service. Coarse-grained services reduce complexity for system developers by limiting the steps necessary to fulfill a given business function, and they reduce strain on system resources by limiting the chattiness of the electronic conversation. Applications by nature are coarse- grained because they encompass a large set of functionality; the components that comprise applications would be fine-grained. Similarly, within an application, a service such as “get account information” (which returns name, account number, and address) could be described as coarse-grained, whereas a service to “get account number” could be described as fine- grained.
- **Asynchronous:** Asynchronous communication is not required of an SOA service, but it does increase system scalability through asynchronous behavior and messaging techniques. Unpredictable network latency and high communications costs can slow response times in an SOA environment, due to the distributed nature of services. Asynchronous behavior and messaging allow a service to issue a service request and then continue processing until the service provider returns a response.

### 1.5.5 EAI vs. SOA vs. SOI

Service Oriented Integration (SOI) uses SOA or Web services to integrate applications. It is an evolution from the Enterprise Application Integration (EAI), with the additional innovation of using services and service contracts for integration. This enables the creation of a set of loosely coupled interfaces that can interact with one another to achieve the purposes of integration with greater flexibility. SOI is a subset of SOA (although some see SOI and SOA as different, when in fact SOA is broad and accommodates a wide range of adoption scenarios). This range accommodates different levels of maturity, and the range of adoptions produces different business outcomes. That is, some organizations use only SOA centers for integration and don't seek other strategic benefits of SOA. SOI adoption may or may not use an enterprise service bus (ESB) because SOI is simply the adoption and use of services for integration. The utility of distinguishing SOI and SOA is not particularly useful unless SOI is used to describe specific integration patterns or is defined in a manner that provides some unique utility for software engineering. EAI integrates applications and systems using middleware. EAI solutions are technology based but often not based on standards such as Web services. This is a primary difference between EAI solutions and SOA. Although SOA builds on EAI for integration, it also offers improvements, largely in the use of services. SOA introduces a higher abstraction level than EAI by using services and service contracts. EAI typically uses application programming interfaces (APIs), and SOA uses service interfaces and contracts. EAI has two main integration patterns: hub and spoke and publish-subscribe. In the hub-and-spoke pattern, an application informs the broker of an event (e.g., a DB create, read, update, or delete action performed in the application). The broker takes care of transforming the message, routing the message, and triggering the right action on the client applications. The broker will know, based on internal logic, which client applications to invoke. In the publish-subscribe pattern, the “provider” application publishes events to the middleware, and “client” applications subscribe to these events based on filtering rules. The middleware on a published event will send the transformed message to the subscribing applications. Another difference between EAI and SOA is that in EAI, often, no direct relation exists between requester and provider applications, whereas in SOA there does. In EAI, applications inform the middleware of some event that took place, but they are ignorant about what happens next. It is the responsibility of the middleware to transform and route the message. In an SOA solution, a client application (the service consumer) calls a service, which is mediated through an ESB, on to the service-providing application. Of course, EAI solutions using messaging middleware could be programmed differently to have such awareness.

### 1.5.6 Building Blocks of an SOA Infrastructure

The building blocks of an SOA infrastructure should address the underlying technical infrastructure components needed to support the layers (Consumers; Business Processes; Services; Service Components; Governance; Data Architecture; Quality of Service, Security, Management, Monitoring; and Integration). In identifying the SOA infrastructure components, Figure 1.5.6 presented in page 19 separates the

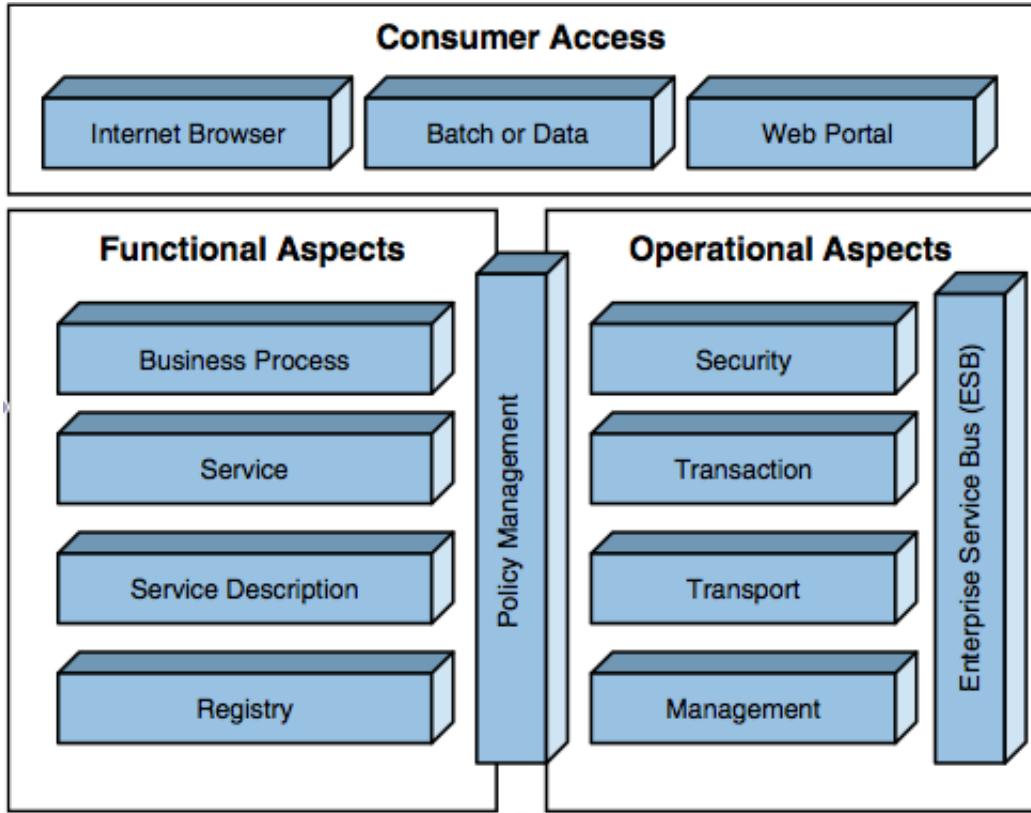


Figure 1.4: Infrastructure SOA Building Blocks [159]

building blocks into three categories (consumer, functional, and operational) to provide an abstraction that can be used for consistency and reuse across platforms within an organization. That is, engineers (enterprise or infrastructure architects) can use this model as a basis for determining the necessary software products or technologies necessary to provision that building block.

### 1.5.7 SOA Role in Embedded or Real-Time Systems

Embedded or real-time systems require a high degree of availability, performance, and response time. Often, embedded systems are responsible for monitoring the health of humans, machines, or services. The role of SOA is akin to its role in providing context-aware services that can be leveraged by embedded or real-time systems. However, the implementation of these systems must address the most stringent performance requirements in latency, availability, and throughout. Therefore, current Web services technologies, for example, might not be suitable for the implementation of such services. There is a class of loosely coupled service implementations that go beyond Web services implementations. They are inspired by SOA service interfaces and loose coupling, and they will be applied to embedded or real-time systems that use mobile robotics and real-time systems that use sensors and actuators within a real-time context.

### 1.5.8 SOA vs. EDA

EDA and SOA are complementary in that SOA allows all the interaction patterns prescribed by EDA and services to be event driven. However, distinct differences exist in the architectural building blocks to support EDA and its interaction patterns. Optimal SOA implementations do not exclusively apply the request-reply interaction pattern, although there may be instances where this is the optimal architectural decision. EDA consists of a set of constructs (runtime artifacts, tools, application programming interfaces) intended to support event-driven behavior. Event-driven behavior can be part of a service design where application logic execution is invoked directly or indirectly because an event has occurred. Perhaps one big difference between EDA and SOA is that the various interaction patterns are designed as services and applications with EDA, whereas with SOA, it's provided externally through the use of ESB, registry patterns, and implementations.

## 1.6 Summary

Today, organizations use computer-based IS to better manage operations in the digital world. These organizations use information systems to provide high-quality goods and services as well as to gain or sustain competitive advantage over rivals. In addition to helping organizations to be competitive, information systems have contributed to tremendous societal changes. An IS can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to supporting decision making, coordination, and control, information systems may also help managers and workers analyze problems, visualize complex subjects, and create new products [45]. IS contain information about significant people, places, and things within the organization or in the environment surrounding it. By information we mean data that have been shaped into a form that is meaningful and useful to human beings. Data, in contrast, are streams of raw facts representing events occurring in organizations or the physical environment before they have been organized and arranged into a form that people can understand and use.

This chapter presented different roles of IS as we move into the digital world and how they have helped fuel decision making. We then highlight what information systems are, how DSS have evolved to become a vital part of modern organizations, what are the main technologies behind DSS, and why this understanding is necessary to become an effective manager in the digital world. Chapter concludes by discussing EA, its components, and different examples of *defacto* standards.

## 1.7 Review Questions

1. Give a definition of each of the following:

- Information System
- Management Support System

- Enterprise Information System
  - Group Support System
  - Knowledge Management System
  - Expert Systems
  - Hybrid Support Systems
  - Enterprise Architecture
2. Enterprise Architecture is a fundamental model of thinking about Organizations' architectures. Given that information, answer the following:
- What is the main difference between Enterprise and Organization?
  - TOGAF, and Zachman are different EA Frameworks. List and describe the different components of each.
  - What is the role of ArchiMate in Enterprise Architecture?
3. What is SOA?
4. What are the Fundamental Constructs of SOA?
5. What is SOA Role in Embedded Systems?

## 1.8 Exercises and Labs

In Labs, you shall be getting familiar with Java programming language, as it is the programming language we will be using to build, deploy, and consume different online Webservices and SOA systems presented in this book.



# Chapter 2

# SW Architecture and Integration

## 2.1 Introduction

It has become accepted that there is a clear need for an ‘architectural view’ of systems [198]. EAs that satisfies nowadays enterprises’ requirements should be presented. Software architecture is one class of EA that should address organizational requirements like systems interoperability, integration, agility, and other requirements [51, 172]. The architectural view of systems (both business and IT systems) is defined in ANSI/IEEE standard 1471-2000 as “the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution”. Enterprises can be thought of as the combination of business needs, and IT capabilities. Agile organizations are those where IT serves business needs, not limiting it [221].

### 2.1.1 Enterprise Architecture Dimensions

Different dimensions of the enterprise need to be defined in order to generate the EA. From enterprise point of view, architectures are classified into four categories [51]: Business Architecture, Information Technology (IT) Architecture, Information Architecture, Application (software) Architecture as depicted in figure 2.1.1 presented in 24. EA tends to define the enterprise from the four dimensions in order to connect between them and present a complete view for the enterprise.

### 2.1.2 Business Architecture

A business or business process architecture defines the business strategy, governance, organization, and key business processes within an enterprise. The fields of Business Process Reengineering (BPR) and BPM focus on the analysis and design of business processes, not necessarily represented in an IT system [51]. Business Architecture defines the business roadmap usually via defining the business processes [29].

### IT Architecture

The IT architecture defines the hardware and software building blocks that make up the overall IS of the organization [51]. IT architecture includes hardware and soft-

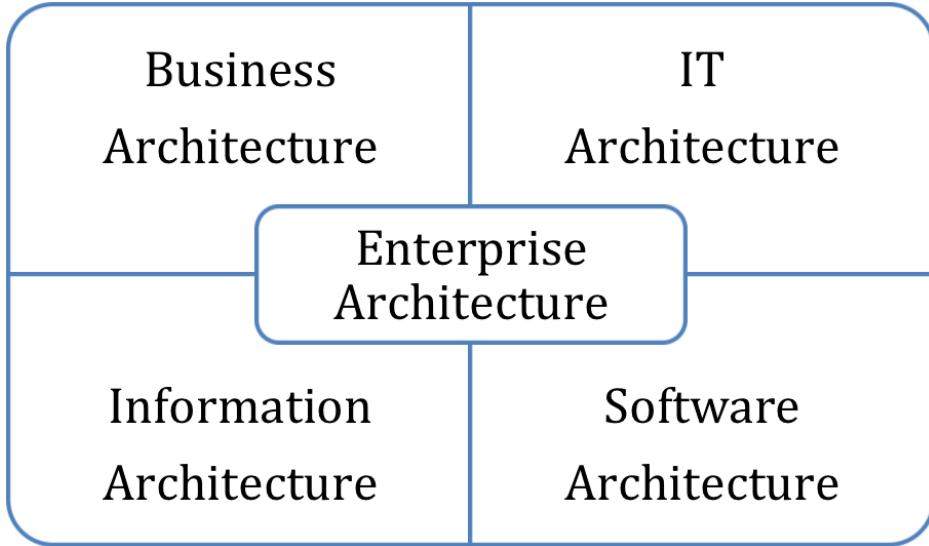


Figure 2.1: EA Dimensions

ware infrastructure including DB and middleware technologies. The IT architecture should enable achievement of the business goals using a software infrastructure that supports the procurement, development, and deployment of core mission-critical business applications. The purpose of the IT architecture is to enable a company to manage its IT investment in a way that meets its business needs by providing a foundation upon which data and application architectures can be built.

### **Information Architecture**

The data architecture of an organization includes logical and physical data assets and data management resources. Information is becoming one of the most important assets a company has in achieving its objectives, and the IT architecture must support it. Information Architecture spans Business and IT Architectures, brings them together, keeps them together, and provides the necessary rich contextual environment to solve the ubiquitous data-quality problem [44].

### **Software Architecture**

Application architecture serves as the blueprint for individual application systems, their interactions, and their relationships to the business processes of the organization. A software application is a computer program or set of programs that uses existing technologies to solve some end-user problem such as the automation of an existing business process. Software architecture can be defined as “the sum of the nontrivial modules, processes, and data of the system, their structure and exact relationships to each other, how they can be and are expected to be extended and modified, and on which technologies they depend, from which one can deduce the exact capabilities and flexibility of the system, and from which one can form a plan for the implementation or modification of the system” [156]. Application Architecture defines the form and function of the applications that will be developed to

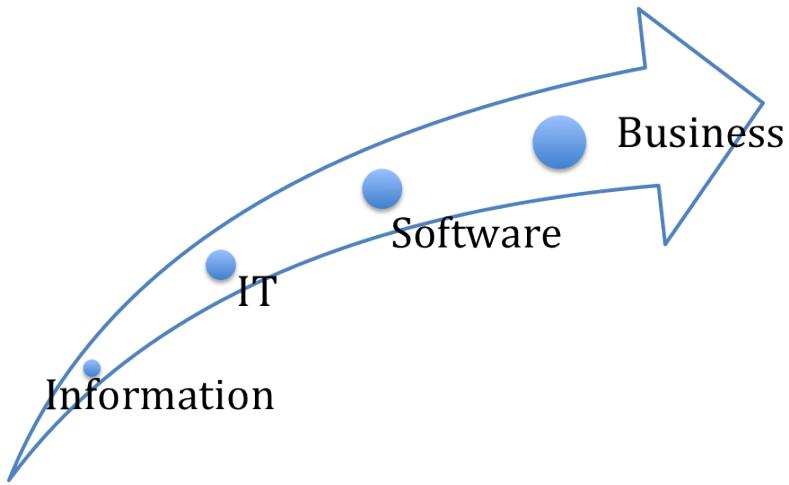


Figure 2.2: Enterprise Architecture Classes Utilization

deliver the required functionality of the system [198]. EA classes utilize each other, and build over each other. Distinctions between classes are blurred because they all serve each other, and serve the enterprise. EA classes' utilization can be thought of as depicted in figure 2.1.2 presented in page 25. Common software architectures are presented in order to address the architecture that might solve most well known software architecture limitations; like lack of scalability, integration, and interoperability with different IS. From this point on, Software Architecture and Architecture will be used interchangeably to refer to Software Architecture.

This chapter goes as follows: Section Two presents the relation between Software Architecture and Systems' Non-Functional Requirements. Section Three discusses the importance of Software Architecture. Section Four lists the most common Software Architecture Patterns while in Section Five authors present "Integration" as the selected criteria for comparing between different patterns. Section Six maps the different integration techniques and different Software Architecture Patterns; highlighting the driving and restraining forces for utilizing each one. Summary is presented in Section Seven with a proposed map between different integration techniques and software architecture patterns.

## 2.2 SW Architecture and System Design

There are aspects that should be designed within any system. Those aspects are: Architecture, Functionality, and Presentation as depicted in figure 2.2 presented in page 26. System design include: Architecturally defining subsystems and their functions, defining interface(s) for each subsystem and determining how those subsystems will interact, and allocating those subsystems to different components, as defined in [114].

Each of System Design's steps includes one or more functionality. Functionality can be mapped to system design steps as follows:

1. Architectural Design: includes identifying and documenting sub-systems mak-

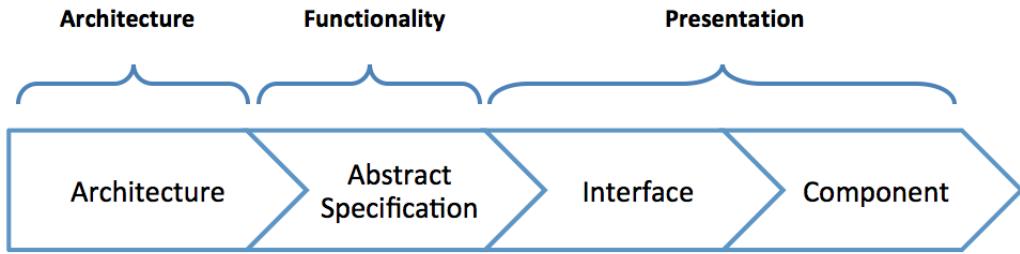


Figure 2.3: Architecture Design is one aspect of System Design

ing the system and their relationships.

2. Abstract Specification: includes producing an abstract specification of each sub-systems services and the constraints under which it must operate.
3. Interface Design: includes defining each subsystems interface with other subsystems is designed and documented.
4. Component Design: includes allocating services to different components and designing interfaces of these components.

Architecture is one aspect of a system to be designed, as well as the presentation of information and the functionality of the system. System requirements are either functional or non-functional [156]. Functional requirements are statements of services that systems should provide. Non-functional requirements present requirements that arose as a result of functional requirements. Architecture design is one step of systems design that shall satisfy non-functional requirements as it satisfies functional requirements [51]. Some of the non-functional requirements that shall be satisfied by architectural design are presented in figure 2.4 presented in page 27.

## 2.3 Importance of SW Architecture

Software (SW) architecture matters because a good one is a key element to success [156]. Though the fact that Systems have architectures but architectures are not systems [51] does not decrease architectures effects on systems. Some of the ways that software architecture influences success are [156]:

- Longevity: It is important to take time defining a good architecture because Architectures live longer than teams created them expects.
- Stability: Architectural stability helps ensure a minimum of fundamental re-work as the system's functionality is extended over multiple release cycles, thus reducing total implementation costs.
- Degree and Nature of Change: Architecture determines the nature of change within the system. Some changes are perceived as applicable; others are perceived as not applicable.

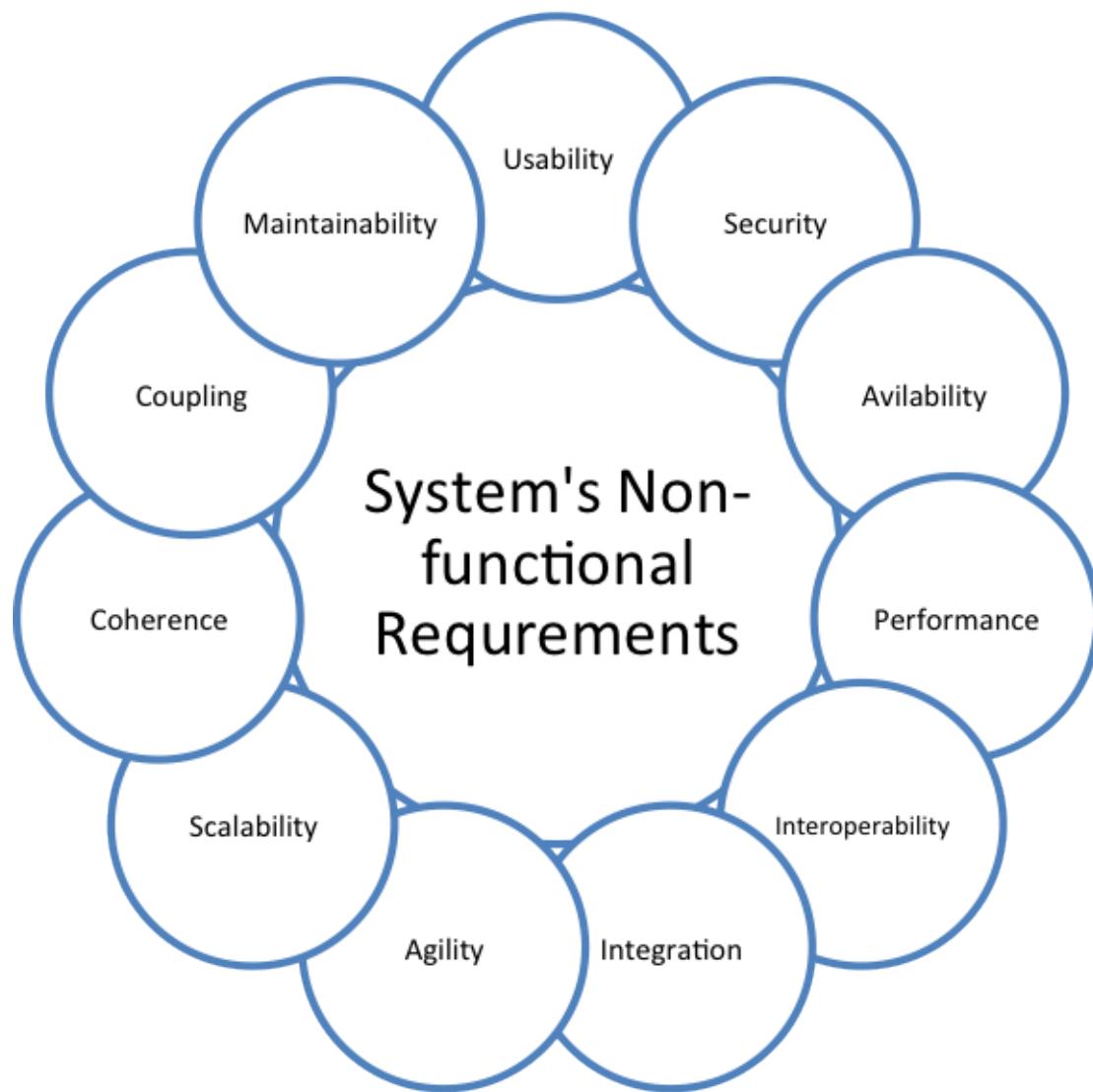


Figure 2.4: Systems non-functional requirements

- Profitability: Profitable architecture is the sustainable architecture within an acceptable cost.
- Social Structure: Architectures helps strengthening and weakening the social structure of the team. Teams can make social benefits from architectures during project time.
- Boundaries Defined: During the architectural design process the team makes numerous decisions about what is "in" or "out of" the system. These boundaries, and the manner in which they are created and managed, are vital to the architecture's ultimate success.

### 2.3.1 Distributed Approach Advantages

Advantages of using a distributed approach to systems development [246]:

1. Resource sharing A distributed system allows the sharing of hardware and software resources—such as disks, printers, files, and compilers—that are associated with computers on a network.
2. Openness Distributed systems are normally open systems, which means that they are designed around standard protocols that allow equipment and software from different vendors to be combined.
3. Concurrency In a distributed system, several processes may operate at the same time on separate computers on the network. These processes may (but need not) communicate with each other during their normal operation.
4. Scalability In principle at least, distributed systems are scalable in that the capabilities of the system can be increased by adding new resources to cope with new demands on the system. In practice, the network linking the individual computers in the system may limit the system scalability.
5. Fault tolerance The availability of several computers and the potential for replicating information means that distributed systems can be tolerant of some hardware and software failures. In most distributed systems, a degraded service can be provided when failures occur; complete loss of service only occurs when there is a network failure.

## 2.4 Common SW Architecture Patterns

As we seek to handle complexity in building software systems, we encounter various architectural styles better suited to solving a certain class of problems. For example, pipes and filters, blackboard architectures, layered architectures, and so on all have been used to solve specific problems in specific domains. However, other architectural styles provide a greater degree of general utility. In other words, the use of that architectural style is not necessarily limited to solving specific problems within

specific domains. Instead, foundation elements of the solutions provided by that architectural style find their way into the foundations of software architecture, rather than being confined to alleviating problems in a specific domain. When developing software systems, service-oriented thinking pushes the boundaries of traditional software architecture and provides new insights into handling complexity, thus deriving commonality and increasing agility. An Architecture Pattern expresses a fundamental structural organization or schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them [1]. Software architecture patterns can be divided into two categories: Data flow and Control flow [156, 64]. Data Flow category focuses on functional modules and data transfers, while Control Flow category describes the way control is passed from one part of the system to the other. Data flow software architecture patterns include:

- Model-View-Controller
- Presentation-Abstraction-Control
- Pipe-And-Filter
- Layered Systems
- Microkernel
- Client-Server
- N-Tier
- Repository
- Blackboard
- Finite State Machine
- Process Control
- Multi Agent System
- SOA
- Master-Slave
- Interpreter
- Message Broker
- Message Bus
- Structural Model
- Peer-to-peer

On the other hand, control flow software architecture patterns include:

- Call And Return a.k.a. Main program And Subroutines
- Implicit Invocation a.k.a. Event Based
- Manager Model
- Emulated Parallel

## 2.5 Non-Functional Requirements

Shortages, limitations, and deficiencies of IS that include lack of agility support and limitations of integration and interoperability are two of the non-functional requirements category. Integration is the main challenge with institutions, and it will be tested against the mentioned software architectures as an indicator for the architecture satisfaction of non-functional requirements.

### 2.5.1 Design Issues of Distributed Systems

Some of the most important design issues that have to be considered in distributed systems engineering are [246]:

1. **Transparency** To what extent should the distributed system appear to the user as a single system? When it is useful for users to understand that the system is distributed?
2. **Openness** Should a system be designed using standard protocols that support interoperability or should more specialized protocols be used that restrict the freedom of the designer?
3. **Scalability** How can the system be constructed so that it is scaleable? That is, how can the overall system be designed so that its capacity can be increased in response to increasing demands made on the system?
4. **Security** How can usable security policies be defined and implemented that apply across a set of independently managed systems?
5. **Quality of service** How should the quality of service that is delivered to system users be specified and how should the system be implemented to deliver an acceptable quality of service to all users?
6. **Failure management** How can system failures be detected, contained (so that they have minimal effects on other components in the system), and repaired?

### 2.5.2 Integration

Application integration is a strategic approach to binding many IS together [192]. The need to integrate applications has been a requirement since business process automation was presented [121]. Application integration can be one of the forms:

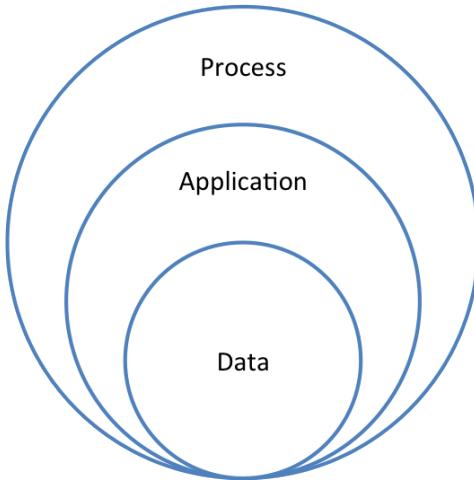


Figure 2.5: Integration Levels

Internal Application Integration, External Application Integration, or EAI [231]. Internal application integration techniques integrate organizational applications with each other, while external application integration techniques consider integrating organizational applications with applications outside organizational boundaries. EAI platforms centrally integrate heterogeneous system landscapes on process, method and data level [65]. Internal and External Application Integration present traditional integration levels, where EAI presents recent integration levels. Figure 2.5 presented in page 31 depicts different integration levels as presented in [121].

Data level integration allows data to be shared among applications without sharing its application logic. Application level integration integrates applications by getting both connected applications logic aware and enablers of exchanging data. Process level integration presents a new level of integrating applications where integration here facilitates or presents a new business processes.

## 2.6 SW Architectures and Integration Techniques

Integration is one of the addressed problems that requires picking suitable software architecture as a solution. The agile organization is more extensively integrated than previous organizational forms [15], and we need to have agile institutions. Considering how architecture works, advantages and shortages of the architecture, there are architectures that can overcome integration obstacles, and others cannot. There are many integration options that include File Transfer, Shared DB, and Messaging [157]. Considering integration techniques defined in [61], there are three common system integration techniques:

1. Enterprise Wide Standards
  - Standard Data Element Definition
  - Standard Enterprise Wide software

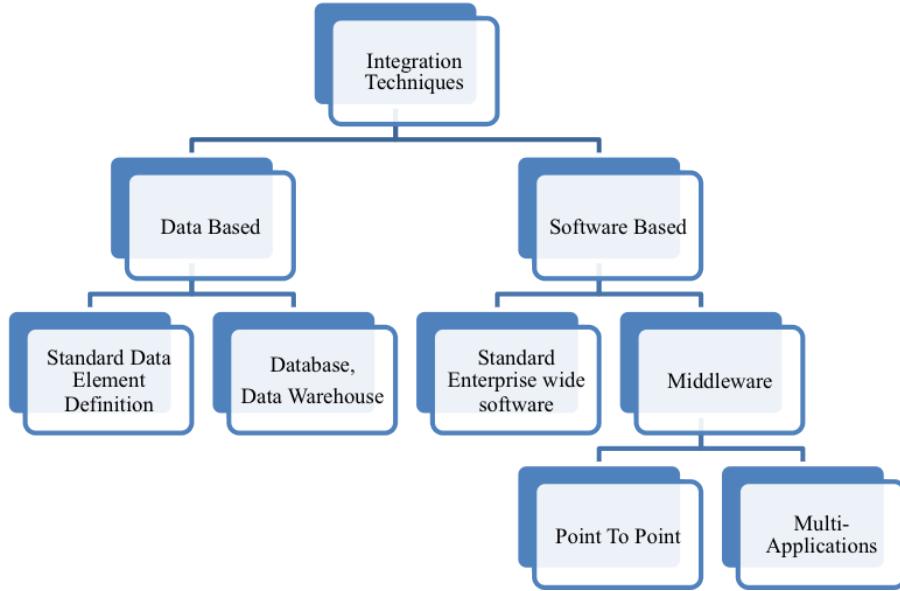


Figure 2.6: Integration Techniques

## 2. Middleware

- TP, Remote Procedure Call (RPC), MOM, CORBA, DCOM, . . .
- Web services

## 3. Additional Components

- DW
- Application Routers

Figure [44] presented in page 32 depicts a reorder of those integration techniques to be categorized in either one of two categories: Data or Software. Each category satisfies an integration level. Data oriented integration techniques satisfy Data level integration, and Software integration techniques satisfy Application level integration. Each integration technique has driving and restraining forces. Driving forces are forces that push software architects towards using the mentioned technique and supporting software architectures. Restraining forces are forces that push software architects not to use the mentioned technique and supporting architectures.

### 2.6.1 Data Based Integration Techniques

Those are techniques that relies on databases to present integration either by presenting unified standard data element definition, so all applications within the organization can share same data, or by presenting databases, and data warehouses as the organizational repository, and all running applications share same data repository. By using either technique method, applications are integrated; unfortunately; only on DB level. Several restraining forces against this technique are presented in the coming two sections.

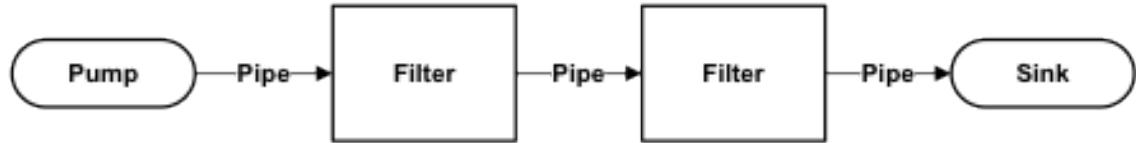


Figure 2.7: Pipe-And-Filter Architecture

### Adopting Standard Data Element Definition

Custom software uses the same data element definitions as all other software within the system. Thus, uniqueness in syntax and semantics are presented. One of the Software Architectures that fits this technique is: Pipe-And-Filter software architecture.

- **Pipe and Filter Architecture:** A very simple, yet powerful architecture, that is also very robust. It consists of any number of components (filters) that transform or filter data, before passing it on via connectors (pipes) to other components. The filters are all working at the same time. The architecture is often used as a simple sequence, but it may also be used for very complex structures. Figure 2.7 presented in page 33 depicts mechanism of action of this software architecture [114, 64].

Components of this architecture are:

- Pump: or producer is the data source. It can be a static text file, or a keyboard input device, continuously creating new data, or data of another application.
- Pipe: is the connector that passes data from one filter to the next. It is a directional stream of data that is usually implemented by a data buffer to store all data, until the next filter has time to process it.
- Filter: transforms or filters the data it receives via the pipes with which it is connected. A filter can have any number of input pipes and any number of output pipes.
- Sink: or consumer is the data target. It can be another file, a DB, computer screen, or another application.

Applications share same standard data element definition, so, they can pipe data easily among them. By doing so, applications are integrated on data basis. Table 2.6.1 presented in page 34 shows Driving and Restraining forces of Adopting Standard Data Element Definition as an integration technique, which unfortunately makes adopting databases and data warehouses not the perfect solution for process level integration. Figure 2.8 presented in page 34 presents another view of the driving and restraining forces.

Table 2.1: Adopting Standard Data Element Definition Pitfalls

Driving Forces	Restraining Forces
Easier exchange o f data	Costs to develop standards definitions
Reduced development time	Costs to change existing systems
Reduced maintenance costs	Existing data definitions are different
	Some definitions need to be different
	Products use different data definitions
	Lack of industry standard definitions
	Mergers and acquisitions

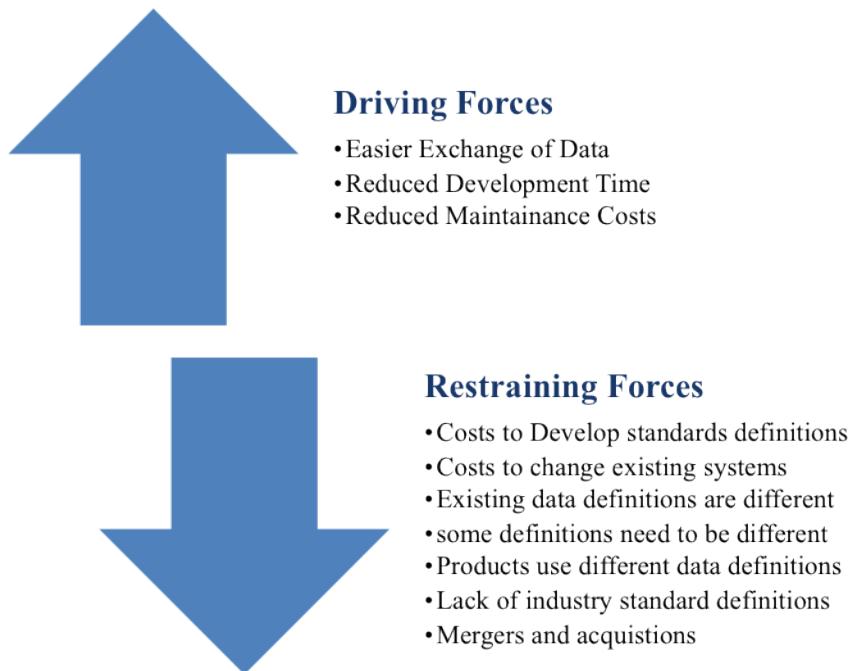


Figure 2.8: Adopting Standard Data Element Definition Pitfalls

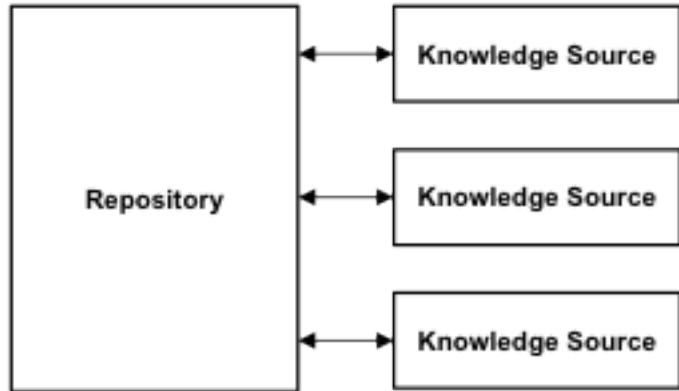


Figure 2.9: Repository Architecture

### Adopting DB and DW

DB is a collection of persistent data <sup>1</sup> that is used by the application systems of some given enterprise <sup>2</sup>. Databases have too many advantages and they play an important role in today organizations. Benefits of data approach include sharing of data, and reducing redundancy [96]. Databases allow data to be shared, that means existing and new applications can share the data in the DB. Databases reduce data redundancy by eliminating, or at least controlling redundancy. DW is the subject-oriented, integrated, nonvolatile <sup>3</sup>, time variant data store in support of management's decisions [165, 167]. DW can be thought of as a collection of data from more than one data source (data sources include databases) for providing a single, quiet large repository for the entire organization. Databases and data warehouses have been widely used to integrate applications within organizations. Typical DW can be found in [93]. One of the Software Architectures that fits this technique is: Repository software architecture.

**Repository Software Architecture** The repository contains a single data structure, the Repository, and a number of modules called Knowledge Sources, that modify this data structure. These are the only characteristics of repository architecture [114, 64]. Figure 2.9 presented in page 35 depicts how repository architecture behaves.

Table 2.6.1 presented in page 36 depicts the driving and restraining forces of adopting databases and data warehouses as an integration solution, which unfortunately makes adopting databases and data warehouses not the perfect solution for process level integration. Unfortunately, there are restraining forces that make adopting databases and data warehouses not the perfect solution for internal and external organizational applications integration. Figure 2.10 presented in page 37

---

<sup>1</sup>Persistent Data are data that once accepted by Database Management System (DBMS) for entry, can subsequently be removed only by some explicit request to DBMS.

<sup>2</sup>Enterprise is simply a convenient generic term for any reasonably self-contained commercial, scientific, technical or other organization.

<sup>3</sup>Nonvolatile means that, once inserted, data cannot be changed, though it might be deleted.

Table 2.2: Adopting Databases and Data warehouses Pitfalls

Driving Forces	Restraining Forces
Easier access to enterprise wide data	Costs of development
Reduced development time	Different semantics in data sources
Reduced maintenance costs	Semantic translation
Minimal effect on operational system	Lack of industry standard definitions
Usage of BI software	Deciding what data to warehouse
	Delays in getting data to warehouse
	Redundancy of data
	Data quality issues
	Brittleness of fixed record exchanges
	Performance tuning

depicts another view of adopting databases and data warehouses pitfalls.

### 2.6.2 SW Based Integration Techniques

Another integration technique presented is the one based on using software as the orientation of integration solution. Either using standard enterprise wide software or presenting middleware is a software oriented solutions.

#### Standard Enterprise Wide Software

This means that whole organization uses the same software; this means that the whole organization is using the same data definitions, semantics, and formats for exchanging data. ERP presents this technique as a solution to whole enterprise integration solution. ERP integrates SCM System, Enterprise Management System, CRM, e-commerce [89, 151, 226]. ERP providers list include Oracle [37], and SAP [39]. To provide a unified solution to the enterprise, layered architecture is required to enable integration of applications in different aspects. Software Architectures that fit this technique include: Layered Systems, Client-Server, and N-tier architecture.

- **Layered Systems:** Layered Systems use layers to separate different units of functionality. Each layer only communicates with the layer above and the layer below. Each layer uses the layer below to perform its function. Communication happens through predefined, fixed interfaces. A Layer is a design construct. It is implemented by any number of classes or modules that behave like they are all in the same layer. That means that they only communicate with classes in layers immediately above or below their layer and with themselves. Figure 2.11 presented in page 38 depicts layered systems architecture [64]. Each layer offers its own kind of functionality. A higher layer uses its lower layer to perform its function. It requires its lower layer. It is possible to define multiple layers at the same level. The user calls a function on an object in the upper layer. This object calls functions in the layer below. These functions in turn approach the layer below and the layer above. The primary disadvantage of layered systems

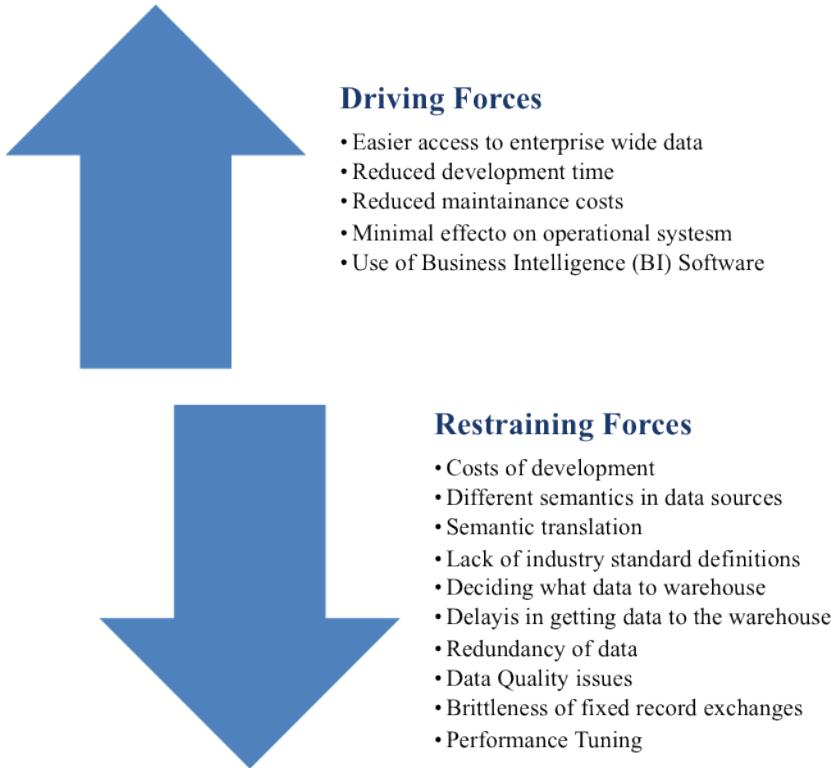


Figure 2.10: Adopting Databases and Data warehouses Pitfalls

is that they add overhead and latency to the processing of data, reducing user-perceived performance [91]. Another software architecture that ERPs heavily rely on integrating enterprise applications is the Client-Server and N-Tier architecture. A detailed SAP architecture based on N-Tier architecture is presented in [54]. There are five architectural patterns for layered systems:

1. Master-slave architecture, which is used in real-time systems in which guaranteed interaction response times are required.
2. Two-tier client-server architecture, which is used for simple client-server systems, and in situations where it is important to centralize the system for security reasons. In such cases, communication between the client and server is normally encrypted.
3. Multitier client-server architecture, which is used when there is a high volume of transactions to be processed by the server.
4. Distributed component architecture, which is used when resources from different systems and databases need to be combined, or as an implementation model for multi-tier client-server systems.
5. Peer-to-peer architecture, which is used when clients exchange locally stored information and the role of the server is to introduce clients to each other. It may also be used when a large number of independent computations may have to be made.

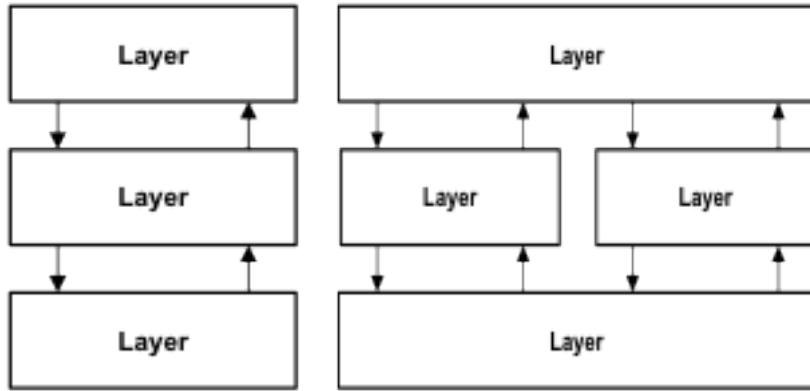


Figure 2.11: Layered Architecture Different Implementations

- **Client – Server Architecture:** Distributed systems that are accessed over the Internet are normally organized as client–server systems. In a client–server system, the user interacts with a program running on their local computer (e.g., a web browser or phone-based application). This interacts with another program running on a remote computer (e.g., a web server). The remote computer provides services, such as access to web pages, which are available to external clients. This client–server model, is a very general architectural model of an application. It is not restricted to applications distributed across several machines. You can also use it as a logical interaction model where the client and the server run on the same computer. In a client–server architecture, an application is modeled as a set of services that are provided by servers. Clients may access these services and present results to end users. Clients need to be aware of the servers that are available but do not know of the existence of other clients. Clients and servers are separate processes. The client-server style is the most frequently encountered of the architectural styles for network-based applications [7]. Client - Server components are: Client triggers process, Server reactive process [56]. Clients make requests that trigger reactions from servers, as presented in figure 2.12 presented in page 39. Variety of Client – Server systems are surveyed in [241, 261]. Client – Server architecture mainly consists of two layers: Single Server and many clients. Figure 2.13 presented in page 40 depicts Client – Server Architecture different implementations. A Client-Server system is one in which the server performs some kind of service that is used by many clients [64]. The clients take the lead in the communication. The basic form of client-server does not constrain how application state is partitioned between client and server components, it is often referred to by the mechanisms used for the connector implementation, such as remote procedure call or message-oriented middleware [7].
- **N-Tier Software Architecture:** N-Tier architecture is a Client-Server architecture combined with the Layered architecture where N equals three or higher [64]. Three-Tier architecture is an example of N-Tier architecture. Figure 2.14 presented in page 41 depicts Three-Tier Architecture. Three-Tier

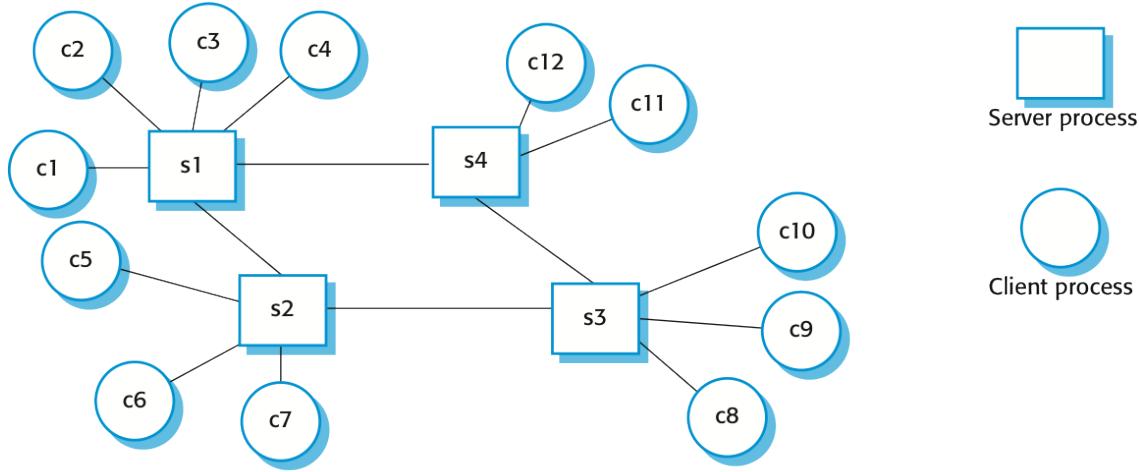


Figure 2.12: Layered Architectural Model of Client-Server [246]

architecture consists of three layers: Presentation, Application, and DB.

- **Presentation Layer:** deals with user interactions. Thin clients interface do not contain business logic, just code required to process user input, send requests to the server, and show the results of these requests.
- **Application Layer:** processes client requests. It is the actual web application that performs all functionality specific to the web application. However, it does not store the persistent data itself. Whenever it needs data of any importance, it contacts the DB server.
- **DB Layer:** contains DB and DBMS (DBMS).

Other applications can be divided into four layers [246]. Figure 2.15 presented in page 41 shows that four Layers applications can be:

1. A presentation layer that is concerned with presenting information to the user and managing all user interaction;
2. A data management layer that manages the data that is passed to and from the client. This layer may implement checks on the data, generate web pages, etc.;
3. An application processing layer that is concerned with implementing the logic of the application and so providing the required functionality to end users;
4. A DB layer that stores the data and provides transaction management services, etc.

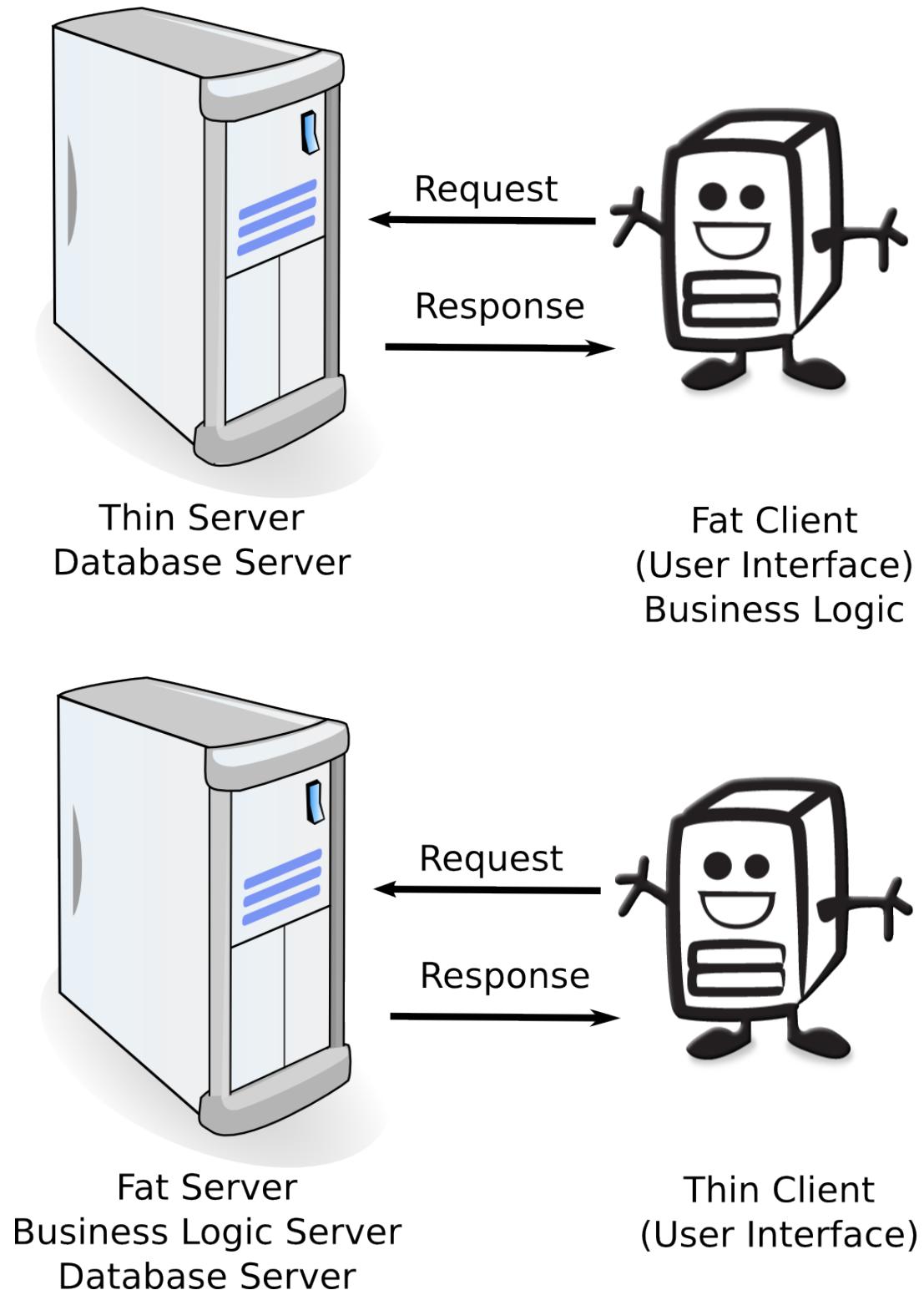


Figure 2.13: Client - Server Architecture Different Implementations

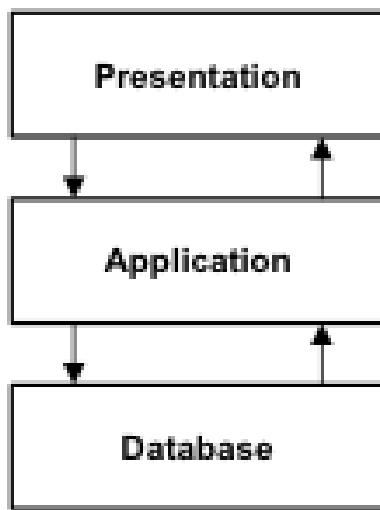


Figure 2.14: Three Tier Architecture

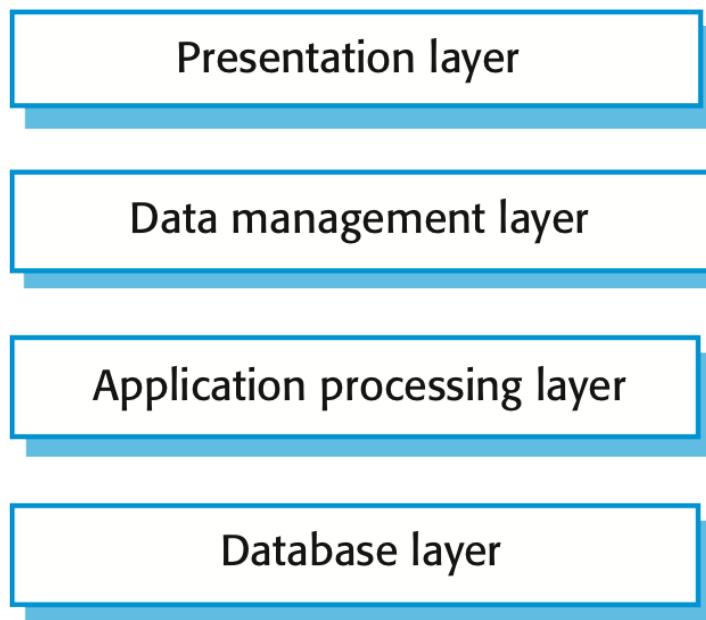


Figure 2.15: Layered architectural model for client–server application [246]

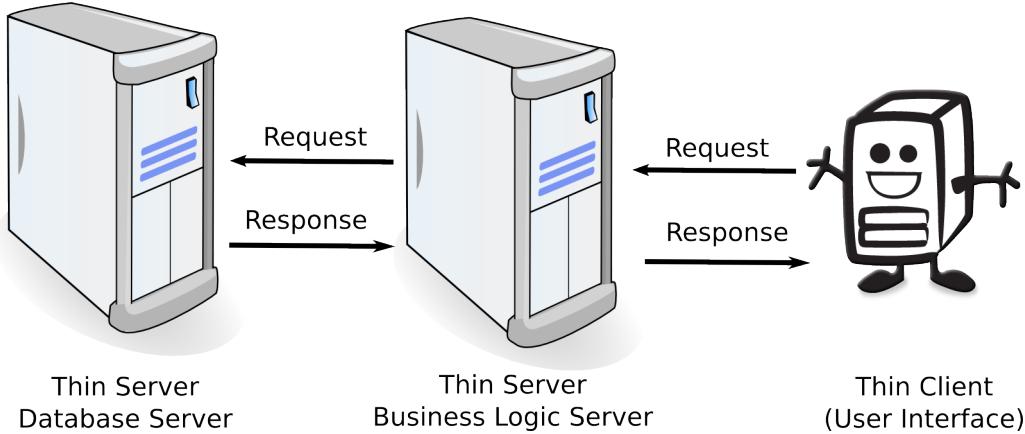


Figure 2.16: Three Tier Architecture Implementation

In the second and third tier there can be multiple instances, because of scalability, load-balancing and redundancy [10]. N-tier architecture (with N more than 3) is really 3 tier architectures in which the middle or bottom tier is split up into new tiers. Figures 2.16, 2.17, 2.18, and 2.19 present different implementations of N-Tier architecture. Figure 2.16 presented in page 42 shows the simplest implementation of Three-Tier architecture, where the most left layer that includes Smart Phone, Cell Phone, and PC presents required logic to display information and validate user inputs. Business logic ‘presented in applications’ required to provide web applications to users are present on the Business Logic server. Required data to support those applications are present in the DB server.

Figure 2.17 presented in page 43 illustrates the repetition of data layer by presenting more than one DB server. DB servers can be connected together or separated. Business logic server can connect to both DB servers, or connect to one DB server incase DB servers are connected together. Servers’ connections is an architectural decision that software architect shall consider while designing software architecture.

Figure 2.18 presented in page 44 presents another N-Tier implementation which replicates business logic servers. Scalability can force such a solution. Business logic servers can be connected to each other or not based on application requirements.

Figure 2.19 presented in page 45 presents the third N-Tier implementation where business logic and DB servers are replicated. Large scaled institutions require this type of architectures to face increasing requirements. N-Tier Architecture serves integration by forcing enterprises to implement standard software at each tier. Even if different software is implemented, by connecting each tier to the other, enterprises overcome integration problems. N-Tier has been widely accepted in today’s organizations as the standard architecture. Unfortunately, N-Tier architecture does not define what tier should include, how tiers should connect to each other, thus, architectural decisions still needs to be taken by software architect. Table 2.6.2 shows driving and restraining forces on adopting wide standard software as an integration solution [104, 33, 34]. Unfortunately, this integration technique is not well suited for process level integration. Figure 2.20 presented in page 45 presents another view of enterprise software pitfalls.

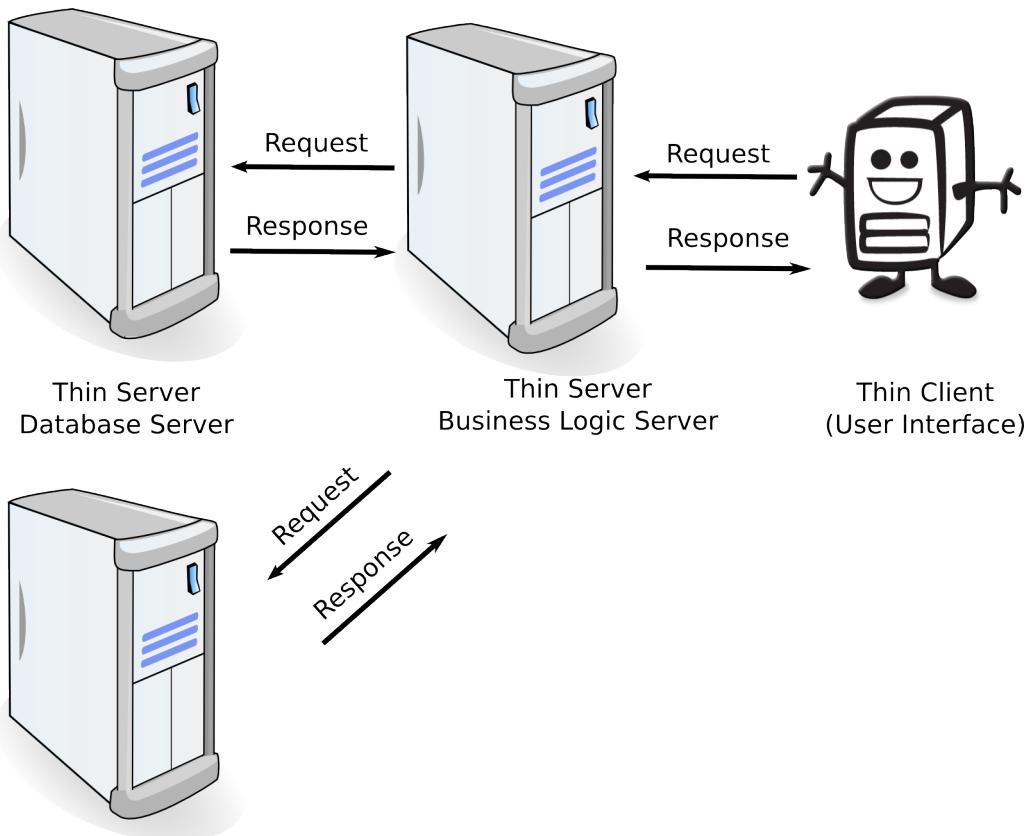


Figure 2.17: Three Tier Architecture First Implementation

Table 2.3: Adopting Standard Enterprise Software Pitfalls

Driving Forces	Restraining Forces
Easier access to enterprise wide data	Mergers and acquisitions
Reduced development time	Departments have different needs
Reduced maintenance costs	Dependence on different software
	Conversion to new software

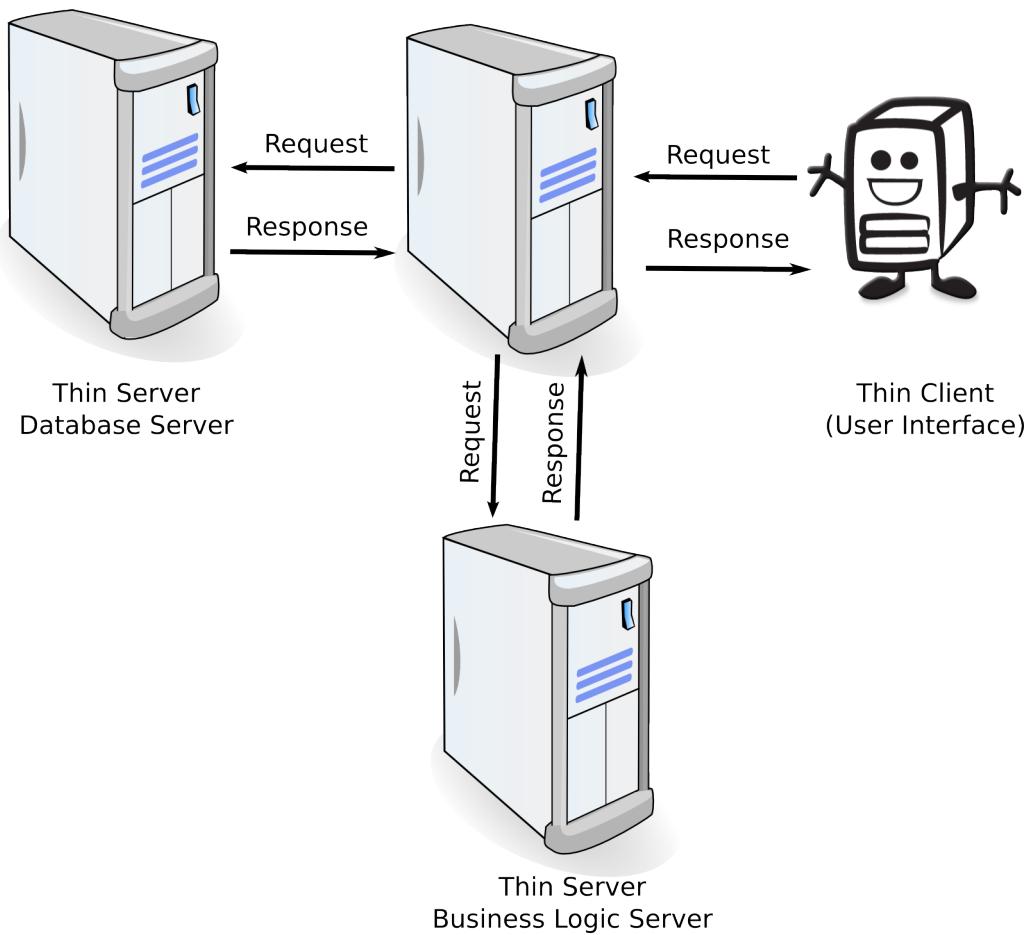


Figure 2.18: Three Tier Architecture Second Implementation

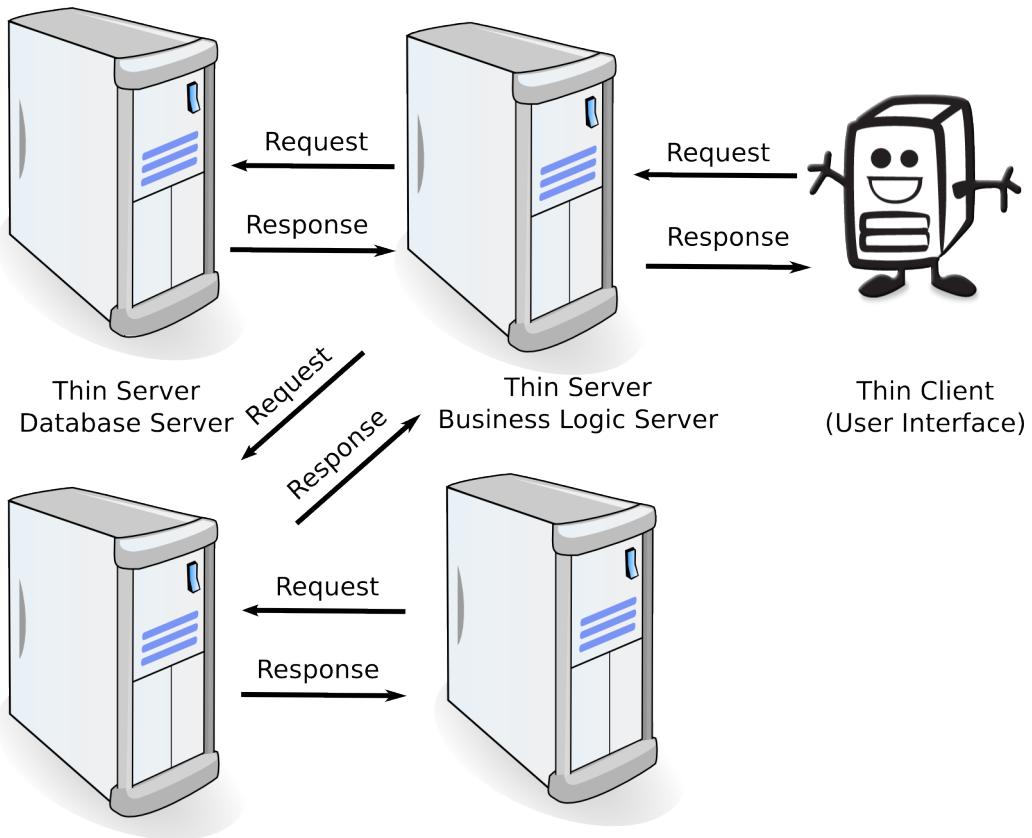


Figure 2.19: Three Tier Architecture Third Implementation

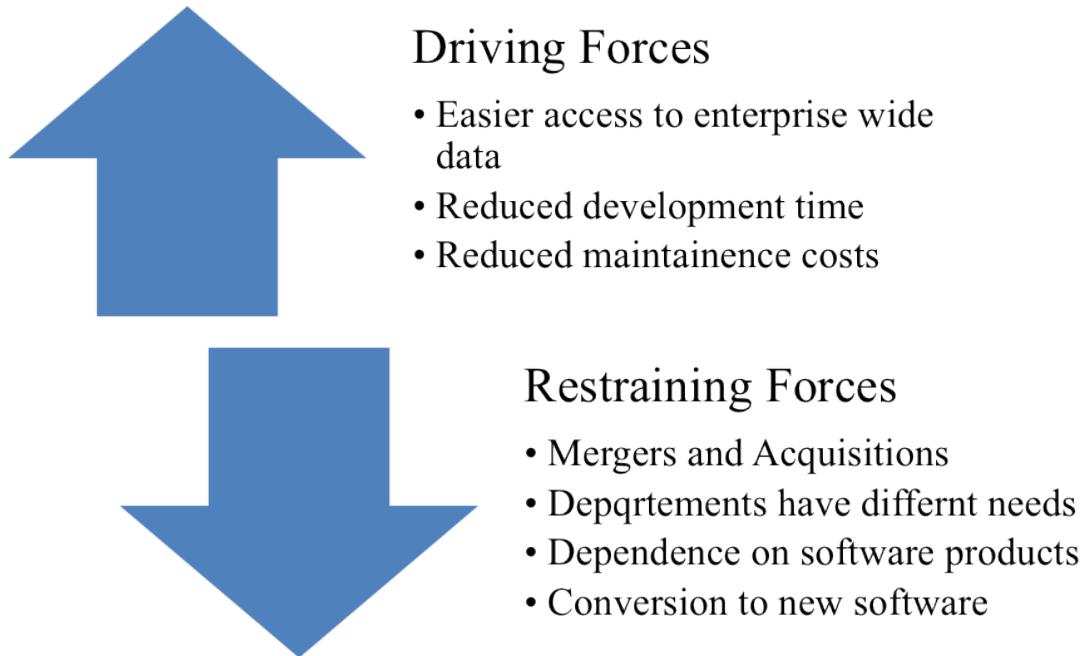


Figure 2.20: Adopting Standard Enterprise Software Pitfalls

## Middleware Integration Technique

Middleware integration facilitates communication between applications in order to exchange data and business logic. This communication is either: Point-To-Point, or Multi-Applications.

- **Point To Point Middleware:** Point To Point refers to direct connection and communication between two applications. This direct communication can be presented by two methods: Adapters, and Remote Procedure Calls (RPCs).
  - Application Adapters: In this case, a special purpose adapter is presented to integrate two applications and provide communication and exchange of data among them. Examples of adapters are presented in [51]. This adapter is capable of transforming data syntax and semantics, and managing communication among connected application. Special adapters are presented for each application.
  - RPC: an RPC allows execution of program logic on a remote system by calling local routine. This calling is done via special message between client and server <sup>4</sup>.
- **Multi-Applications Middleware:** The components in a distributed system may be implemented in different programming languages and may execute on completely different types of processor. Models of data, information representation, and protocols for communication may all be different. A distributed system therefore requires software that can manage these diverse parts, and ensure that they can communicate and exchange data. The term ‘middleware’ is used to refer to this software—it sits in the middle between the distributed components of the system. Middleware is normally implemented as a set of libraries, which are installed on each distributed computer, plus a run-time system to manage communications. Middleware is general-purpose software that is usually bought off the shelf rather than written specially by application developers. Examples of middleware include software for managing communications with databases, transaction managers, data converters, and communication controllers. Multi-Applications middleware relies heavily on messages in order to hide complexity of communication between systems. It is not acceptable to provide adapters for the whole system; instead, a middleware is required to integrate applications within the system [61]. Exchanging messages enabled integration on software level. Table 2.6.2 presented in page 47 depicts driving and restraining forces of adopting message routing integration technique. Software architectures supported different implementations for integrating applications via exchanging messages. Those architectures include: Message Broker, Message Bus, and Broker. Figure 2.21 presented in page 47 depicts another view of adopting message routing pitfalls.
  - Message Bus: Message bus architecture connects all applications through the logical component known as Message Bus [28]. Message bus is respon-

---

<sup>4</sup>In this case, machine that initiated RPC is considered Client, while the machine that holds code to be invoked and executed by reception of RPC is Server.

Table 2.4: Adopting Message Routing Pitfalls

Driving Forces	Restraining Forces
Consistent enterprise wide data	Costs of development
Reduced development time	Different semantics in data sources
Reduced maintenance costs	Semantic translation
Minimal effect on systems	Lack of industry standard definitions
	Deciding what data to route
	Delays getting data updates distributed
	Data quality issues
	Brittleness of fixed record exchange

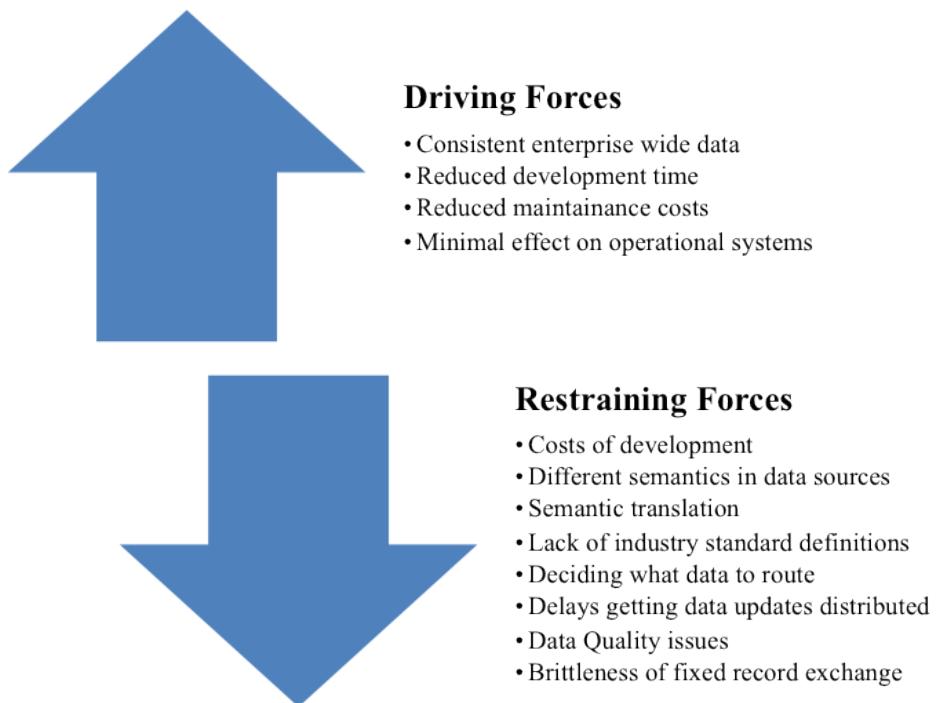


Figure 2.21: Adopting Message Routing Pitfalls

sible for delivering messages to destined applications. Sending application has no responsibility of direct connection with destined application. Sending application needs to prepare the message in the message bus agreed format, and delivers it to the message bus. Figure 2.22 presented in page 49 depicts how message bus architecture works. There are three options for subscribing the application to the Message Bus [38]:

- \* List-Based Publish/Subscribe: There are managed lists for subscribers. All subscribers receive certain messages when they are sent to the bus.
- \* Broadcast-Based Publish/Subscribe: Messages are sent to all connected applications.
- \* Content-Based Publish/Subscribe: it enables Message Bus to investigate message contents to determine recipient of the message.

Limitations include that message bus can expose only one interface. In order for an application to be part of the architecture, it should implement the message bus interface, use the message bus messages type, and registers itself to the message bus. Performance issues are also a potential problem; poor application partitioning can create excessively high volumes of messages, and some use cases can be impacted through high network latency [155]. One of middleware integration techniques implementation that utilizes message bus is CORBA. CORBA: is the acronym for Common Object Request Broker Architecture, OMG's open, vendor-independent architecture and infrastructure that computer applications use to work together over networks [9]. CORBA applications are composed of objects. CORBA creates a distributed objects infrastructure which makes activating and accessing remote objects transparent [71]. Legacy application is wrapped in code with CORBA interfaces and opened up to clients on the network. Systems wrapped with CORBA can be integrated under CORBA architecture. Table 2.6.2 presented in page 49 depicts driving and restraining forces of adopting CORBA as an integration technique implementation. Figure 2.21 presented in page Figure 2.23 presented in page 49 depicts another view of adopting message CORBA pitfalls.

- Message Broker: The message broker can expose different interfaces to the collaborating applications, and it can translate messages between these interfaces [33]. Message broker does not enforce a common interface on the applications, thus handles shortages of Message Bus. Figure 2.24 presented in page 50 depicts Message Broker architecture. Subscribing options are the same as mentioned in Message Bus architecture. Commercially, Message Broker is referred to as Message Oriented Middleware (MOM). One commercial implementation of MOM is Microsoft BizTalk Server [3]. Software Architectures that support Message Broker includes: Broker 'SOA'.
- SOA: Servers can register and deregister with the broker. If a server fails, it will be automatically (after a timeout) unregistered by the bro-

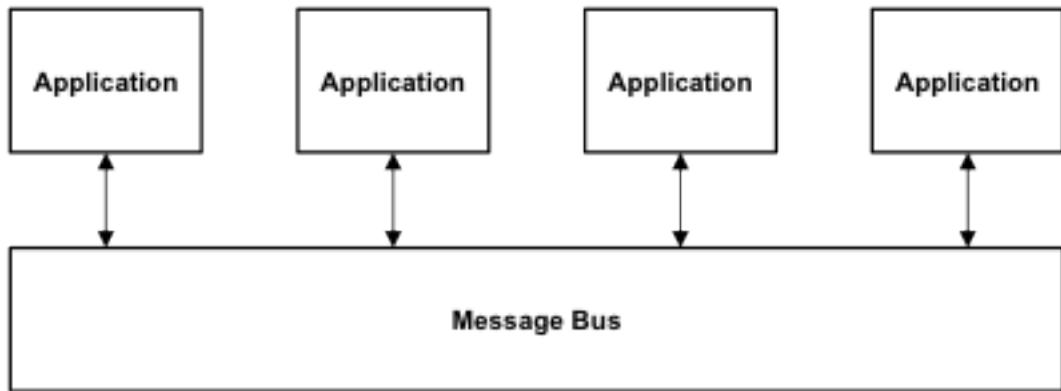


Figure 2.22: Message Bus Architecture

Table 2.5: Adopting Message CORBA Pitfalls

Driving Forces	Restraining Forces
Interoperable networked applications	Different semantics in data sources
Reduced development time	Semantic translations
Reduced maintenance costs	Lack of industry standard definitions
	Lack of CORBA/DCOM product support
	Perceived CORBA/DCOM complexity
	Effect on operational systems of data requests
	Merges and acquisitions

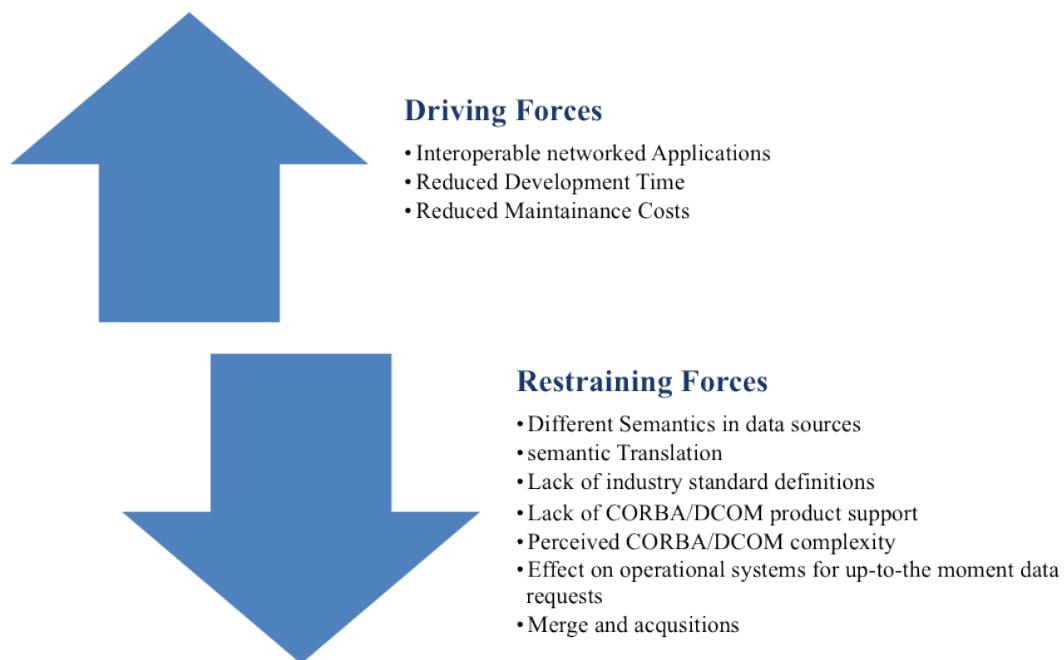


Figure 2.23: Adopting Message CORBA Pitfalls

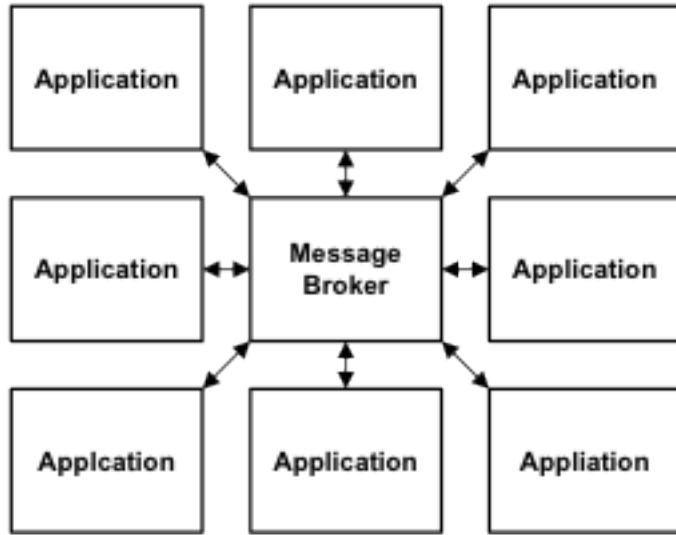


Figure 2.24: Message Broker Architecture

ker. Broker handles user requests by providing user capability to find different service providers, pick the most suitable one, and utilizes its service [64]. The client requests a specific service. It formats its request in a specific format and sends it to its broker. The broker then selects the most suitable server to process the request. When the link between the client and the server is set up, they may start communicating directly, freeing the broker. Figure 2.25 presented in page 51 depicts broker architecture mechanism of action. CORBA was attempted to satisfy Broker architecture requirements, unfortunately CORBA itself has shortages that inhibited it from being the standard integration solution. SOA is the most suitable architecture to integrate organizations on process level. SOA integration is the strongest integration provided to organizations [65]. With this importance for SOA, CORBA limitations were addressed among software vendors and industry standards creators. Web services were presented as the industry solution to overcome CORBA limitations. Table 2.6.2 presented in page 51 depicts driving and restraining forces of adopting Web services as an integration solution. Figure 2.26 presented in page 52 depicts another view of adopting Web services pitfalls. With those driving forces mentioned for adopting Web services, and with the fact that SOA being the only software architecture capable on providing a process level integration, combining both Web services and SOA is almost a must. Most software capabilities will be provided and consumed as services. When the service is abstracted from the implementation it is possible to consider various alternative options for delivery and collaboration models [96].

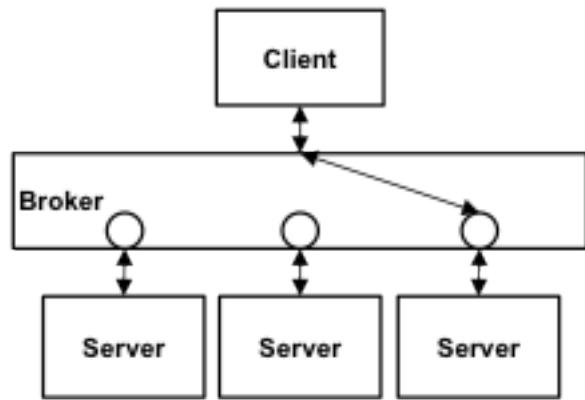


Figure 2.25: Broker Architecture

Table 2.6: Adopting Web services Pitfalls

Driving Forces	Restraining Forces
Interoperable networked applications	Different semantics in data sources
Emerging industry wide standards	Effect on operational systems and data requests
Support of Web services in products	
Easier exchange of data	
Reduced brittleness using tags	
Reduced development time	
Availability of external services	
Availability of training and tools	
Mergers and acquisitions	
Semantic translation	

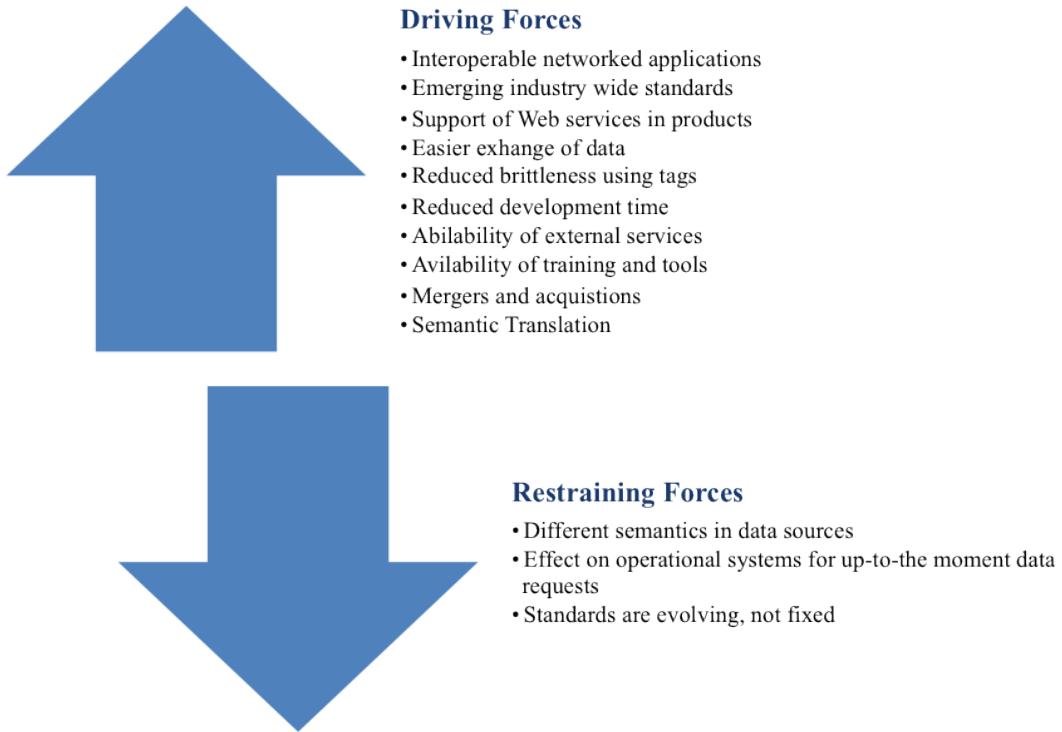


Figure 2.26: Adopting Web services Pitfalls

## 2.7 SOA Selected Topics

### 2.7.1 Relationship Between EA and SOA

SOA can be applied to different levels of scope: from the enterprise level to individual project level. SOA is integral to successful EA, and for some organizations it renews the value of EA. A service portfolio is one of the primary linkages of SOA to EA as enterprise architects play a major role in advancing the use and awareness across the enterprise of the service portfolio. The service portfolio is an enterprise asset, and enterprise architects must play a governance role in the use of services, vertically and horizontally in the enterprise. An architectural review board that includes enterprise architects will govern the service portfolio as an enterprise asset. EA is an architectural discipline that merges strategic business and IT objectives with opportunities for change and governs the resulting change initiatives. SOA represents a strategic business and IT objective that EA helps govern across multiple project instances. EA uses SOA principles and assets (e.g., service portfolio and SOA reference models) to integrate with business architecture, information architecture, application architectures, and infrastructure architectures. EA has an enterprise-wide focus and addresses both SOA and non-SOA aspects of the enterprise.

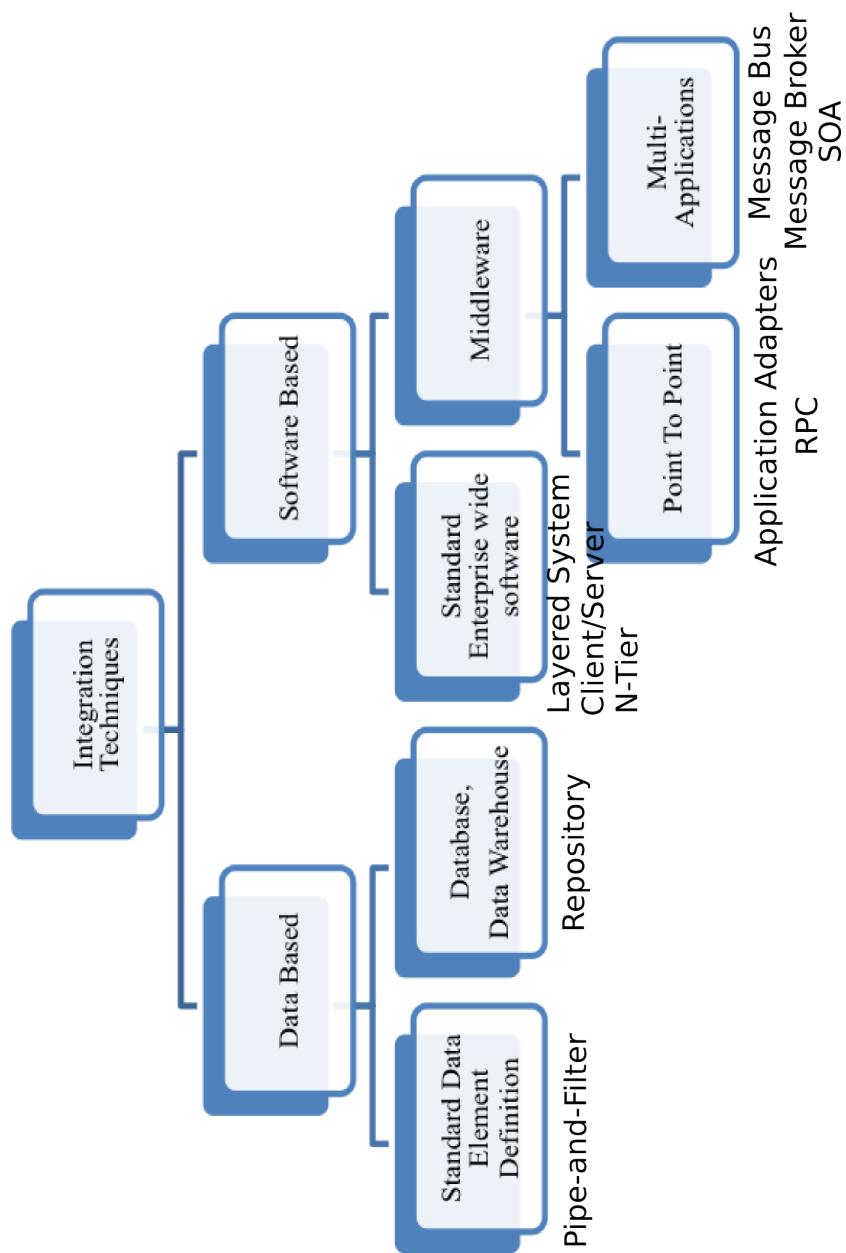


Figure 2.27: Integration Techniques and Software Architectures

### 2.7.2 Services Identification Methods

Services are optimally identified using three complementary techniques that provide a balance between tactical imperatives and strategic vision:

- Goal service modeling looks at business opportunities, strategy, and business goals to both confirm and validate that candidate services have been identified, which fulfill goals and enable the business strategy.
- Domain decomposition focuses on business process modeling, rules, information, and potential variability of services.
- Asset analysis addresses the reality that businesses have accumulated legacy systems and applications that must be integrated, enhanced, or leveraged. This bottom-up approach looks at the existing application portfolio and other assets that can be used in identifying candidates for service exposure. In contrast, goal service modeling combines the top-down (domain decomposition) and bottom-up (asset analysis) approaches and pulls them together into alignment.

### 2.7.3 CBA

Component Based Architecture (CBA) is a software architecture that is based on components. A component is a software object, meant to interact with other components, encapsulating certain functionality or a set of functionality. A component has a clearly defined interface and conforms to a prescribed behavior common to all components within an architecture [220]. A component model is probably used for the developing and executing of components. This model defines a framework, which defines structural requirements for connection- and composition options as well as behavior-oriented requirements for collaboration options to the components. Beyond that a component model provides an infrastructure which implements frequently used mechanism like persistence, message-exchange, security and versioning. The idea is to build exchangeable software units through clearly defined interfaces. Different manufactures offer platforms like DCOM, JavaBeans, Enterprise JavaBeans, and CORBA. There is no clear dividing line between SOA and Component Based Architecture. In principle SOA is the enhancement of Components: The individual services are single components, which can be linked to gain new business logic, new services or a new component. The big difference is the connection between and the possibilities of offering single services for third parties. For example, EJBs (especially Session Beans) can be designed to offer its business methods like services in a context free way. These services of this EJB can be used by other EJBs or clients. In a big company (or a coalition of parties gaining access to each others EJB's) single departments could offer their services (in the shape of the business methods of the EJBs) for other departments, so that the same effect could be achieved as with services supporting Simple Object Access Protocol (SOAP): The business methods of the EJBs represent the activities one department offers. Other departments could use these services for their belongings and perhaps use them in a way the business

method wasn't built for. So, this usage of Enterprise JavaBeans could be seen as service oriented too.

#### 2.7.4 Agile/OOAD and SOA

Many of the current system development methods focus on making IT more effective, cheaper, or faster, whereas SOA methods focus on making the business and IT more effective and faster. SOA methods provide prescriptive software engineering guidance by addressing the following:

- They provide guidance on how to identify and develop reusable business services that can be reconfigured to provide new business capability or repurposed to serve different business processes or market opportunities.
- They focus on how to reduce, using services, the lifetime cost of the application.
- They reduce system development activities, allowing for an accelerated system development process using services.
- They allow engineer applications to be built for change.

To identify and build reusable, reconfigurable, and flexible services as business assets, you must change existing methods to accommodate the identification, specification, and realization of five primary constructs:

- Business processes
- Services
- Components
- Information
- Rules/policies (and their flows)

A business service catalog will be the result, which will grow over time, project by project. Business services should be reused across applications and channels supporting vertical and horizontal business processes. SOA methods provide guidance on how to identify, specify, and realize reusable business services.

## 2.8 Summary

EA is the framework of business, IT, information, and software architectures. Software architecture design is an important step within system design because software architectures should satisfy non-functional as it satisfies functional system requirements. Non-functional requirements like integration, interoperability, and agility can be satisfied by architectural design, if considered. Software architecture satisfaction to integration; as one of the non-functional requirements, is used in this chapter as an indicator to the software architecture of non-functional requirements satisfaction.

The most common Software architectures are mapped to presented integration techniques; highlighting architectures driving and restraining forces, and its satisfaction level to system non-functional requirements. Figure 2.27 presented in page 53 summarizes the integration techniques and different software architectures presented as an enabler of each technique. Each one of the presented software architecture has its driving and restraining forces which presents strengths and weaknesses of each one. Web services based SOA achieved the maximum driving forces and satisfied almost all non-functional requirements addressed in the comparative study.

## 2.9 Review Questions

1. What are Enterprise Architecture Dimensions?
2. Discuss the importance of SW Architecture.
3. What are the advantages of Distributed SW Approaches?
4. List, and describe in brief the most common SW architecture patterns.
5. What are the main SW architectures integration techniques?
6. What are the advantages and disadvantages of each of the following techniques in SW integration:
  - (a) Standard Data Element Definition
  - (b) Middleware Integration Technique
  - (c) Standard Enterprise Wide Software
  - (d) Middleware
7. What is the relationship between EA and SOA?
8. What is CBA?

## 2.10 Exercises and Labs

In Labs, you shall be getting familiar with Java programming language, as it is the programming language we will be using to build, deploy, and consume different online Webservices and SOA systems presented in this book.

# **Chapter 3**

## **Service Oriented Architecture**

### **3.1 Introduction**

A web service is an instance of a more general notion of a service, which is defined as: “an act or performance offered by one party to another. Although the process may be tied to a physical product, the performance is essentially intangible and does not normally result in ownership of any of the factors of production”. The essence of a service, therefore, is that the provision of the service is independent of the application using the service. Service providers can develop specialized services and offer these to a range of service users from different organizations. SOA are a way of developing distributed systems where the system components are stand-alone services, executing on geographically distributed computers. Standard Extensible Markup Language (XML)-based protocols, such as SOAP and WSDL, have been designed to support service communication and information exchange. Consequently, services are platform and implementation-language independent. Software systems can be constructed by composing local services and external services from different providers, with seamless interaction between the services in the system. This chapter presents different topics related to SOA, including: SOA advantages, technologies, different methods of implementation, examples of IS based on SOA, and concludes with an important concept, that is ESB.

### **3.2 SOA**

SOA is defined as the policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer. Services can be invoked, published and discovered, and are abstracted away from the implementation using a single, standards based form of interface [98 - 101].

#### **3.2.1 SOA Advantages**

SOA advantages are categorized into implementation, and organizational benefits.

### Implementation Benefits

Implementation benefits satisfy the loose coupling objective. Using a resource only via its published service and not by directly addressing the implementation gives system capabilities of [102]:

- Changes to the implementation by the service provider should not affect the service consumer. Services are exposed via standard interfaces and are thought about as black boxes that changes within it do not affect consumers.
- Service consumer could choose an alternative instance of the same service type without modifying requesting application. As long as new service implements the same interface, theoretically there are no problems.
- Service consumer and service provider do not have to implement same technologies for implementation, interface, or integration when Web services are used.

### Organizational / Business Benefits

Businesses are dealing with two fundamental concerns: the ability to change quickly (agility), and the need to reduce costs [103]. Business must adapt quickly to internal factors such as acquisitions and restructuring, or external factors like competitive forces and customer requirements to remain competitive. Cost-effective, flexible IT infrastructure is needed to support the business. SOA can realize several benefits to help organizations succeed. Organizational/Business benefits of adopting SOA are:

- Leverage existing assets: by wrapping existing software applications as services instead or rebuilding new solutions from scratch.
- Easier to integrate: integration on service level in order to satisfy business processes integration presents the highest integration technique that solved many problems as depicted in chapter three.
- More responsive and faster time-to-adapt (Agility): the ability to compose new services out of existing ones provides a distinct advantage to an organization that has to be agile to respond to demanding business needs. Leveraging existing components and services reduces the time needed to go through the software development life cycle leading to rapid development of new business services and allows an organization to respond quickly to changes. SOA provides the flexibility and responsiveness that is critical to businesses agility.
- Reduce cost and increase reuse: with core business service exposed in a loosely coupled manner, they can be more easily used and combined based on business needs. This means less duplication of resources, more potential for reuse, and lower costs.

## 3.3 SOA Technologies

SOA implementations include: Software Agents, and Web services.

### 3.3.1 Software Agents

Software Agent is a computer system that is situated in some environment and is capable of autonomous actions in this environment in order to meet its design objectives [104,105]. Different SOA implementations using different software agents are presented in [106 – 110]. SOA implementation via mobile agents technology can be found in [108]. The main idea is about having one or more software agents perform certain task(s), those tasks can be exposed as services that compose SOA. Software agents have many classification criteria. Agents can be classified as either Desktop, Internet, or Intranet agents according to their action environment [111]. Many internet related agents can be classified into subcategories as presented in [112]. IAs can be classified into Simple reflex agents, Model-based reflex agents, Goal-based agents, and Utility-based agents [113]. Agents can be also classified by application types, and by characteristics [114]. Software Agents have characteristics that make them suitable to perform complex functionality. Characteristics include: Autonomy, Interactivity, Reactivity, Proactivity, Intelligence, and Mobility [114]. Agent is autonomous; it is capable of acting on its own. An agent is goal oriented, collaborative, and flexible, so, it must be autonomous. Agents are designed to interact with other agents, humans, or software programs (Interactivity). Instead of making a single agent conduct several tasks, additional agents can be created to handle un-delegated task. Agents perceive environment via preceptors [113] and respond to changes (Reactivity). Agents do not just act in response to their environment, but agents are able to exhibit goal-directed behavior by taking an initiative (proactive). Agent may need mobility to work on different machines. An agent with this capability is called mobile agent, it can transport itself across different system architectures and platforms, and is far more flexible than those that cannot. Many electronic commerce agents are mobile [114]. Mobile agent is an executing program that can migrate during execution from machine to machine in a heterogeneous network [110].

### Multi-Agent System

Multi-Agent Systems (MASs) are becoming increasingly important: as a scientific discipline, as a software engineering paradigm, and as a commercially viable and innovative technology [115]. A Multi Agent is any system that contains [116]:

- Two or more agents;
- At least one autonomous agent; and
- At least one relationship between two agents where one satisfies goal of the other.

Many Multi-Agent frameworks have been presented. Some of Multi-Agent frameworks proposed from 2005 till now include works presented in [117 – 122]. MAS characteristics are [123]:

- Each agent has incomplete capabilities to solve problems.
- There is no global system control.

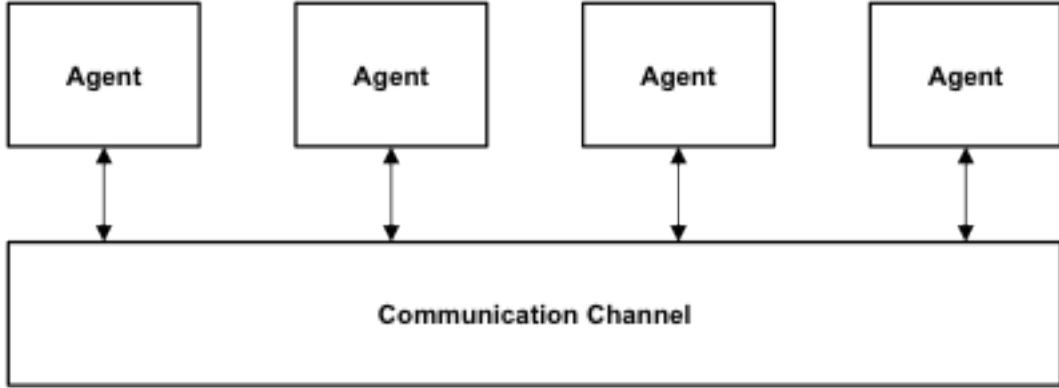


Figure 3.1: Flat MAS Architecture

- Data is decentralized.
- Computation is asynchronous.

### Common MAS Architectures

Agents can be organized in different MAS architectures. MAS architectures include: flat, fixed hierarchy, Subsumption, and Modular [124].

- Flat MAS Architecture: Agents directly contact all other agents. The system is either closed, so all agents know the location of all other agents, or open, which requires agent location mechanism. Agent location mechanism can be one of message based architectures. Figure 3.1 presented in page 60 depicts Flat MAS Architecture.
- Fixed Hierarchy MAS Architecture: Agents communicate only to agents directly above or below them in the hierarchy. Hierarchy is fixed leading to defects in systems scalability. Figure 3.2 presented in page 61 depicts hierarchical MAS architecture.
- Subsumption MAS Architecture: Agents can contain other agents. Highly performance and wide scalability is the main advantage of subsumption architecture. Figure 3.3 presented in page 61 shows subsumption MAS architecture.
- Modular MAS Architecture: Agents are grouped in modules as presented in figure 3.4 presented in page 62. Communication takes place between agents composing the module, and agents of different modules. Each module is specified with one or more task(s). Presenting systems as collection of modules facilitated problem solving.

Multi-Agent architecture standards were attempted in order to force MAS standardization and global integration. Knowledge Query and Manipulation Language (Knowledge Query and Manipulation Language (KQML)) was presented in order to support knowledge sharing among agents [125]. Knowledge Interchange Format

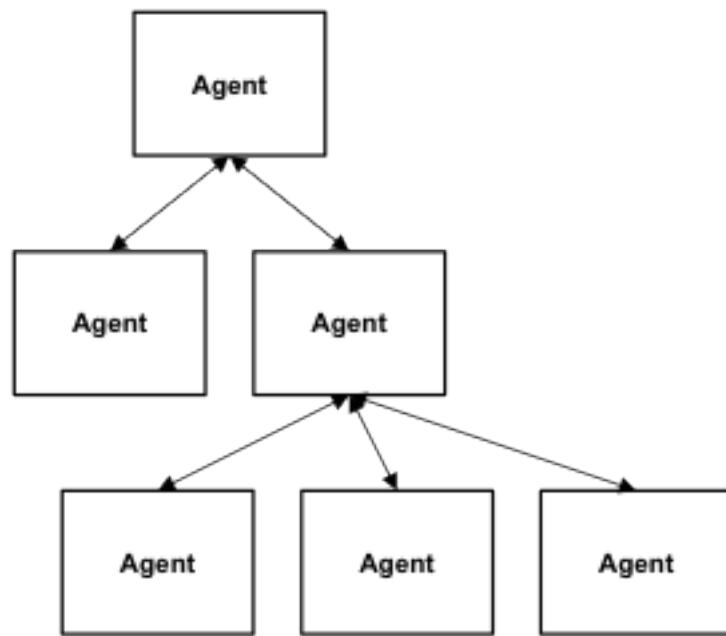


Figure 3.2: Hierarchical MAS Architecture

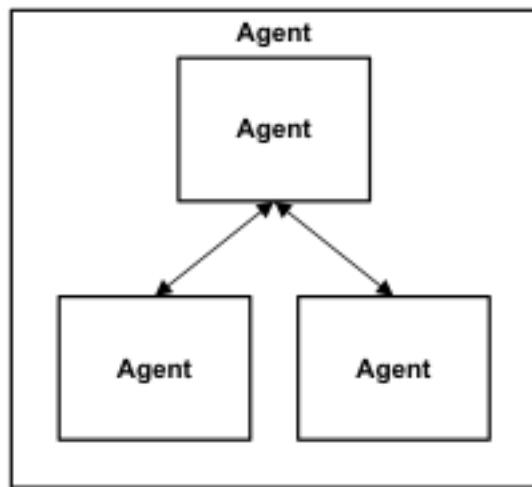


Figure 3.3: Subsumption MAS Architecture

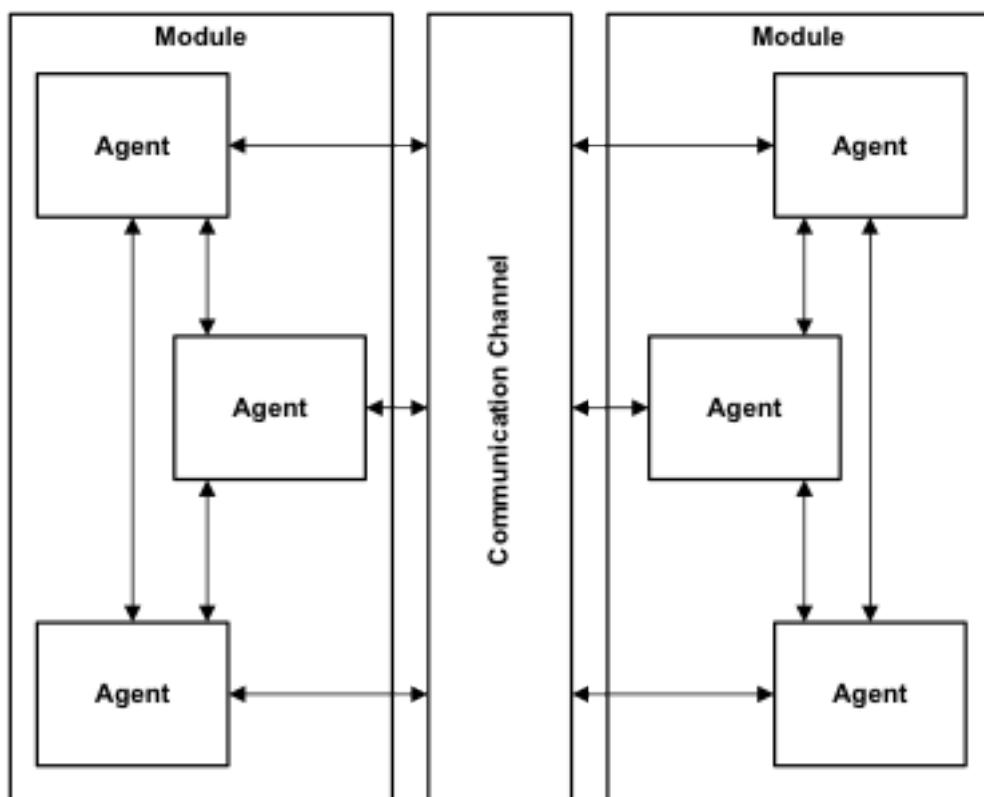


Figure 3.4: Modular MAS Architecture

(Knowledge Interchange Format (KIF)) is a computer-oriented language for the interchange of knowledge among disparate programs [126]. The OMG group proposed a reference model as an attempt to standardize the development of agent technologies [127]. Knowledgeable Agent-oriented System (KAoS) is described as "an open distributed architecture for software agents." The KAoS architecture describes agent implementations, and elaborates on the interactive dynamics of agent-to-agent messaging communication by using conversation policies [128]. The Foundation for Intelligent Physical Agents (Foundation for Intelligent Physical Agents (FIPA)) is a multi-disciplinary IEEE standardizing group pursuing the standardization of agent technology. FIPA's approach to MAS development is based on a "minimal framework for the management of agents in an open environment" [129]. Unfortunately, as a result for all the standardization effort, there were no universally accepted commercially supported standard yet.

### 3.3.2 Webservices

Web services are not just an integration solution. Web services technology is currently the major implementation of SOA [130]. All service models are specific to the WS-Coordination specification and related protocols [99]. Web services adoption by organizations solved many problems. Web services is a general framework that expedites the sharing of heterogeneous data and software resources dispersed on the internet. The standard-based resource sharing and platform-neutral characteristics of web services have motivated many organizations to apply the technology in diverse areas, such as SCM, virtual enterprise, homeland defense, e-government, and e-business [131]. Software agents can consume Web services as presented in [132].

## 3.4 Webservices Technology

Web services are applications that use standard transports, encoding, and protocols to exchange information [133]. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. Web service has an interface described in a format that machines can process (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using Hyper Text Transfer Protocol (HTTP) with XML serialization in conjunction with other Web-related standards [134].

### 3.4.1 Webservices Architecture

Web service technology is based on open technologies; this provides broad interoperability among different vendor solutions [103]. Figure 3.5 presented in page 64 shows the basic Web services architecture, that consists of specifications (SOAP, WSDL, and Universal Description Discovery and Integration (UDDI)) that support the interaction of a Web service requester with a Web service provider and the potential discovery of the Web service description [135]. The provider typically publishes a WSDL description of its Web service, and the requester accesses the description

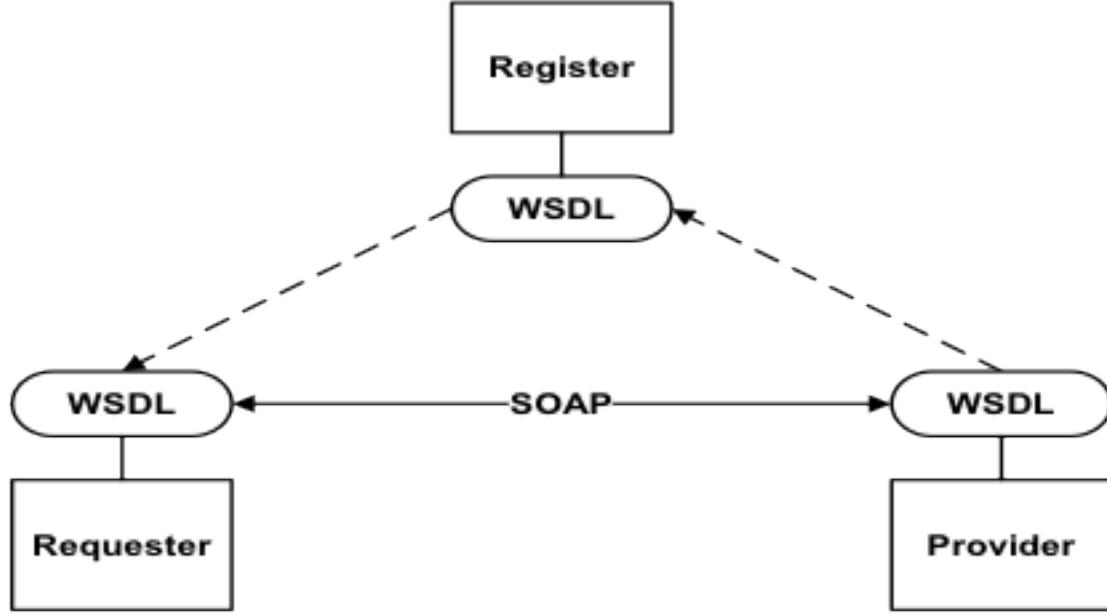


Figure 3.5: Basic Webservices' Architecture

using a UDDI or other type of registry, and requests the execution of the provider's service by sending a SOAP message to it.

Web service protocols cover all aspects of SOAs, from the basic mechanisms for service information exchange (SOAP) to programming language standards (WS-BPEL). These standards are all based on XML, a human and machine-readable notation that allows the definition of structured data where text is tagged with a meaningful identifier. XML has a range of supporting technologies, such as XSD for schema definition, which are used to extend and manipulate XML descriptions. Erl (2004) provides a good summary of XML technologies and their role in web services. Briefly, the key standards for web SOAs are as follows:

1. SOAP This is a message interchange standard that supports the communication between services. It defines the essential and optional components of messages passed between services.
2. WSDL is a standard for service interface definition. It sets out how the service operations (operation names, parameters, and their types) and service bindings should be defined.
3. WS-BPEL This is a standard for a workflow language that is used to define process programs involving several different services. A service discovery standard, UDDI, was also proposed but this has not been widely adopted. The UDDI standard defines the components of a service specification, which may be used to discover the existence of a service. These include information about the service provider, the services provided, the location of the WSDL description of the service interface, and information about business relationships. The intention was that this standard would allow companies to set up registries with UDDI descriptions defining the services that they offered.

### 3.4.2 Webservices Key Features

Some of Web services key features are [103]:

- Self-contained: On the client side, a programming language with XML and HTTP client support is the only requirement. On the server side, a Web server and a servlet engine are required.
- Self-describing: format and content of request and response messages (loosely coupled application integration) is what only matters. Messages are descriptive, not instructive.
- Modular: Web services are a technology for deploying and providing access to business functions over the Web; J2EE, CORBA, and other standards are technologies for implementing Web services.
- Web published, located, and invoked: by applying the previously mentioned standards SOAP, XML, WSDL, and UDDI, Web services can be published, located, and invoked via internet.
- Language independent: interaction between a service provider and a service consumer is designed to be completely platform and language independent. Interaction requires a WSDL document to define the interface and describe the service, along with a network protocol (usually HTTP).
- Interoperable: Because service provider and service consumer do not share same platforms or languages; only communicate via standard protocols, interoperability is achieved.
- Inherently open and standards based: XML, SOAP, WSDL and HTTP are the technical foundation for Web services. A large part of the Web service technology has been built using open source projects.
- Dynamic: dynamic e-business can become a reality using Web services because, with UDDI and WSDL, the Web service description and discovery can be automated.
- Composable: Web services can be aggregated to complex ones.

### 3.4.3 Webservices Advantages

Web services advantages arise from Web services key features. Web services were designed to satisfy all software requirements arose over the years, and avoid all drawbacks of previous technologies. Web services advantages include [133,136, 137, 138, 139]:

- Interoperability: Software interoperability is the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units. Web services are interoperable.

- Language agnostic: Web services are neither based on a programming language nor on a programming data model. Any Programming language can be used to implement Web services.
- Relatively simple: Web services are simple to build, expose, and consume because they are:
  - Based on Web technologies: so, they are scalable, and easily to control security features.
  - Do not necessarily require a huge framework in memory: A small amount of code could expose a service as a Web Service, besides Web services can be used easily in today's Web Interface.
- Loosely coupled applications: applications based on Web services are loosely coupled applications by default. Loose coupling is a design goal that describes a resilient relationship between two or more applications, systems, or organizations with some kind of exchange relationship [140]. In Web services, exchange relationship is established via request/response messages.
- Support of software industry leaders: Hundreds of IT vendors have participated in the Web services standardization process under the sponsorship of the World Wide Web Consortium (World Wide Web Consortium (W3C)), Organization for the Advancement of Structured Information Standards (OASIS) and Web Services Interoperability Organization (WS-I). IT vendors include, not only: Microsoft, Oracle, bea, SAP, IBM, and Sun..
- Integration with the World Wide Web: Web services tended to integrate with World Wide Web from the very first beginning in order to use internet as the infrastructure. Internet integration provides the most advantage of adopting Web services within applications.

#### 3.4.4 Enhanced Standards

A number of companies, such as Microsoft, set up UDDI registries in the early years of the 21st century but these have now all closed. Improvements in search engine technology have made them redundant. Service discovery using a standard search engine to search for appropriately commented WSDL descriptions is now the preferred approach for discovering external services. The principal SOA standards are supported by a range of supporting standards that focus on more specialized aspects of SOA. There are a very large number of supporting standards because they are intended to support SOA in different types of enterprise application. Some examples of these standards include the following:

1. WS-Reliable Messaging, a standard for message exchange that ensures messages will be delivered once and once only.
2. WS-Security, a set of standards supporting web service security including standards that specify the definition of security policies and standards that cover the use of digital signatures.

3. WS-Addressing, which defines how address information should be represented in a SOAP message.
4. WS-Transactions, which defines how transactions across distributed services should be coordinated.

Current web services standards have been criticized as being ‘heavyweight’ standards that are over-general and inefficient. Implementing these standards requires a considerable amount of processing to create, transmit, and interpret the associated XML messages. For this reason, some organizations, such as Amazon, use a simpler, more efficient approach to service communication using so-called RESTful services.

### 3.4.5 RESTful Webservice

The set of web services protocols that have been developed support a very general model of web services and have taken into account important enterprise issues such as security, reliability and transactions. They allow services from different providers to be dynamically linking and executed.

However, an inevitable consequence of this generality is that there is significant processing overhead in interpreting the XML documents that are exchanged by web services. This can mean that the performance of service-oriented systems is reduced. This has led to alternative approaches to service implementation which use simpler protocols and which are usable in situations where the generality offered by ‘big’ web services is not required.

REST (REpresentational State Transfer) is an architectural style based on transferring representations of resources from a server to a client. It is the style that underlies the web as a whole and has been used as a much simpler method than SOAP/WSDL for implementing web services [246]. A RESTful web service is identified by its URI (Universal Resource identifier) and communicates using the HTML protocol. It responds to HTML methods GET, PUT, POST, and DELETE and returns a resource representation to the client. Simplistically, POST means create, GET means read, PUT means update, and DELETE means delete. RESTful services involve a lower overhead than so-called ‘big web services’ and are used by many organizations implementing service-based systems that do not rely on externally provided services.

So-called RESTful web services follow the REST (REpresentational State Transfer) architectural style, with all communications based on the simple HTTP protocol. Services are modeled as resources (e.g. a parts catalogue) and respond to HTTP methods GET, POST, PUT and DELETE. The service is addressed using its URI (Universal Resource Identifier).

The information exchanged by RESTful services is the resource representation which may be XML, but, in principle, other representations such as JSON (Javascript Object Notation) may also be exchanged.

The HTTP methods are usually interpreted as [244]:

- POST=Create
- GET = Read

- PUT = Update or Create
- DELETE = Delete

The key benefit of RESTful services is that they are simpler and more efficient than 'Big' web services. The amount of processing overhead is reduced, which is particularly important for simple services than implement small increments of functionality.

SOAP and REST are not incompatible and it is possible for web services providers to offer both SOAP and REST service interfaces. Major providers such as Amazon offer both and their experience is that the REST interface is often preferred by developers.

### 3.5 An in-car Information System

SOAs are loosely coupled architectures where service bindings can change during execution. This means that a different, but equivalent version of the service may be executed at different times. Some systems will be solely built using web services and others will mix web services with locally developed components. To illustrate how applications that use a mixture of services and components may be organized, consider the following scenario: An in-car IS provides drivers with information on weather, road traffic conditions, local information, and so forth [246]. This is linked to the car radio so that information is delivered as a signal on a specific radio channel. The car is equipped with GPS receiver to discover its position and, based on that position, the system accesses a range of information services. Information may then be delivered in the driver's specified language. Figure 3.6 presented in page 69 illustrates a possible organization for such a system. The in-car software includes five modules. These handle communications with the driver, with a GPS receiver that reports the car's position and with the car radio. The Transmitter and Receiver modules handle all communications with external services. The car communicates with an external mobile information service that aggregates information from a range of other services, providing information on weather, traffic information, and local facilities. Different providers in different places offer these services, and the in-car system uses a discovery service to locate appropriate information services and bind to them. The discovery service is also used by the mobile information service to bind to the appropriate weather, traffic, and facilities services. Services exchange SOAP messages that include GPS position information used by the services to select the appropriate information. The aggregated information is then sent to the car through a service that translates that information into the driver's preferred language.

### 3.6 Webservices as SOA enabler

Web services are relatively new technology that have received wide acceptance as an important implementation of SOA [103]. Web services is not SOA, and SOA is not Web services. It is important to differentiate between Web services; as a technology, and SOA as a design pattern. Web services enabled the maximum advantages of

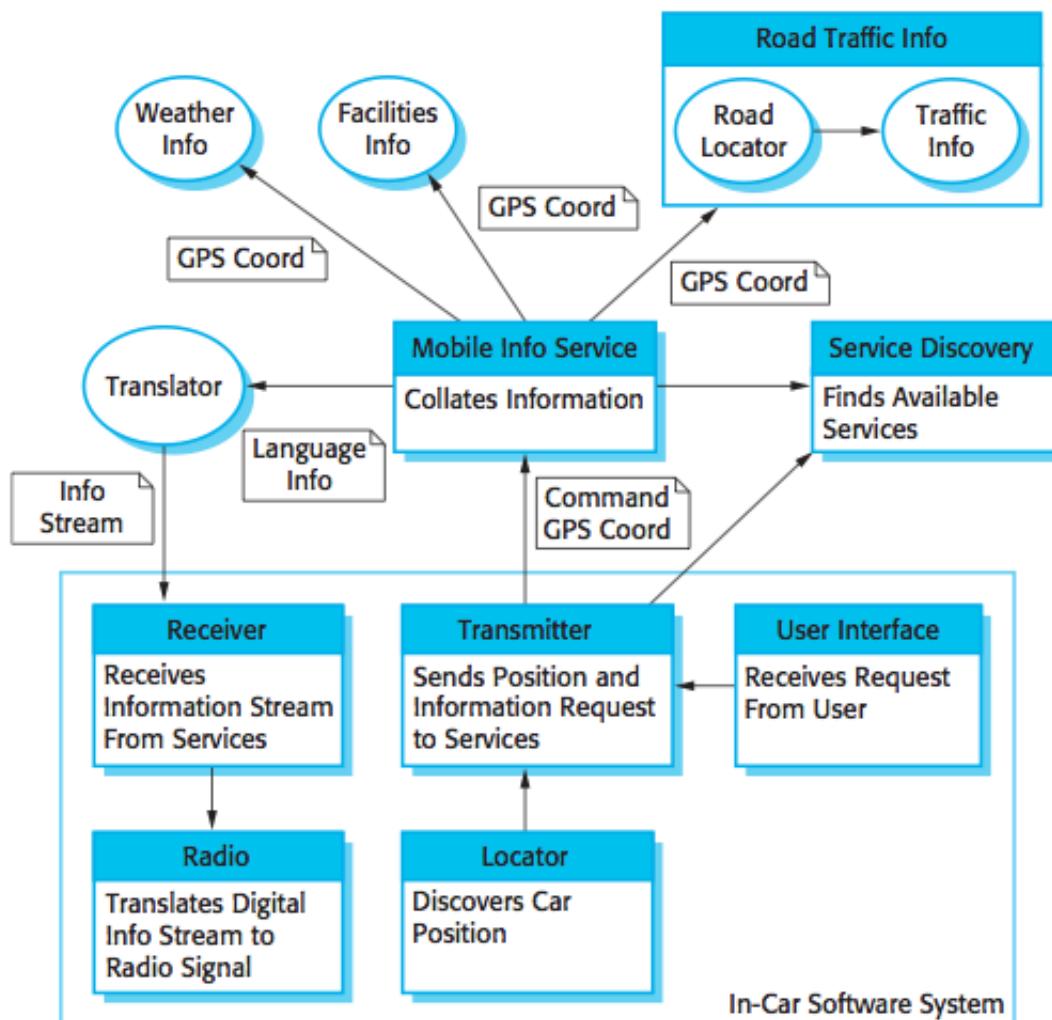


Figure 3.6: Service-based in-car IS [246]

SOA, this is mainly why Web services is the main implementation and enabler of SOA. SOA presents systems as collection of services to be published, and consumed when required. When exposed services are Web services, system gains the advantage of Web services standardization, like interoperability. But still systems need to be studied carefully, and analyzed intensively to determine services will be exposed, suitable services interfaces, and design the required databases. SOA is not about exposing and consuming Web services to facilitate interoperability or integration or other requirements, but it is rather the science, art, and practice of building loosely coupled fine granular services to form applications.

## 3.7 Enterprise Service Bus

ESB provides a comprehensive, scalable way to connect a large number of applications without the need for each pair of applications to make a direct connection. Such a direct connection between two applications is called a point-to-point connection. Note that even in the case of Web Services, the connection between the service consumer application and the service provider application is “point to point.” The point-to-point connection approach does not scale well because the number of applications involved in the integration increase; therefore, this integration approach is not suitable for a large enterprise where a large number of applications need to be integrated.

### 3.7.1 ESB

ESB can be defined in two contexts: design and run-time. As a design-time context, ESB is an architectural pattern that has multiple motivations:

- Supports large numbers of service interactions in a manageable way
- Provides support for advanced service interaction capability, such as transactions, store and forward, infrastructure services, security, quality of service, and so on
- Supports a variety of interaction styles such as synchronous request/response, messaging, publish/subscribe, and events
- Provides a robust, manageable, distributed integration infrastructure consistent with the principles of SOA
- Supports service routing and substitution, protocol transformations, and other message processing
- Supports both Web Services and traditional EAI communication standards and technologies

The ESB pattern can be implemented in one or more of the following hybrids:

- EAI technology

- Messaging technology
- Technology rebranded or classified as an ESB product
- “Gateway” technology
- Appliance technology
- Bespoke

The ESB mediates characteristics of interactions between service requestors and service providers. The ESB enables the substitution of service providers or implementations transparent to service requesters. The ESB supports a variety of means to attach requesters and providers, and it allows intermediary services to be sequenced between requesters and providers. The ESB may provide a broad set of capabilities dependent on business needs and implementation in several areas, including network communications, integration, security, message processing, quality of service, and service management.

### 3.7.2 ESB vs. Integration Technology

An ESB is a connectivity infrastructure for integrating applications and services, in contrast to EAI, which focuses on the integration of applications. ESB infrastructure differs from EAI in the following aspects:

- ESB infrastructure is more than integration because it performs routing of messages between services, converts transport protocols between consumers and providers, transforms message formats between requesters and providers, and distributes business events from disparate sources. Although EAI solutions can address all of these aspects, integration technologies are usually much more narrowly focused. ESB handles a variety of interaction patterns, including events.
- ESB requires management such that the status of a business transaction can be assessed. Has the transaction completed? How long did it take? Did the process step complete? Although the ESB will not be the only technology to assist in business activity management, it will be a part.
- ESB product technologies will be federated such that various technologies (e.g., gateways and appliances) can be used to fulfill a single purpose and provide a single interface to applications. Heterogeneous platforms can be supported allowing different ESB technologies to operate as a single logical ESB.

### 3.7.3 ESB Functionality

Business-and IT-level events are used by service-level automation to enforce the policies associated with the business services they host. A service bus can be used to collect, aggregate, and evaluate those events for presentation to business process participants in business activity management scenarios. The ESB provides a broad set of capabilities dependent on scenarios:

- Communications: Routing, addressing, protocol, publish/ subscribe interactions, asynchronous interactions, event handling, and other features.
- Service Interaction: Interface definition, SOAP, REST, and other features
- Integration: DB, legacy, and middleware connectivity; service aggregation, application server connectivity, protocol transformation, and other features
- Quality of Service: Transactions and delivery assurance, and other features
- Security: Authentication, authorization, non-repudiation, confidentiality, standards support, and other features
- Service level: Performance, throughput, availability, scalability, and other features
- Message Processing: Message and data transformations, intermediaries, content-based routing, and other features
- Management and Autonomic: Service provisioning and registration, logging, metering, monitoring, system management, and other features
- Appliances: Parsing, compression, routing, and other features
- Infrastructure Intelligence: Business rules, policy-driven behavior, pattern recognition, and other features

### 3.7.4 BizTalk Server

Microsoft BizTalk Server uses adapter technology to connect disparate entities and enable the integration of data, events, processes, and services. An entity may be an application, department, or even an altogether different organization that needed to be able to share information with. A software adapter is typically used when we need to establish communication between two components that do not natively collaborate. BizTalk Server adapters are built with a common framework which results in system integration done through configuration, not coding. Traditionally, BizTalk Server has solved problems in three areas. First, BizTalk Server acts as an EAI server that connects applications that are natively incapable of talking to each other. The applications may have incompatible platforms, data structure formats, or security models. For example, when a new employee is hired, the employee data in the human resources application needs to be sent to the payroll application so that the new employee receives his/her paycheck on time. Nothing prevents you from writing the code necessary to connect these disparate applications with a point-to-point solution. Many organizations choose to insert a communication broker between these applications. Some of the benefits that you would realize from such an architectural choice include:

- Loose coupling of applications where one does not have a physical dependency on the other

- Durable infrastructure that can guarantee delivery, and queue messages during destination system downtime
- Centralized management of system integration endpoints
- Message flow control such as in-order delivery
- Insight into cross-functional business processes through business activity monitoring

BizTalk Server solves a second problem by filling the role of business-to-business (B2B) broker that facilitates communication across different organizations. BizTalk supports B2B scenarios by offering Internet-friendly adapters, industry-standard EDI message schema, and robust support for both channel- and message-based security. The third broad area that BizTalk Server excels in is Business Process Automation (BPA). BPA is all about taking historically manual workflow procedures and turning them into executable processes. For example, consider the organization that typically receives a new order via email and the sales agent manually checks inventory levels prior to inserting the order into the Fulfillment System. If inventory is too low, then the sales agent has to initiate an order with their supplier and watch out for the response so that the Inventory System can be updated.

## 3.8 SOA Selected Topics

### 3.8.1 SOA service vs. Webservice

SOA services can be realized as web services, but not all web services are equal to SOA services. Web services represent the use of both a published standard and a set of technologies for invocation and interoperability. SOA services are services that fulfill a key step or activity of a business process and can be described as business services and are often exposed as web services. We can distinguish between SOA service and Webservice in design time, and runtime.

- In design, we identify and specify a service that provides the design, or we identify and specify interfaces that include method specifications. The combination of the definition of the method and the interface at design time is what we refer to as a service from an SOA perspective. Use cases can be used to capture the functional requirements for a service.
- In runtime, In an SOA, business processes, activities, and workflow are broken down into constituent functional elements called services. They can be accessed and used directly by applications, or they can be mixed and matched with other services to create new business capabilities. Business services or SOA services are reusable business capabilities. Examples in banking include open account or change address. For transportation, it might be get reservation or hold reservation, and with loan processing, get loan, apply for loan, and update address are examples of business services.

### 3.8.2 SOA vs. DCE and CORBA

The difference between SOA and earlier approaches, such as DCE or CORBA, has a lot to do with standardization. This was initially brought about through the use of Web services and Web services standards. WSDL, SOAP, and other Web services standards enable the industry to converge around a common infrastructure model for runtime. Different vendors would have different implementations, but they would generally conform to the base models. The Web Services Interoperability Organization (WS-I) is an example of an effort to pull things together so that different vendor implementations are consistent and compliant to the degree possible. Web services, unlike earlier approaches, built upon existing and deployed infrastructure as it took advantage of the Internet, resulting in less cost for adoption and reduced risk. Agreed upon versus *defacto* standards is a huge difference between SOA and earlier approaches. Web services standards have been the genesis of SOA, and Web services are more language independent than object-oriented technology integration approaches, which are often language specific (e.g., Java, C, or Smalltalk). XML is language neutral and renders naturally into languages of choice: COBOL, C++, Java, or others. XML and Web services standards, such as WSDL, have improved flexibility than approaches such as CORBA or RPC found in DCE, where changes and additions to the data structures often resulted in breakages of the code that used such structures. In contrast, XML does not use offsets, and it is therefore possible to reorder or add data elements without a break in older versions. Web services also use one type space for interfaces, and that type is XML. The other approaches use one type for databases (e.g., SQL), another for in-flight messages (e.g., Internet Inter-Orb Protocol, IIOP), and another Interface Definition Language (IDL; e.g., CORBA). One approach versus three creates an easier-to-use developer toolkit and application programming interface (API) set, and it makes the code base less brittle and easier to change. Contracts that provide a valid sequence of interaction with a service and policies that govern the nonfunctional characteristics of a service have augmented the notion of the interface found in earlier approaches. One of the most important advances that SOA provides is that the set of services that are required by organizations and captured in the service portfolio provides a business language between business and IT to discuss fulfillment of business needs. A service portfolio is governed as an enterprise asset and used by business and IT stakeholders. This represents a big change from CORBA and RPC, which are something IT uses. SOA and services provide much more than programmatic notions; they provide services at a granularity that the business understands and can use.

### 3.8.3 SOA vs. CBA

The differences between a service-oriented approach and a component-oriented approach to distributed systems architectures are [245]:

1. Services can be offered by any service provider inside or outside of an organization. Assuming these conform to certain standards (discussed below), organizations can create applications by integrating services from a range of providers. For example, a manufacturing company can link directly to services provided by its suppliers.

2. The service provider makes information about the service public so that any authorized user can use the service. The service provider and the service user do not need to negotiate about what the service does before it can be incorporated in an application program.
3. Applications can delay the binding of services until they are deployed or until execution. Therefore, an application using a stock price service (say) could dynamically change service providers while the system was executing.
4. Opportunistic construction of new services is possible. A service provider may recognize new services that can be created by linking existing services in innovative ways.
5. Service users can pay for services according to their use rather than their provision. Therefore, instead of buying an expensive component that is rarely used, the application writer can use an external service that will be paid for only when required.
6. Applications can be made smaller (which is important if they are to be embedded in other devices) because they can implement exception handling as external services.
7. Applications can be reactive and adapt their operation according to their environment by binding to different services as their environment changes.

### 3.8.4 WSDL vs. REST

SOA is an architectural style, and that the use of Web services is only one way to implement or realize this architectural style. In fact, there may be a set of different implementations of a service portfolio. Suppose, for instance, that the service portfolio has 100 services; 50 of them may be implemented using Web services, and the rest may be implemented using a combination of Representation State Transfer (REST) and Simple Object Access Protocol (SOAP). Or, an architectural decision can be made to use other Java or .NET mechanisms that do not use REST or SOAP. SOA does not constrain a solution to using Web services, SOAP, or REST. However, using Web services is a best practice, and using WSDL is the fundamental aspect of what makes a Web service not SOAP or REST. There are different patterns for SOA implementation patterns. Not all of them promote SOA best practices. Pattern 1 is a tightly coupled interaction, which does not use a Services Layer but may use a service interface in the operational system. In this pattern, it's unlikely that SOAP or REST is used, and most likely a proprietary messaging interface is the transport of choice. In pattern 2, a service component is used with a service interface but no Services Layer is in place. In pattern 3, a Services Layer is present, but the service maintains state in its interactions with the service component and Operational Systems Layer. In pattern 4, a packaged solution provides a service interface and most likely makes REST or SOAP options available. In pattern 5, a business state machine might be in use directly interacting with service interfaces. In pattern 6, a Business Process Layer and Services Layer are leveraged. The primary reason

for illustrating the various SOA interaction patterns is to illustrate that there is no right or wrong choice when picking SOAP or REST. Instead, choices may be architecturally weak or strong based on the interaction pattern needed, the quality of service (QoS) attributes that must be achieved, the examination of any constraints imposed by existing operational systems, and the determination of where the greater flexibility lies. One can then make an architectural choice: SOAP, REST, or another interaction. The use of SOAP often is equated as using Web services when in fact Web services can also be REST-based Web services. In SOA adoption, the interaction patterns are architectural decisions along with whether to use SOAP, REST, or other service.

### 3.8.5 Fundamental Types of Services

There are three fundamental types of service that may be identified [122]:

1. Utility services These are services that implement some general functionality that may be used by different business processes. An example of a utility service is a currency conversion service that can be accessed to compute the conversion of one currency (e.g., dollars) to another (e.g., euros).
2. Business services These are services that are associated with a specific business function. An example of a business function in a university would be the registration of students for a course.
3. Coordination or process services These are services that support a more general business process which usually involves different actors and activities. An example of a coordination service in a company is an ordering service that allows orders to be placed with suppliers, goods accepted, and payments made.

### 3.8.6 REST

REST is an architectural style that uses the analogy of the state transition diagram and maps that to an application having a set of resources that are connected by state transitions or links. A network of nodes such as the World Wide Web provides an ability to click a link and be transferred to the resource for that link. For example, if you are on a Web page that is a representation of a resource, you are then considered to be in a particular state when you are on that Web page. By clicking a link within that initial Web page that represents the current state, you convey the intent to transition to another Web page, which is really a resource representation of the next state. Thus, a network of states and transitions is traversed by selecting the appropriate link to the appropriate resource. This resource representation puts you, accessing the resource, into a particular state. When you click a link on that page, you get a representation of another resource. When you continue to access various resources by following the links, you keep changing state. This is essentially what is meant by REST.

### 3.8.7 Web services vs. REST

There are three main differences between Web services and REST:

- Service oriented versus resource oriented. With Web services, you are requesting a service (and so this is service oriented). With REST, however, you are implying you are looking for a specific type of resource (and so this is considered to be resource oriented).
- Use of HTTP. REST typically uses HTTP as the transport, whereas SOAP has no restriction on the use of a particular transport. Many people use REST-style interactions using HTTP.
- Quality of service. When you use REST, all the QoS parameters must be provided by the transport itself. In Web services, a significant number of specifications must be written to support QoS options, using the WS-\* set of standards.

## 3.9 Summary

A web service is an instance of a more general notion of a service, which is defined as: “an act or performance offered by one party to another. Although the process may be tied to a physical product, the performance is essentially intangible and does not normally result in ownership of any of the factors of production”. The essence of a service, therefore, is that the provision of the service is independent of the application using the service. Service providers can develop specialized services and offer these to a range of service users from different organizations. SOA are a way of developing distributed systems where the system components are stand-alone services, executing on geographically distributed computers. Standard XML-based protocols, such as SOAP and WSDL, have been designed to support service communication and information exchange. Consequently, services are platform and implementation-language independent. Software systems can be constructed by composing local services and external services from different providers, with seamless interaction between the services in the system. This chapter presented different topics related to SOA, including: SOA advantages, technologies, different methods of implementation, examples of IS based on SOA, and concluded with an important concept, that is ESB.

## 3.10 Review Questions

1. What is SOA?
2. What are SOA advantages?
3. What are SOA main technologies?
4. What is Webservice?

5. What are Webservices advantages?
6. What is Webservice Enhanced Standards?
7. What is ESB?
8. What is REST?
9. What are the fundamental types of Services?

### **3.11 Exercises and Labs**

In Labs, you shall be getting familiar with Java programming language, as it is the programming language we will be using to build, deploy, and consume different online Webservices and SOA systems presented in this book.

# Chapter 4

## BPM and SOA

### 4.1 Introduction

Business Agility is a relatively new paradigm painted as a solution for maintaining competitive advantage during times of uncertainty and turbulence in the business environment [238]. Business improvement approaches, such as Total Quality Management (TQM), Business Process Reengineering (BPR), and Workflow Management Systems (WfMS) failed to satisfy business agility requirements due to different reasons. Three approaches realized the need for BPM [84]. Business process is a series of inter-related activities that cross functional enterprise boundaries with individual inputs and outputs [129, 99]. BPM is a systematic, structured approach to analyze, improve, control, and manage processes [84]. BPMS is an EIS that supports designing, administrating, and improving the business processes [147]. EISs suffer from lack of agility and inefficiency [103]. Lack of agility refers to enterprises inability to match business requirements, especially new ones, onto current IT infrastructure and ISs architecture. Inefficiency refers to the ISs development process that costs too much when compared to the actual output. SOA as a design pattern presents IT infrastructure and ISs architecture as loosely coupled, fine granular services that can address business requirements once they are presented either by adding new services or modifying existing ones. SOA also addresses enterprises information systems inefficiency by enhancing reusability, thus theoretically, shortening ISs development time and effort required. Two main approaches for designing and implementing SOA based ISs are software agent based SOA, and Web services based SOA. Both approaches are presented in this chapter.

This chapter goes as follows: section two presents business agility principles, and framework. Section three presents business process aspect of business agility, and the need for BPMS. Section four presents SOA as the solution to technical agility as one of business agility aspects. Section five presents the proposed model aligning BPMS and SOA highlighting potential enterprises' benefits and advantages of this align regarding business agility.

## 4.2 Business Agility

Business agility is a concept that extends adaptability and flexibility to include speed and scalability [62]. Adaptability is the opportunity for EIS adaptation to different specified environments without applying other actions or means than provided for this purpose. Flexibility is the ability of an EIS to respond to potential internal or external changes affecting its value delivery, in a timely and cost effective manner [165]. Scalability refers to the ease with which an EIS can be modified to fit the problem area. Anarchic scalability defines the need for architectural elements to continue operating when they are subjected to an anticipated load, or when given malformed or maliciously constructed data, since they may be communicating with elements outside enterprises control [128]. Business Agility is the term applied to an organization's ability to react to unanticipated market change, defining informed rapid transformation of organizational process and product [259]. Business Agility is the ability to sense change and opportunity in the marketplace, respond quickly, and execute successfully. Business agility is the sustained ability to sense and respond to change after change, executing well, occasionally inflicting change on competitors, yielding market leading returns [134]. Agile enterprise is the enterprise capable of responding quickly to changes [105].

Business agility is a promising strategy in times of change and uncertainty [62]. Enterprises must increasingly be organized, managed, and executed in ways that allow them to effectively sense and respond to unpredictable events in their environment. Enterprises must demonstrate agility in order to stay competitive [227]. Business change is inevitable, and whatever organizational framework is developed, it needs to be flexible to change [125, 175].

### 4.2.1 Business Agility Principles

Business agility principles can be extracted from different approaches of adaptive organizations [47]. Adaptive organizations are organizations that get affected by the social characteristics of its participants as well as the varied pressures imposed by its environment [204]. There are many approaches for adaptive organizations, like Haeckel's approach. Business agility principles include [47, 185]:

1. Agile enterprise entities must address four main agility characteristics:
  - Cooperate to enhance competitiveness.
  - Enrich the customer.
  - Master change and uncertainty.
  - Leverage the impact of people and information.
2. Traditional EISs that work well in stable predictable environment is replaced with adapted EIS with specific governance mechanism.
3. Translation of enterprise mission and practices into information that can be communicated and interpreted.

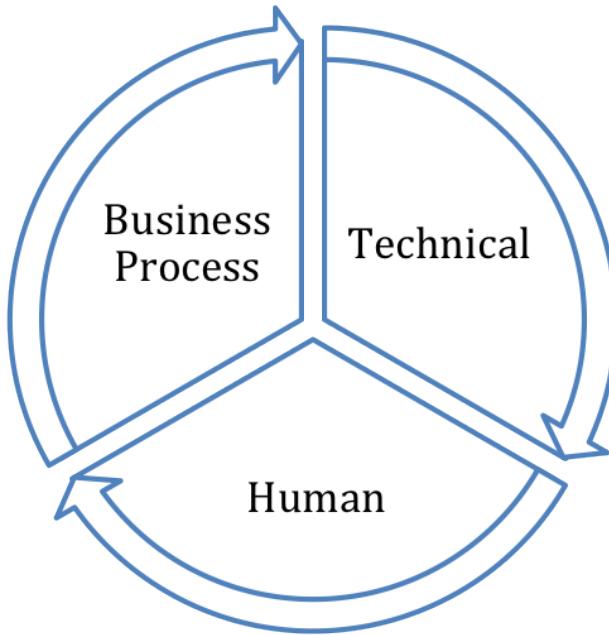


Figure 4.1: Business Agility Framework Components

4. Each business process should have dimensions that manage its relationship with other different enterprise entities.
5. Enterprise needs to implement business process lifecycle on different levels. Business process lifecycle is one of BPM promises about business process processing. Business process lifecycle phases include: model, design, simulate, execute, monitor, and optimize [243].
6. Enterprise needs to implement mechanisms for business process learning on different levels. Everyone in the enterprise should have access to the same business processes repository. The repository is used to support quality, definition of information requirements, and BPM modeling [92].

### 4.2.2 Business Agility Framework

Business agility means fast reaction to change and the ability to rapidly implement changes. Business agility needs to be holistic in scope [125]. Figure 4.1 presented in page 81 presents business agility framework core components. Business agility consists of three interoperable components: Human, Business Process, and Technical agility. Humans are assumed to be agile in management and operations for the enterprise to be agile. Human agility is the main enabler of business agility. Business process agility has gone long road till it reached BPM [243]. BPMS is the EIS that realizes BPM. Technical agility, that addresses IT infrastructure and ISs architecture, can make use of SOA. Aligning BPMS and SOA can enhance business agility.

## 4.3 Business Process Agility

BPM (BPM) is the key to business agility [243]. Business process is a series of inter-related activities that cross functional enterprise boundaries with individual inputs and outputs [129, 99]. Business processes are either operational or supporting [193]. Operational business processes are associated with the way enterprise develop strategies, invent, market and sell products or services. Support processes include the provision of Human Resource Management (HRM) activities, ISs infrastructure, and finance and asset management. BPM, as one of business improvement approaches, has a number of predecessors that include TQM, BPR, and WfMS. BPM predecessors have realized the need for process management [84].

### 4.3.1 TQM

Total Quality Management (TQM) wave as a business improvement approach has risen at the late 1980s [243]. TQM is a management approach for an organization, centered on quality, based on the participation of all its members and aiming at long-term success through customer satisfaction, and benefits to all members of the organization and to society [253]. Unfortunately, quality is neither easily defined nor measured. TQM does not have a firm measure by which progress can be monitored. Quality must be converted into something more tangible to be measured, but this does not take place with TQM. Further, TQM does not introduce a clear path to follow.

### 4.3.2 BPR

Business Process Reengineering (BPR) as a business improvement approach has emerged in the early 1990s [103]. BPR promised delivery of dramatic business improvements in a relatively short period of time by completely reinventing existing business processes that is, starting from scratch with new, optimized ways of doing business throwing out the old, encrusted, inefficient procedures of the past [103, 243]. Unfortunately BPR involved lot of manual work, failed to provide agility or support ongoing change, could not adequately represent the full complexity of processes due to dependence on computer systems, and provided process discovery and design only in team meetings.

### 4.3.3 WfMS and BPR

Workflow Management System (WfMS) and ERP promised Chief Executive Officer (CEO) the most flexible IT and providing enterprises requirements via software packages. Unfortunately, software packages were not flexible, installation could take months/years, subsequent applications couldn't be configured to meet all business requirements, business users were enforced to add components. Business process mapping Tools were added to ERPs, and WfMSs promising to produce documentation about the way company works, and to assist review and refinement of business process descriptions, but business process mapping tools could not carry business process models to execution, used proprietary formats and incompatible notations,

focused mainly on modeling input-output activities and dataflow, with no focusing on collaborative activity aspect or processes complexity.

#### 4.3.4 BPM

BPM is the art of understanding, codifying, automating, and improving the way a company does business [99]. BPM is a systematic, structured approach to analyze, improve, control, and manage processes [84]. BPM promises that Processes viewed by human as information and machines as executable code, Process design precedes both from top down - at the level of business strategy and business process design - and bottom up - at the level of leveraging existing IT systems, Process design is reflected directly in the IT infrastructure, leverage existing investments by connecting databases, legacy systems and package solutions into flexible end-to-end business processes. BPM introduces the concept of 'process processing' and stresses that this concept is not limited to the automatic execution of digital process models, but 'encompasses the discovery, design, and deployment of business processes, as well as the executive, administrative, and supervisory control over them to ensure that they remain compliant with business objectives' [103, 201].

#### 4.3.5 BPMS

BPMS enables enterprises to realize BPM initiatives [147]. BPMS is a revolutionary IS that supports designing, administrating, and improving the business processes systematically. BPMS is applied to create, execute, and optimize the business process model that powers the life cycle of each business process instance [127]. BPMS can be used to manage people-to-people, machine-to-machine, machine-to-people, and people-to-machine interactions [254].

##### BPMS Key Features

BPMS will bring all the benefits of integration, flexibility, end-to-end visibility and control to the whole extended enterprise. BPMS key features include:

- BPMS integrates and orchestrates other IT systems: BPMS will connect existing communication systems together, convey the required messages from one business entity to another, and ensure that the supply chain constituents stay aligned, with the system putting all of these interactions in the context of a whole process.
- BPMS connects and integrates existing databases, legacy systems, and package solutions into flexible end-to-end business processes.
- Allows processes to be shared across BPMS and across business boundaries: the use of a standard language for describing all aspects of a process will allow the same model to be deployed on several different systems.

## 4.4 Technical Agility

Technical agility refers to the ability to quickly change the type and flow of information within an organization within enterprise. Technical agility parameters are IT infrastructure, and IS architecture. IT advance has not yet satisfied business requirements due to improper ISs architectures. SOA addresses technical agility requirements by presenting composability, modularity, and loose coupling concepts as services that wrap underlying IT infrastructure, databases, and legacy systems and present them via standard interface. There is a need to stabilize IT infrastructure rather than developing new ones [62] and SOA enables this stabilization. Enterprises should balance IT to become better positioned and more agile [26]. Services are the building Blocks of an agile enterprise [217].

W3C defines Service as ‘A Component capable of performing a task’. Service is ‘A vehicle by which a consumer’s need or want is satisfied according to a negotiated contract (implied or explicit) which includes Service Agreement, Function Offered and so on’ [6]. SOA is the design pattern that utilizes services concept to achieve architectural advantages. W3C defines SOA as ‘A set of components which can be invoked, and whose interface descriptions can be published and discovered’. This definition can be expanded to include the science, art and practice of building applications [29]. SOA is defined as ‘The policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer. Services can be invoked, published and discovered, and are abstracted away from the implementation using a single, standards based form of interface’ [6].

IT Benefits of adopting SOA include EISs capabilities of [268, 120]:

- Avoid effect(s) of service provider change of implementation as a result of using standard interface.
- Choose alternative instance of same service type without modifying requesting IS.
- Implement different technologies than partners. Implementation of standard interface enables integration and interoperability.
- Increase services exposed in a loosely coupled manner, so ISs can easily combine existing services, and new ones based on business needs. This means less duplication of resources and more potential for reuse.
- Leverage existing legacy applications by wrapping them with standard interface.

### 4.4.1 SW Agents as SOA implementation

Different SOA implementations using different software agents and mobile agents were presented [227, 277, 224, 166, 70]. One or more agent can perform a certain task; tasks can be thought as services that compose SOA. Unfortunately, certain

limitations inhibited software agents from being the widely accepted SOA implementation. Software Agent is a computer system that is situated in some environment and is capable of autonomous actions in this environment in order to meet its design objectives [269, 178].

Software Agents have characteristics that make them suitable to perform complex functionality. Characteristics include: Autonomy, Interactivity, Reactivity, Proactivity, Intelligence, and Mobility [255]. Agent is autonomous; it is capable of acting on its own. An agent is goal oriented, collaborative, and flexible, so, it must be autonomous. Agents are designed to interact with other agents, humans, or software programs (Interactivity). Instead of making a single agent conduct several tasks, additional agents can be created to handle un-delegated task. Agents perceive environment via preceptors [234] and respond to changes (Reactivity). Agents do not just act in response to their environment, but agents are able to exhibit goal-directed behavior by taking an initiative (proactive). Agent may need mobility to work on different machines. An agent with this capability is called mobile agent, it can transport itself across different system architectures and platforms, and is far more flexible than those that cannot. Many electronic commerce agents are mobile [255, 46]. Mobile agent is an executing program that can migrate during execution from machine to machine in a heterogeneous network [178].

Multi-Agent Systems (MASs) are becoming increasingly important: as a scientific discipline, as a software engineering paradigm, and as a commercially viable and innovative technology [252]. A Multi Agent is any system that contains [102]:

- Two or more agents;
- At least one autonomous agent; and
- At least one relationship between two agents where one satisfies goal of the other.

Some of Multi-Agent frameworks proposed from 2005 till now include work presented in [206, 184, 208, 233, 271, 87]. Multi-Agent architecture standards were attempted in order to force MAS global integration [132]. Knowledge Query and Manipulation Language (KQML) was presented in order to support knowledge sharing among agents [130]. Knowledge Interchange Format (KIF) is a computer-oriented language for the interchange of knowledge among disparate programs developed by the Advanced Research Projects Agency (Advanced Research Projects Agency (ARPA)) sponsored Knowledge Sharing Effort [140]. The OMG group proposed a reference model as an attempt to standardize the development of agent technologies [258]. Knowledgeable Agent-oriented System (KAoS) is described as "an open distributed architecture for software agents." The KAoS architecture describes agent implementations (starting from the notion of a simple generic agent, to role-oriented agents such as mediators and matchmakers), and elaborates on the interactive dynamics of agent-to-agent messaging communication by using conversation policies [53]. The Foundation for Intelligent Physical Agents (FIPA) is a multi-disciplinary IEEE standardizing group pursuing the standardization of agent technology. FIPA's approach to MAS development is based on a "minimal framework for the management of

agents in an open environment.” [25]. Unfortunately as a result for all the standardization effort, there were no universally accepted commercially supported standard yet. Software agents are not the main SOA enabler because:

- Problems not easily defined into MAS organizations
- Absence of unified framework
- Security is a very big concern, specially with mobile agents
- Lack of commercial support
- Semantic web and Web services presentation (availability) as alternative technologies

#### 4.4.2 Webservices as SOA enabler

Web services are applications that use standard transports, encoding, and protocols to exchange information [239]. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. W3C defines Web service as ‘A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a format that machines can process (specifically WSDL), Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with XML serialization in conjunction with other Web-related standards’. Web service can also be defined as ‘A programmatic interface to a capability that is in conformance with Wsnn protocols’ [6]. Wsnn protocols are present efforts in the W3C and more recently in OASIS to reach a Web service maturity model. Wsnn protocols include WSDL, SOAP, and XML [56]. Web services is a general framework that expedites the sharing of heterogeneous data and software resources dispersed on the internet. The standard-based resource sharing and platform-neutral characteristics of web services have motivated many organizations to apply the technology in diverse areas, such as SCM, virtual enterprise, homeland defense, e-government, and e-business [276]. Within SOA context, Web services are categorized into utility and application services [122]. Utility services address and map business process workflow logic to application services. Web services are stateless, so business workflow logic should be maintained explicitly via utility services. Business process workflow logic can include Web services consumed, Web services will be consumed, and how to handle exceptions. Application services represent services that contain logic derived from a solution or technology platform. Application services present abstraction of system functionality. Web service that maps business process workflow is called process service, and it is a subcategory of utility service. Process services reside in the orchestration or choreography service layer that is a superior layer to the application services layer. Business process workflow logic is extracted, abstracted, and presented as process services. Business process workflow is easily maintained when presented explicitly as process service than embedded within individual solution components. Orchestration layer expresses a business process that is typically owned by the organization. Choreography layer addresses the realm of collaboration between multiple services from different enterprises.

## 4.5 Utilizing SOA for BPMS

BPM and BPMS are nothing new, but the use of technology to manage and improve the execution of business processes is more recent [99]. BPMS needs to address new enterprises requirements to support business agility. BPMS should enable enterprises to present and manage business processes as a new information type. BPMS is a necessity for agile enterprises [243].

Technically, BPMS is not Web services [127]. Utilizing SOA for BPMS is not about adopting Web services in BPMS. Though BPMS architecture doesn't have to be SOA based, the tremendous advantages of SOA direct BPMS architecture towards it. BPMS interoperability requirements is extremely different than other EISs, because BPMS does not just require interactions with other BPMSs; in order to share business processes; but BPMS is aimed to interact with different IT infrastructures and ISs exposed by different enterprises in order to support business processes. IEEE defines Interoperability as 'the ability of two or more systems or components to exchange information and to use the information that has been exchanged'. According to ISO/IEC 2382-01, interoperability is defined as "The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units". BPMS, either SOA based or not, lies above SOA exposed IT infrastructures and EISs that include databases, legacy systems, packaged solutions. BPMS tends to connect and integrate different EISs horizontally rather vertically. Horizontal integration is achieved via integrating business processes. In order to give enterprises flexibility to integrate business processes, their IT infrastructure and ISs should be exposed as loosely coupled stateless services that is accessible via standard interface. Business processes based integration presents maximum integration benefits, and is enabled mainly by SOA [113]. Figure 4.2 presented in page 88 presents the proposed model for aligning SOA and BPMS. Proposed model utilizes standards available for mapping BPM concepts via BPMS into SOA, and consists of three layers: Business, Business Services, and Application Services layer.

Business layer enables business executives to handle business processes as information, BPMS resides in this layer. Business services layer is the layer that maps BPM concepts and requirements addressed by BPMS into SOA based IT infrastructure and EISs. Application Services layer holds the core services ready to be consumed by different BPMSs and shared among enterprises.

### 4.5.1 Business Layer

Business layer is responsible for supporting business process life cycle. Business process lifecycle consists of five stages: design, model, simulate, monitor, manage, and optimize business processes. Business layer users are either business analysts, or business managers [264]. Business analysts create the initial drafts of the business processes, and business managers will manage and monitor those business processes. Business Process Modeling Notation (Business Process Modeling Notation (BPMN)) is the standard developed by Business Process Management Initiative (BPMI) to provide a notation that is readily understandable by all business layer users. Utiliz-

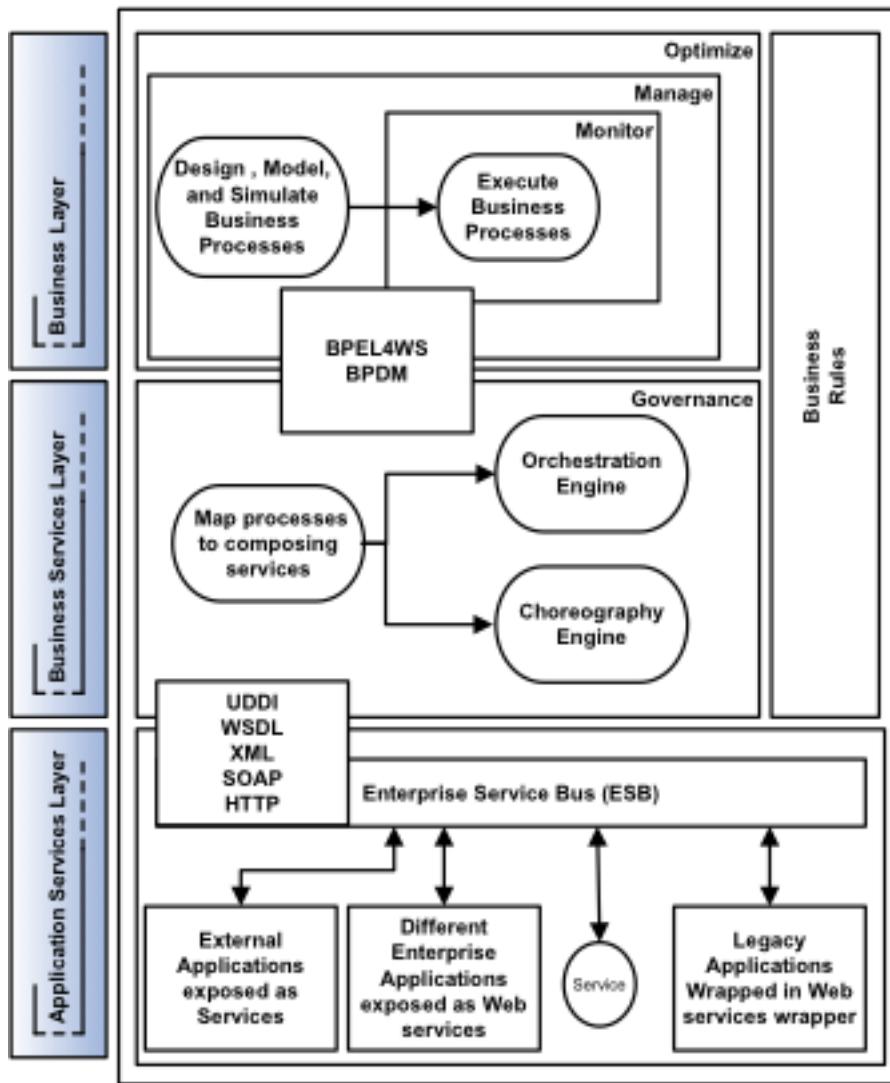


Figure 4.2: Proposed SOA and BPMS alignment model  
online: <http://zoom.it/hIWb>

ing UML to model business processes were presented in [191, 98, 171, 80]. BPMN provides a number of advantages to modeling business processes over the UML; it offers a more conductive process flow modeling technique, and a solid mathematical foundation to map to business execution languages, whereas UML is not. BPMN can map to UML [216]. BPMN has been intended as a human readable layer that hides the complexity of designing business processes in executable XML languages.

Business rules describe the enterprise's constraints, policies, and rules that apply to in achieving its goals. Business contracts are the constraints, policies, and rules orientation among enterprises. Agile enterprises need to present business rules repositories that can be managed, maintained, and checked against current or new executing business processes and / or services.

For business layer to achieve required functionality, a suitable IT architecture should be presented. IT architecture that utilizes Web services in order to obtain Web services advantages within organizations is presented at the Application Services layer. The connection between business layer and application layer is presented at Business Services Layer.

#### 4.5.2 Business Services Layer

Business services layer holds orchestration and choreography engines under governance mechanisms to map business processes to composing Web services. Orchestration and choreography engines are the mapping enablers of business processes into executable services. Web services are stateless services that can not maintain business logic, operation flow, or user state; so, the need of an orchestration layer to include business logic is addressed. Orchestration and choreography engines maintain business process workflow logic, performance requirements, and system/user state. Business services layer has access to business rules repository.

BPMN maps to Business Process Execution Language for Webservice (BPEL4WS) in order to translate designed business processes into executable orchestration and choreography Web services actions [216]. BPMN maps to Business Process Execution Language (BPEL) either directly or via XML Process Definition Language (XPDL). XPDL is a format standardized by the Workflow Management Coalition (Workflow Management Coalition (WfMC)) to interchange Business Process definitions among different BPMSSs. The goal of XPDL is to store and exchange the process diagram, to allow one tool to model a process diagram, and another to read the diagram and edit, another to "run" the process model on an XPDL-compliant BPM engine. XPDL is not an executable programming language, but specifically a process design format that literally represents the "drawing" of the process definition. XPDL is effectively the serialization of BPMN, as well as any non-BPMN design method or process model which use in their underlying definition the XPDL meta-model. XPDL maps to BPEL4WS.

BPEL4WS allows BPMS to transform business processes into executable complex processes by creating and wiring together different activities that can perform Web services invocations, manipulate data, throw faults, or terminate a process. These activities may be nested within structured activities that define how they may be run, such as in sequence, or in parallel, or depending on certain conditions

[66]. BPEL is an "execution language" designed to provide a definition of Web services orchestration, specifically the underlying sequence of interactions, and flow of data from point-to-point. For this reason, BPEL is best suited for straight-through processing or data-flow application integration. New BPEL extensions that include BPEL4People and BPEL Extension for Sub Processes (BPEL-SPE) are proposed.

A new specification that is expected to be adopted in 2007 is the Business Process Definition Metamodel (BPDM). BPDM is an XML-based proposal developed by the Object Management Group (OMG) that will define a set of abstract business process definition elements for specification of executable business processes that execute within an enterprise, and may collaborate between otherwise-independent business processes executing in different business units or enterprises." The specification developed in response to this RFP is expected to achieve, among other things:

- The ability to integrate process models for workflow management processes, automated business processes, and collaborations between business units.
- Support for the specification of Web services choreography, describing the collaboration between participating entities and the ability to reconcile the choreography with supporting internal business processes.

Governance refers to putting a consistent process in place to make sure there are checks and balances that ensure that the expected results happen. In the case of SOA, governance refers to keeping checks and balances between business and IT, between the business and government regulations, and between service and performance [165]. Governance rules are maintained explicitly in the proposed model to be checked by executing orchestration and governance engines to present flexibility for executing engines and ease manageability of applied rules. The separation of orchestration and choreography engines from business processes and composing services increased enterprise agility because the workflow logic encapsulated by an orchestration and choreography engines can be modified or extended in a central location [122].

### 4.5.3 Application Services Layer

Application services layer holds applications exposed as services, newly added services, and legacy applications wrapped by standard Web services interface. Services at Application Services layer are set of stateless Web services that perform certain task(s). Business process is the summation of tasks performed by one or more services of application services layer at the sequence maintained by orchestration and choreography engines. Services of Application Services layer are reusable among different business processes, can be integrated in new applications, and can be extended to address new business processes. Legacy systems are computer systems that have been in operation for a long time, and whose functions are too essential to be disrupted by upgrading or integration with another system despite its poor competitiveness. Legacy systems compatibility with modern equivalents has been facilitated via wrapper services. Wrapper service is a type of integration service that encapsulates and exposes logic residing within a legacy system via standard Web services

interface to be integrated in the new SOA based systems. Utilizing SOA for building new applications within enterprises exposes the ease of integration capabilities between new adapted/developed applications and existing applications. In SOA, messages are critical to delivering end-to-end services. Messages must be guaranteed a quick and correct delivery. To enhance messages transportation between services, SOA can use an enterprise service bus (ESB). ESB is the Infrastructure software that makes reusable business services widely available to users, applications, business processes, and other services [67]. ESB is a special layer that runs on top of the network that provides a guaranteed messaging service for the most important messages on the network, including the messages that the components of SOA continuously exchange. Adopting ESB in SOA solutions is not a must, but a recommended best practice.

## 4.6 SOA Selected Topics

### 4.6.1 Business Stackholders and SOA

Companies that need customizable solutions or use IT for competitive value, companies seeking to leverage IT capabilities for business advantage, these are companies that should care about SOA. Business stakeholders should care about SOA adoption if a business wants to be more responsive to their markets, increase market share, and improve customer loyalty, anything that represents a business outcome where IT can make a difference. Many of these benefits cannot be realized without a synergistic relationship between business and IT, which requires that stakeholders in business and IT understand enough about SOA to help make its promised benefits a reality. Although IT has a larger role in SOA adoption, active business participation will be necessary to achieve strategic goals and ensuing SOA benefits. Reasons include, not only:

- Pursuing business or IT transformation initiatives
- Seeking faster time to value from IT
- Transforming or modernizing strategic applications
- Attempting to lower the lifetime cost of applications or infrastructure
- Pursuing reuse as a goal to bring products or capabilities to the market faster
- Looking for greater flexibility in strategic applications
- Increasing revenue
- Reducing business process cycles
- Reducing time and costs for systems integration

#### 4.6.2 SOA ROI

Early SOA adopters have not focused on measuring Return On Investment (ROI) for SOA because bigger, more-profound forces are impacting decisions:

- how to change the ratio of IT spend so that more is spent on innovation rather than costs
- how to make strategic applications more flexible
- how to deliver faster value to the business
- how to be more responsive to their customers and markets
- and how to improve business performance.

Without SOA the long-term cost of the IT solution to the business will be higher. There are several reasons for non-SOA solutions having a higher cost, with the most prevalent being that non-SOAs don't have flexibility built in. SOA reduces the lifetime cost of an IT solution, and this is a key ROI metric. Several studies by vendors, analyst firms, and third parties consistently show that when asked, surveyed, or evaluated, 100% of clients who have successful SOA adoptions report and show improved flexibility, another key ROI metric.

#### 4.6.3 Measuring SOA Effectiveness

Measuring the effectiveness of SOA starts with identifying specific value propositions and assigning a suitable metric for assessing and tracking. When looking at effectiveness, the following value propositions are viable and potential metrics:

- **Improve IT's ability to respond to the businesses needs:** Metrics: Reduced calendar time to deploy new solutions, increase in types of changes business stakeholders (non-IT) can make, opportunity value related to faster time to market
- **Speed up delivery time:** Metrics: Reduced hours or % of development schedule, reduced test cycles or time
- **Improve flexibility of applications:** Metrics: % reduction in life cycle time from concept to production, elapsed days a functional type changes (e.g., regulatory rules) can be deployed into production, number of changes that business users can make or deploy, speed of moving through test environments
- **Improve flexibility of business processes:** Metrics: Number of processes that can be reconfigured, number of services being used, number of standard business processes.

Measuring the efficiency of SOA also begins with identifying value propositions and assigning metrics. Efficiency measurements, like effectiveness metrics, require organizations to baseline data so that improvements can be measured and tracked. Common efficiency value propositions include the following:

- **Reduce the lifetime cost of applications:** Metrics: Reduction in resources for maintaining code; reduction in cost for fixing code-related problems in production; reduction in number of defects, number of services being created, number of services being created with a single consumer
- **Reduce cost of integration:** Metrics: Number of application using shared SOA infrastructure, integration cost avoided or reduced by production-deployed infrastructure, reduced cost of building interfaces and infrastructure to support application integration
- **Improve productivity:** Metrics: Person days required to build a service, cost to build a service, project delivery times are shortening, number of projects being delivered, improvements in time to market for new capabilities
- **Reuse of SOA assets:** Metrics: Number of services with multiple consumers, number of services being consumed, service reuse ratios
- **Support long-term sunset strategy for applications:** Metrics: Number of common functions from multiple applications converted to services

#### 4.6.4 SOA Governance Effectiveness

For most organizations, when there is a failure to achieve any of the following, a red flag should be raised as to the effectiveness of SOA governance:

- Reduce the time to deploy business functions or changes to existing functions
- Reuse SOA assets by other projects or lines of business
- Improve flexibility of applications
- Utilize ESB to reduce costs
- Reduce maintenance costs
- Achieve development savings in new development of shared services

#### 4.6.5 Design and Run time SOA Governance

Design time governance includes the definition of policies and proper life cycle associated with a service as it is designed, tested, implemented, monitored, and registered in the service registry. Design-time governance provides a full life cycle view of a service from inception to deployment to retirement. Design-time governance uses registry and repository tools to track service design, management, policies, and any artifacts associated with the service. Such artifacts might include a test report demonstrating that a service successfully passed certain quality-assurance tests. Design-time governance includes design tools to facilitate the modeling and creation of services, deployment tools addressing service implementation, and test tools. Runtime governance uses the operational policies to monitor the runtime execution of the services against the policy criteria defined and against operational

requirements such as service level agreements. Runtime governance practices address managing the quality of a service such that a service is known in the context of its application flow or business transactions.

## 4.7 Summary

Business Agility is a relatively new paradigm painted as a solution for maintaining competitive advantage during times of uncertainty and turbulence in the business environment. Business agility consists of three interoperable components: Human, Business Process, and Technical agility. Human are the initiators and maintainers of enterprises agility. Business Process agility can be maintained by applying BPM concepts. BPM concepts are achieved via BPMS. Technical agility has gained new perspectives by adapting SOA for EISs. SOA based applications are being heralded as the only viable way to overcome the complexities involved in supporting agile enterprises. Business agility principles are satisfied via utilizing SOA for BPMS by providing the capability of:

- Translating enterprise mission and practices into information that can be communicated and interpreted. This translation is addressed by presenting the business rules as an external standalone repository that reflects enterprises constraints, policies, and practices. Business rules guides interactivity relations among enterprises.
- Replacing traditional EIS with adapted one with specific governance mechanism. Governance mechanisms are exploited explicitly in the presented over executable service via orchestration and governance engines.
- Providing dimensions for each business process that manage its relationship with other different enterprise entities. Business processes are presented in the model as new information type that has an external repository and management, and monitoring activities.
- Implementing business process lifecycle on different levels. Business process lifecycle is one of BPM promises about business process processing. Business process lifecycle phases include: model, design, simulate, execute, monitor, and optimization.

## 4.8 Review Questions

1. What is Business Agility?
2. What are Business Agility Principles?
3. Define each of the following:
  - (a) TQM
  - (b) BPR

- (c) WfMS
  - (d) BPM
  - (e) BPMS
4. How can we measure SOA effectiveness?
  5. How can we measure SOA governance effectiveness?

## 4.9 Exercises and Labs

In Labs, you shall be getting familiar with Java programming language, as it is the programming language we will be using to build, deploy, and consume different online Webservices and SOA systems presented in this book.



# Chapter 5

## Integrating UMIS and LMS

e-Learning is the "learning" process revolution enabled by new technologies that, hopefully, will present an effective and efficient learning process that doesn't exist today. LMSs are responsible for "learning" activities, while university management information systems (UMISs) are responsible for handling University managerial activities. Sociotechnical systems recognized many years ago that organizations functioned most effectively when their social and technological networks were compatible [265]. This is the case exactly with e-Learning systems. LMSs can't provide the managerial functions needed to support universities, and UMISs don't support the "learning" process. Both systems have to integrate and operate together to support educational institutions and e-Learning.

In my attempt to clarify this confusion, I thought about surveying a prototypical e-Learning model and presenting one of the UMIS models, casting them side-by-side to help clarify the confusion by highlighting their differences. My goal is to make this piece of research available to students and e-Learning researchers, so we can together overcome this confusion and start focusing on the "learning" process as the main asset of "e-Learning."

### 5.1 Introduction

When you type "e-Learning" into a search engine, you find the word has been widely used to refer to computer-based systems that do not necessarily maintain the objective of "learning."

Researchers believe that one of the goals of introducing e-Learning was to revolutionize the learning process, either by facilitating many of the challenges that face instructors and learners daily, or by presenting opportunities that might have not existed before.

Evaluating e-Learning is a wide and tough research field that needs more work. Some e-Learning researchers believe that utilizing technology within the learning process has not to date achieved many objectives because requirements were not addressed correctly and clearly. Too often, the learning process and its main activities were neglected for the sake of introducing and focusing only on the technological aspects. The same group believes that one road to a solution is to start by addressing effective future e-Learning requirements, current e-Learning shortcomings, and

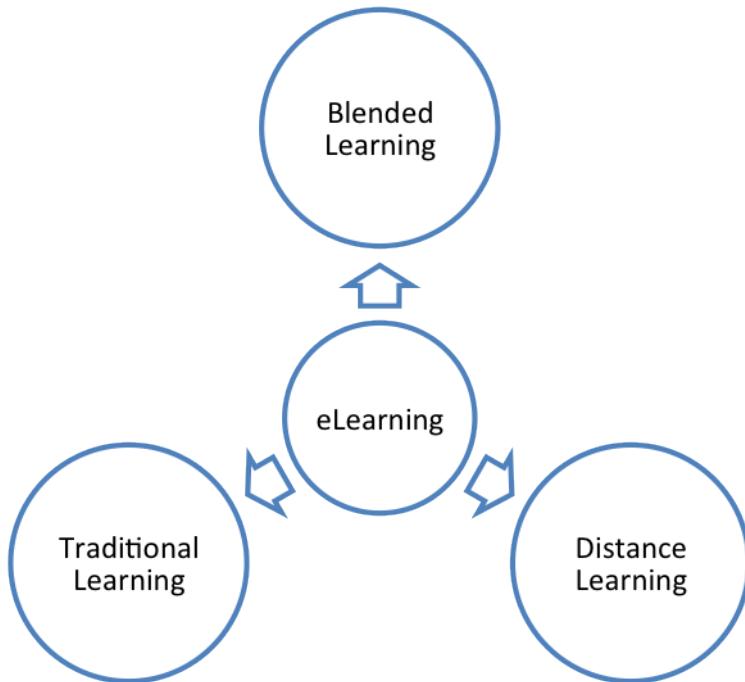


Figure 5.1: Learning Models

today's technological limitations. Others, such as the University of Phoenix believe that online learning has achieved quite a bit of success and diffusion.

## 5.2 Learning Models

e-Learning can be thought of as the learning process created by interaction with digitally delivered content, services, and support [83, 115, 228]. It involves intensive use of information and communication technology to serve, facilitate, and revolutionize the learning process [108, 160, 197, 207, 256].

Figure 5.1 presented in page 98 shows three main learning models that we have come to recognize over the years [108, 160, 207, 256].

1. **Traditional learning:** In traditional learning, students head to a school, college, or other physical space in which to learn. Information and communication technology can enhance the learning process, but is not necessarily included. For example, data show and presentations can be thought of as an implementation of e-Learning within a traditional learning institution [26].
2. **Distance learning** In distance learning, an instructor and students are separated by time, location, or both. Education or training courses are delivered to remote locations via synchronous or asynchronous means of instruction [213]. Distance education does not preclude the use of the traditional classroom [174].
3. **Blended learning** Blended learning combines multiple models to learning. For example, students in a traditional class can be assigned both print-based and online materials [256].

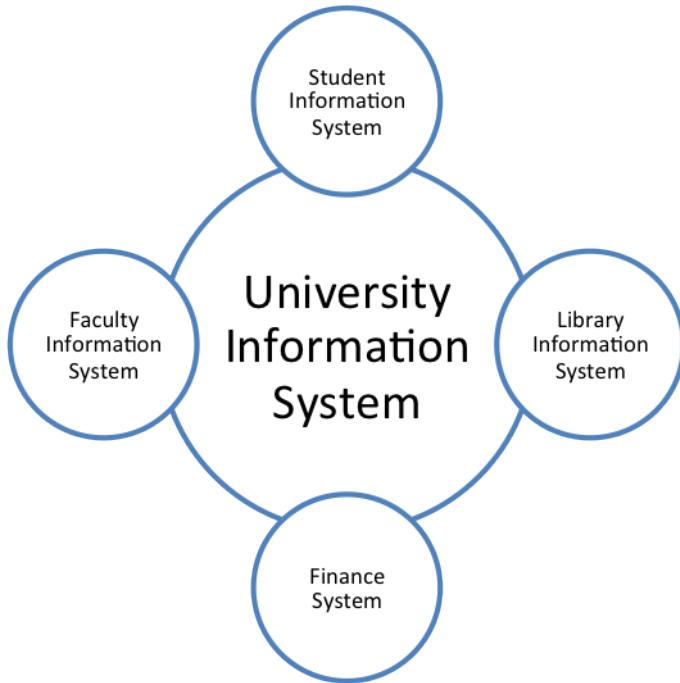


Figure 5.2: Prototypical University Management Information System

## 5.3 UMIS

e-Learning tends to revolutionize and manage the learning process [68], not only to manage universities. Ignoring the learning process while designing and developing e-Learning systems leads to Management Information System (MIS), which are important in managing educational institutions activities and help educational institutions achieve a mature level of automation, but are not themselves learning-focused systems.

Managing universities activities requires a university management information system. UMIS refers broadly to a computer-based system "collection of hardware, software, people, data, and information" that provides managers with the tools for organizing, evaluating, and efficiently running their departments [194, 262].

Examples of UMIS components include a student information system (SIS), a library information system, a faculty information system, and a finance system, as illustrated in Figure 5.2 presented in page 99. Below, I present each component in brief detail, listing the reasons they can't be considered part of an e-Learning system.

### 5.3.1 Student IS

Student Information System (SIS) is the IS responsible for managing students' data at the university. A typical student record in the SIS might includes the student's ID, social security number, name, age, gender, address, email, username, password, date of birth, faculty, university status, and department [230].

SIS by itself is not an e-Learning system because personal data that it provides

and manages differs in nature than data required for education [72]. For example, the SIS should not be able to provide things like detailed records of what the student has already learned (at the level of learning object, rather than a module or program), or a profile of his or her learning preferences, or a development portfolio of transferable skills. On the other hand, a learning portfolio might actually include a history of the student's interaction with tutors, peers, and other significant learning conversations.

This kind of data is intended to be used to force the learning process to be a learner-oriented by adapting the system to fit the learner's needs, characteristics, and capabilities. Unfortunately, SIS does not serve this purpose, and does not handle such data <sup>1</sup>.

### 5.3.2 Library IS

Library information systems are responsible for managing and automating libraries within faculties and/or universities. Automated libraries contain material in digitized form [211]. The DB records in these libraries reflect the managerial tasks performed by librarians in order to effectively manage the libraries. A typical record will include ISBN, name, authors, keywords, and data like section, a list of all books, a list of available books, a list of borrowed books, who is borrowing, when the books are due to return, and so forth.

Automated library information systems do not in and of themselves serve the learning process and thus are not on their own considering e-Learning systems. Learners should be able to access fully available digital libraries as part of the learning process.

### 5.3.3 Faculty IS

Faculty information systems manage and automate managerial activities related to instructors, employees, courses, and the intersections between them. A typical faculty information system DB record includes 1) faculty data: ID, name, departments, courses data; 2) course information: course id, name, description, instructors; and 3) faculty personal data: ID, social security number, name, age, gender, address, email, username, password, date of birth, year, department; and 4) employee data, which is the same as the instructor's data but with customized data about one's job [230].

A faculty information system's main goal is to organize faculty and university managerial activities; the learning process is not part of the main objective and therefore it is not, on its own, considered e-Learning. The system's capabilities are to generate courses reports, for example, that includes course managerial issues.

### 5.3.4 Finance System

A finance system manages financial issues related to any organization, even if this organization is a faculty or university. However, I believe financial issues of the

---

<sup>1</sup>The informative model is the model that forces one way of information transformation from instructor to learner, no matter what the personal differences are, whereas the learner-oriented model modifies the system to fit the individual based on personal data.

educational institution doesn't have anything to do with e-Learning at all. Though e-Learning systems might encompass some financial issues of selling courses, that doesn't entitle the e-Learning system to include a complete set of a university's financial system.

### 5.3.5 UMIS Role

UMIS achieved success over the years and proved efficiency and effectiveness within educational institutions. UMIS is required for any successful e-Learning implementation in the three learning models, but with constraints about the role it should play. UMIS manages educational institutions, and more attention should be paid to the learning process with the presence of UMIS [30].

## 5.4 Prototypical e-Learning

Researchers have tried to define a prototypical e-Learning system over the years, resulting in different points of views, not to mention a variety of terms acronyms. Each term reflects its presenter's point of view and what system features must be present. Some of these terms are:

- adaptive teaching system [219]
- assessment management system [42]
- authoring system [66]
- CAI: computer assisted instruction [148, 153]
- CAPA: computer assisted personalized approach [214]
- CBT: computer-based training
- CIT: computer-information-television [118]
- collaborative learning [145]
- computer managed learning system [73]
- computer assisted learning [73]
- CMS [196]
- courseware authoring tool [60]
- distance education [81, 149]
- electronic courses [164]
- enterprise CMS [170]
- online courses [164]

- integrated student information system [273]
- LCMS: learning content management systems [43]
- LMS: LMS
- interactive learning environment [126]
- integrated learning system [78]
- PLE: personal learning environment
- telecast [27]
- virtual college [164]
- virtual conference [55]
- virtual classroom [26]
- virtual university [35]
- VLE: virtual learning environment [72]
- WBT: web based training [43]
- web-based interactive course [158]

By studying all these terms, it's clear that there is some confusion regarding which tools, features, and concepts each includes in its definition. In attempt to organize those acronyms, extract features, and tools, clarify concepts—and keeping in mind that the learning process is the main concern—Figure 5.3 presented in page 103 shows one way to categorize them together. e-Learning implements technology that enables virtual/digital university, and/or personal learning environments. A virtual or digital university is a university that implements online LMSs or Virtual Learning Environment (VLE)s and provides tools for virtual college. e-Learning is the main concept that includes enabler technologies implementation for both virtual/digital university and personal learning environments (PLE). [LMS and VLE reflect the same implementation of the same concept. LMS is widely used in the U.S. and was presented in 1980, while VLE is used in U.K. and was presented in 1983 [81].]

PLE represents a new trend in e-Learning that claims student's right to use only one gateway to be able to access different LMSs provided by different universities. Those different LMSs should be personalized and integrated within this gateway and be able to interchange educational student data and information, so provide students with portability between different systems.

Universities and colleges are digitized by implementing ICT. The maximum extent of digital university is the virtual university, where the whole learning process is managed and maintained digitally. LMS, VLE, and extended LMS are the main implementation of e-Learning today.

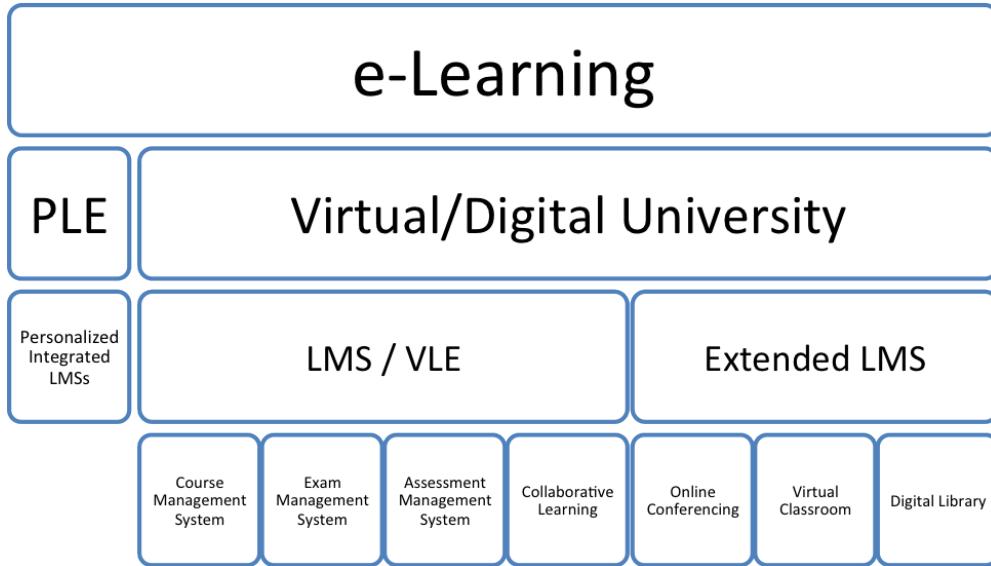


Figure 5.3: As an umbrella term, "e-Learning" covers many features.

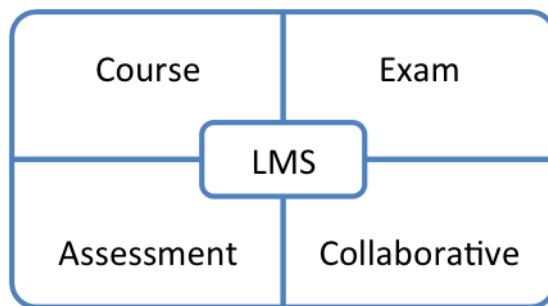


Figure 5.4: The Main LMS Functionalities

## 5.5 LMS and VLE

LMS and VLE are more or less interchangeable, and LMS is used to refer to both. LMS is the software that automates the administration of training. The LMS registers users, tracks courses in a catalog, records data from learners, and provides reports to management.

An LMS is typically designed to handle courses by multiple publishers and providers. It usually doesn't include its own authoring capabilities. Instead, it focuses on managing courses created by a variety of other sources [83, 115]. Britain and Liber give an example of atypical LMS [72].

LMS features can be categorized into four main separate systems as depicted in Figure 5.4 presented in page 103. Those four systems are concerned with courses, exams, assessments, and collaborative features. LMS can be thought of as the integration of four separate systems, each presenting specific functionalities via specific tools. Figure 5.5 presented in page 104 depicts the most common features that should be available in each of those four separate systems.

Course Management System (CMS)	Exam Management System	Assessment Management System	Collaborative Learning
<ul style="list-style-type: none"> <li>• Course Authoring</li> <li>• Course Publishing</li> <li>• Manage Content Repositories</li> <li>• Track Student Progress</li> <li>• Online Courses</li> <li>• Search Tools</li> <li>• Course Management Features:</li> <li>• Add new Course to Course list</li> <li>• Remove course from course list</li> <li>• Define Course outline</li> <li>• Define Course homepage</li> <li>• Define Course Syllabus</li> <li>• Determine Course Materials</li> <li>• Define Course Outline</li> </ul>	<ul style="list-style-type: none"> <li>• Grade books</li> <li>• Track Exams</li> <li>• Exams Repository</li> <li>• Answer Sheet Facility</li> <li>• Spell Check</li> <li>• Text to Speech</li> <li>• Generate Powerful exams</li> <li>• Different types of Questions</li> <li>• Automated MCQ</li> <li>• Multiple Right Choices</li> <li>• True/False</li> <li>• Short Answer</li> <li>• Interactive</li> <li>• Oral</li> <li>• Laboratory based exams</li> <li>• Essay based</li> </ul>	<ul style="list-style-type: none"> <li>• Homework Submission</li> <li>• Track Assessment Submissions</li> <li>• Assessments Answers Analysis</li> <li>• Answer Sheet Facility</li> <li>• Spell Check</li> <li>• Text to speech</li> <li>• Generate Different types of Assessments</li> <li>• Automated Assessments</li> <li>• MCQ</li> <li>• Multiple Right Choices</li> <li>• True / False</li> <li>• Short Answer</li> <li>• Complex Assessments</li> <li>• Collaborative Assessments</li> </ul>	<ul style="list-style-type: none"> <li>• Async discussion</li> <li>• Discussion Forum</li> <li>• Newsgroup</li> <li>• E-mail System</li> <li>• Sync discussion</li> <li>• Instant Messengers</li> <li>• Virtual Meetings</li> <li>• Simulations</li> <li>• Online Laboratory</li> <li>• Students Area</li> <li>• Students Homepages</li> <li>• Blogs</li> <li>• Bookmarking</li> <li>• Recommended List</li> <li>• Class List</li> <li>• Notice Board</li> <li>• Calender</li> <li>• Search Tools</li> </ul>

Figure 5.5: LMSs offer a long list of component functionality.

### 5.5.1 Extended LMS

Extended LMSs include added functionality that doesn't necessarily provide e-Learning per se, such as communication tools and digital libraries. Communication tools, such as support for online conferencing and chat, enable or enhances communication among instructors and students and support the existence of a virtual community.

#### Online conferencing

Providing conference capabilities over the internet is known as online conferencing, and it can be take three different forms:

1. Data conferencing [117], or sharing data interactively among several users in different locations. Data conferencing is made up of virtual whiteboards and application-sharing software, which are often used in conjunction with an audio or video connection. A whiteboard is the electronic equivalent of a chalkboard or flip chart, and participants at different locations can simultaneously write and draw on an on-screen notepad visible to all the participants. Application-sharing software is the same as remote control software, in which multiple participants can interactively work in an application that is loaded on only one user's machine. Application "viewing" is similar to application "sharing" except that only one person can manipulate what's visible.
2. Audio conferencing usually takes place when there are three or more people who are geographically dispersed. It is provided by a conference function in a multi-line telephone, or by the telephone companies in some countries, or via VOIP [116].
3. Video conferencing is a real-time video session between two or more users or between two or more locations. Videoconferencing may comprise any number of end points [118].

#### Virtual Classroom

A virtual classroom is the online learning space where students and instructors interact [174]. Virtual classrooms provide unique online features [23] such as:

- chat
- discussion: chat between more than two participants, which can be made public or private
- question and answer: individual participants may ask questions, and instructors may provide public or private answers
- whiteboard
- group browser: displaying the same screen to a geographically dispersed group
- break-out sessions, which allow a subset of learners to use a private chat area within the virtual classroom.

## Digital Library

Digital libraries are libraries that contain electronic materials [211]. Digital libraries' implementations might include digital data from academic institutions, public libraries, government agencies, and museums [212].

Digital libraries play an important role in the learning process due to the tremendous amount of data available to any of the LMS components anytime, and anywhere. My faculty is working currently on a project to digitize graduation projects or final works as a main source of information to fresh students and as a mean to communicate back and forth between instructors and students. This digital library will be available online so readers can review books authored by faculty instructors (if they approve, of course), and review graduation projects ideas, concepts, and documentation. Hopefully this project will enhance students' graduation projects by giving them a digital library of their colleges' work, and enhancing knowledge-sharing and collaboration among faculty.

## 5.6 UMIS OR LMS

Universities require both UMIS and LMS for efficiency and effectiveness. Neither UMIS nor LMS can replace the other. University managerial requirements are addressed by UMIS, and learning process requirements are addressed by LMS. Universities need both!

Open-source LMSs are ones for which the source code is available. Free LMSs can be downloaded and offer unlimited use, but do not typically provide source code. Commercial LMSs are another option and typically come with tech support. By surveying LMSs, it is clear to find that most LMSs implement same features as depicted in prototypical LMSs as the one depicted in "Supporting Education and Research—A higher education committee"(from Joint Information Systems Committee). Table 5.6 lists the most important Open Source, Commercial, and Free LMSs.

One of the international initiatives that provides information to learning institutions on investing in and using information technology infrastructure is e-Framework for Education and Research. Its primary goal is to facilitate technical interoperability within and across education and research communities through improved strategic planning and implementation.

e-Framework's techniques are not intended to be applied immediately; rather, the initiative promotes a journey that takes steps and stages to achieve interoperability goals.

Socio-technical systems recognized many years ago that organizations functioned most effectively when their social and technological networks were compatible. LMSs are responsible for learning activities while UMISs are responsible for handling University managerial activities.

But both UMISs and LMSs have to integrate and operate together to support educational institutions and e-Learning.

Table 5.1: Examples of Currently Available LMSs

Open-Source	Commercial	Free
.LRN	BlackBoard	KnowEdge eLearning Suite
COSE	Desire2Learn	
LON-CAPA	IntaLearn	
moodle	WebMentor	
ATutor	Janison Toolbox	
Claroline	Unicon Academus	
Eledge	BSCW	
KEWL	Colloquia	
ILIAS	eCollege AU+	
MimerDesk	Internet Campus Solution	
SAKAI	IBM Lotus	
OLAT	Centra	
	The Learning Manager	
	Angel	

## 5.7 IBM Lotus

IBM Lotus is one of the leading LMSs that implements too many complicated features at a high level [81,82]. IBM Lotus is one of LMS dominants [83]. Figure 5.7 presented in page 108 presents a detailed Lotus architecture and list of functions that are performed by Lotus components. IBM Lotus is AICC certified since 1997 [82]. By studying Lotus preview [84] and architecture [85], it becomes clear that it is not applicable to satisfy all the educational institutions requirements within one LMS no matter what efforts are attempted by companies. Concerning Lotus as one of LMS leaders, neither the presence of 11 servers nor letting down some of the functionalities is acceptable. Issues like scalability, interoperability, and integration have been mainly serious issues for current available commercial LMSs that forced many educational institutions towards in-house LMS development and implementation to satisfy educational institutions special requirements.

## 5.8 Evaluation of Current LMSs

Research is an academic activity that comprises defining and refining the problems, formulating hypothesis or suggested solutions, collecting, organizing, and evaluating data, making deductions and reaching conclusions, and at last carefully testing the conclusions to determine whether they fit the formulating hypothesis [86]. Evaluation is a main step of scientific research that enables in concluding and reporting research results, efficiency, effectiveness, and goals achievement. Evaluation utilizes many of the same methodologies used in traditional scientific and social research, but because evaluation takes place within a political and organizational context, it

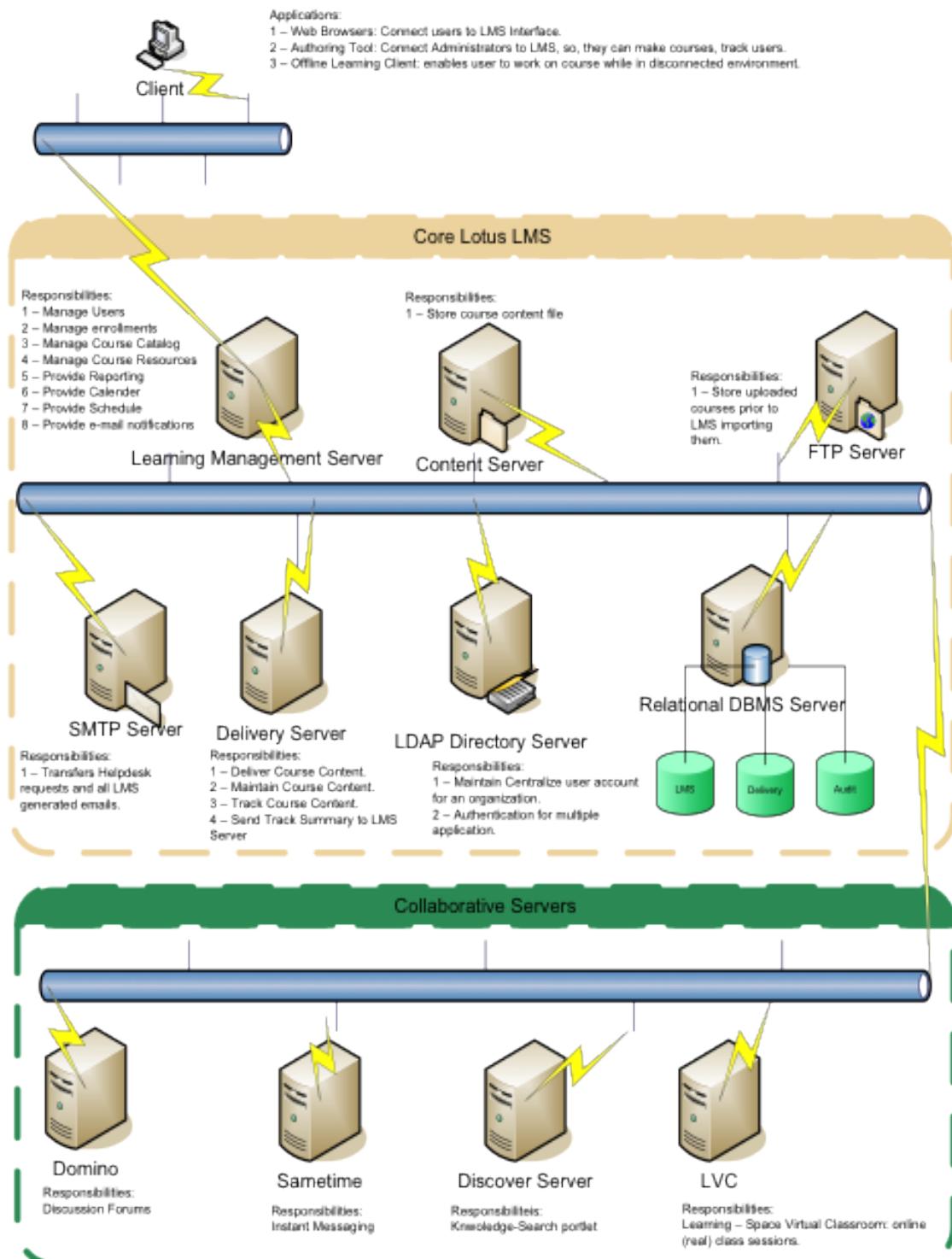


Figure 5.6: Detailed IBM Lotus Architecture  
online: <http://zoom.it/TYkM>

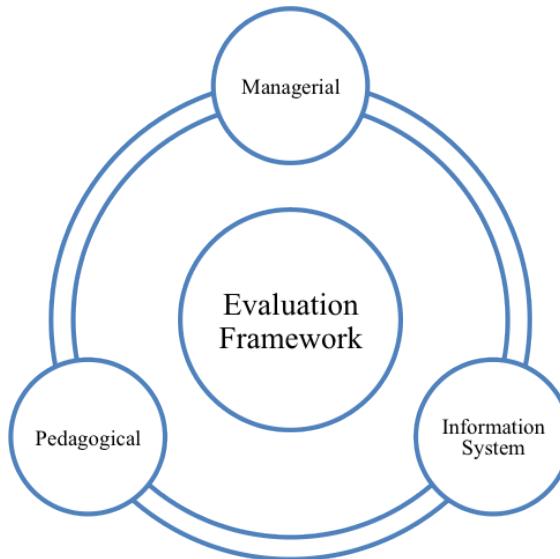


Figure 5.7: LMS Evaluation Framework

requires group skills, management ability, political dexterity, sensitivity to multiple stakeholders and other skills that scientific and social research in general does not rely on as much [87]. Evaluation-is the collection and analysis of information by various methodological strategies to determine the relevance, progress, efficiency, effectiveness, and impact of programs activities [88, 89, 90]. Evaluating LMS is not like evaluating any software system, because LMS should address pedagogical aspects, beside architectural and managerial aspects as depicted in figure 5.7 presented in page 109.

### 5.8.1 Managerial Evaluation

Total Cost of Ownership (TCO), among other methodologies like Cost-Benefit Analysis (CBA), is widely used as an analytical and justification tool for software assessment, replacement, and acquisition projects. Unfortunately, TCO analysis is time-consuming to complete based on assumptions, and sometimes hard to quantify. TCO evaluation of a Financial Information Systems is available in [91]. Same TCO managerial evaluation model can be applied from the managerial perspective to LMS. Managerial LMS Evaluation is out of scope.

### 5.8.2 Pedagogical Evaluation

Pedagogically, LMS shall enable universities and educational institutions to provide educational services in an easy, effective, and efficient manner. LMS providers and evaluators must be aware of pedagogical effects that will affect instructors and students. Current LMSs do not provide the required pedagogical effects [73]. One of the reasons is technology limitations. If technologies applied in LMS will not enhance learning process then pedagogically it is not a necessity and unfortunately



Figure 5.8: Non-ISO Addressed IS Quality Parameters

this is the case today.

### 5.8.3 IS Evaluation

One of the most important models to be used in evaluating ISs is the one presented by the standard ISO-IEC 9126. Figure 5.8 presented in page 110 presents other architectural parameters that can be used in evaluating IS. Presented parameters has not been addressed by ISO-IEC 9126 Standard, but still can be used by ISs evaluators. It is evaluators' responsibilities to determine the most valuable architectural aspects to be considered in the evaluation process. Another architectural evaluation model that deserves mention and well consideration is the one presented in [92].

The study of most well known LMSs; including Lotus, has lead to the realization of the reality that there are many shortages of current LMSs affecting e-Learning in educational institutions. This part tends to present some of the shortages addressed through evaluations of current LMSs. Some of the addressed deficiencies are: Integration, Agility, Scalability, Extensibility, Flexibility, Interoperability, and Redundancy.

#### Integration

e-Learning solutions are clearly a combination of two main categories of applications: LMS and UMIS. As presented in figure 5.9 presented in page 111, there are four integration challenges:

- 1. Category components integration:** Composing applications might come from different software vendors, legacy applications that have been running for

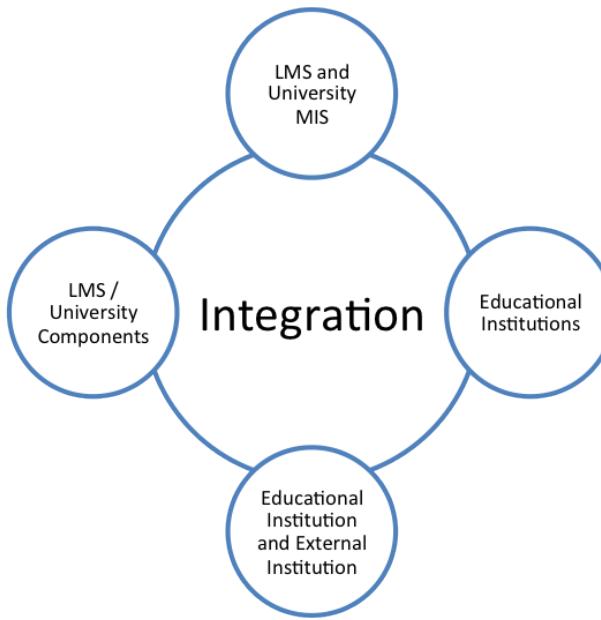


Figure 5.9: University Integration Challenges

a long time needs to be integrated with new applications, and new software applications might be installed. Existing applications might stand in the way of installing new applications due to integration problems.

2. **UMIS and LMS integration:** UMIS advances LMSs by decades [26]. UMIS achieved stability levels that most major educational institutions are running the same UMIS for almost a decade with no noticeable problems. LMS is the new concept that needs to be added to educational institutions, and it has to integrate with long time running UMIS. Due to differences in nature of design objectives between UMIS and LMS, integration issue should be addressed clearly. There will be new features that university will be able to provide incase full integration of UMIS and LMS took place. The capability of presenting individualized assessments that can be generated by integrating SIS; which will include two profiles for student; a managerial profile, and an educational profile; and the assessment management system should be addressed. The capability of presenting individualized learning plans by integrating CMS and SIS that holds the learning profile should be addressed. Personalized courses will be generated based on learner history profile. It is not acceptable by learning experts to fix time and change amount to be learned, instead, amount to be learned should be fixed and personalized. Such individualizations enable PLE.
3. **Integrate different LMSs:** It is hard to find an educational institution that implements only one LMS because educational institutions requirements are hardly met by one LMS [73]. Different LMSs, either within the same educational institutions or over different ones needs to be integrated. Integration over courses level as provided by utilizing SCORM and AICC is not the typi-

cal solution of integration problems, because educational institutions need to share more than courses.

4. **Integrate educational institution and external institutions:** Educational institutions are not isolated islands that work alone, but they are part of the society. Educational institutions need to be integrated with all other governmental institutions. Tasks like generating a report with all current certain grade students, or students that are qualified for certain task because they have attended certain courses should be easily executed. Querying the educational institutions databases should be available efficiently. Student's data shall be available to give the student ability to transfer his data between educational institutions as required.

## Agility

From certain perspective, educational institutions are organizations, just like any other organizations. Organizations need agility, and so do educational intuitions. Agility is the ability to sense change and opportunity, respond quickly, and execute successfully. Educational institutions need to reflect governmental new or modified rules on current UMIS, overcome merge and acquisitions challenges, and provide new and customized services based on new added system features. An example of new customized services that should be presented when new system features are added is collaborative assignment. Collaborative assignment can be provided by the system after installing new collaborative tools. If the system were not built with agility in mind, new features adding or existing features editing becomes almost impossible, limiting the system from adapting and presenting new functionalities.

## Scalability

System should expand and contract its resource pool easily to accommodate to heavier loads. Client – Server based LMSs can not scale to support large number of components, or interactions among components, within an active configuration [92]. Most today's LMSs are Client – Server based [219]. Scalability needs to be addressed to offer educational institutions with the LMS solutions that satisfy the increasing requirements.

## Extensibility

Extensibility is the ability to add functionality to a system. Dynamic extensibility implies that functionality can be added to a deployed system without impacting the rest of the system [92]. That implies not only adding new features to existing LMS without the need to install the new version, but also adding customized functionalities required by the educational institution. Unfortunately, current LMSs are released periodically to provide new functionality, with the need to install the new version every time. Besides, most commercial LMS are categorized according to functions available by each product level, without guarantee on what are exactly the actions required to be taken to make the move to the next level incase required.

### Flexibility

Flexibility is the ability of a system to respond to potential internal or external changes affecting its value delivery, in a timely and cost effective manner. The ability to change or to be changed according to circumstances is not by default provided by Client – Server architectural based LMSs, and unfortunately most LMSs are Client – Server based [219]. Changes include addition/loss of servers, and types of services provided by the system.

### Interoperability

Interoperability is the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units [93]. Interoperability is concerned with processes require the system to communicate with external systems to accomplish tasks. Educational institutions need to interact with external systems to achieve tasks successfully. Interoperability between LMS and external systems needs to be addressed and enabled. Current commercial LMS architectures and used protocols are not provided with LMS. A standard interface to LMS functionality is required.

### Redundancy

Redundancy refers to the unmanaged, uncontrolled, and unwanted more than single time occurrences of either data or functionality.

**Data Redundancy** refers to existence of the same piece of data in more than one place. Data is said to be in consistent state when all data redundancy are controlled. Data redundancy should be controlled for insert, update, and delete queries. Figure 5.10 presented in page 114 depicts an example of uncontrolled data redundancy that causes inconsistent data and system state. SIS, CMS, assessment, exam, and finance information systems need student Data, unfortunately with different formats and different types. Questions like which system should manage this data? Shall every system manage its own data? Shall the system take the risk of having a redundant data? Need to be answered.

**Functionality Redundancy** : Functionality overlapping and redundancy is clear from the literature review. Functionality redundancy comes from the attempt of LMS software vendors to increase the tasks performed by LMS to include functions that are not part of it. Redundancy is accepted only if it is managed.

## 5.9 Selected Topics

### 5.9.1 SOA and Service Modeling

Because SOA projects vary in size and scope, not all SOA projects require service modeling. SOA projects whose goals include engineering business applications that

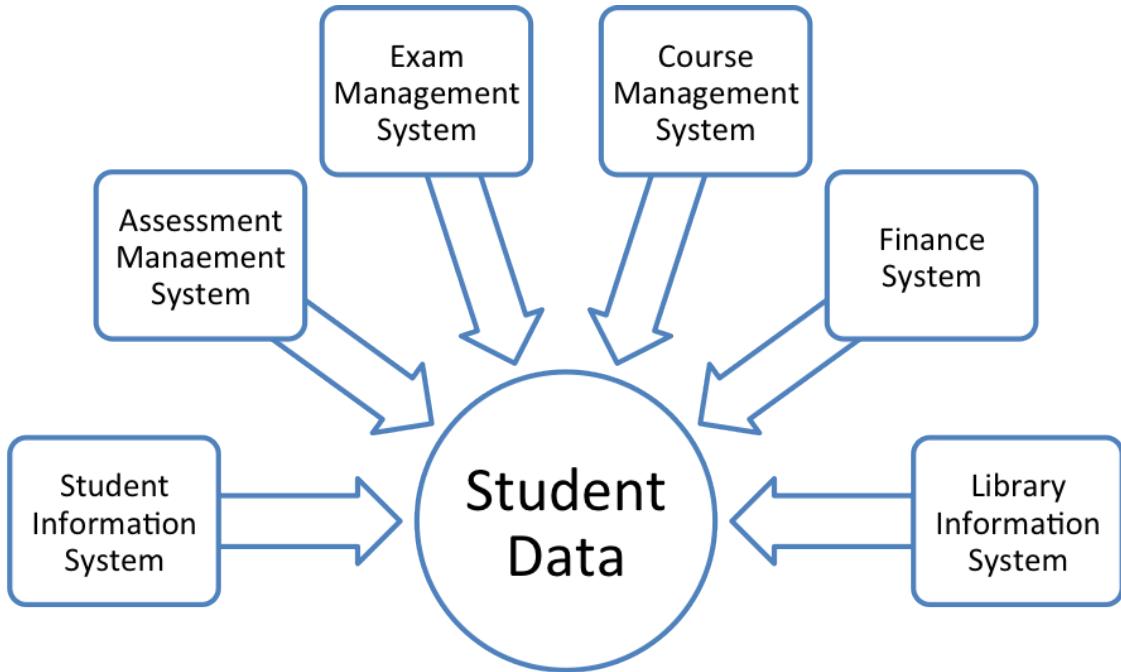


Figure 5.10: Redundant Student Data

are built to change, leveraging and reusing business services, reducing the lifetime costs of an application, or using SOA to accelerate time to value require the identification, specification, and realization of services. These SOA projects require service modeling. Service modeling produces a categorized list of services, business functionality, and capabilities required by the business. This list can be described as a service portfolio.

### 5.9.2 Applications and Composite Services

Composite services are not applications but services that invoke other services. The invocation is independent and without knowledge by the consumer that other services are being invoked.

### 5.9.3 BPM and Applications

BPM is a management discipline focused on the following:

- Aligning business process performance and the results with strategic objectives and business goals
- Understanding and documenting business processes so that they may be consistently executed
- Measuring, monitoring, and controlling process performance, including key inputs and outputs

- Actively designing and improving business processes to meet or exceed the expectations of customers while achieving business goals (e.g., cost and revenue)

BPM solutions vary from a focus on human-centric business processes, to document and content-focus processes, to structure and system-oriented processes. A business process can be represented as collaboration between components.

## 5.10 Summary

This chapter presented the ISs utilized by educational institutions. They are UMIS and LMS. UMIS serve educational institutions to automate managerial tasks, and LMS provides the required functionalities to automate the learning process. Educational institutions need both UMIS and LMS to satisfy e-Learning and enhance the learning process. Both UMIS and LMS are composed of collection of task specific applications. Educational institutions; and as a result the learning process, can be enhanced by enhancing the IS that serves them. Shortages of current LMS are evaluated in order to find solutions that serve educational institutions to better address information, learning, and functionality requirements. SOA (SOA) is one of the software architectures that satisfy most of the non-functional requirements addressed by ISs. SOA principles need to be presented and studied in order to utilize SOA for UMIS and LMS to overcome current LMS shortages, limitations, and deficiencies. SOA as a design pattern is expected to solve many of the current IS problems by easing integration, interoperability, and agility. The idea of this thesis is to adopt SOA in e-Learning via presenting UMIS and LMS based on SOA, and evaluate the proposed components to determine the efficiency of SOA. Evaluating LMS is not limited by evaluating the architectural style of the system, it shall be expanded to include pedagogical and managerial aspects too.

## 5.11 Review Questions

1. What are the main components of UMIS?
2. What is the difference between LMS and VLE?
3. What is Extended LMS?
4. What are the perspectives of evaluating LMS?

## 5.12 Exercises and Labs

In Labs, you shall be getting familiar with Java programming language, as it is the programming language we will be using to build, deploy, and consume different online Webservices and SOA systems presented in this book.



# Chapter 6

## LMS from Monolithic to Services

### 6.1 Introduction

e-Learning can be thought of as the learning process created by interaction with digitally delivered content, services, and support. e-Learning involves intensive usage of Information and Communication Technology (ICT) to serve, facilitate, and revolutionize learning process [197, 160, 207]. Learning methods include traditional learning “face-to-face”, distance learning “complete asynchronous time and place learning delivery; mainly online”, and blended learning “learning that combines instruction lead learning with online learning activities leading to reduced classroom contact hours”. Blended learning has the potential to increase student learning while lowering attend rates compared to equivalent fully online courses [109]. Blended learning is the learning paradigm that attempts to optimize both traditional learning and distance learning advantages, potentials, and benefits while eliminating both learning paradigms shortages and challenges. Nowadays blended learning is commonly referred to as e-Learning. When compared to traditional learning paradigm, blended learning is found to be consistent with the values of traditional learning paradigm adopted in almost all higher education learning institutions for decades, and has the proven potential to enhance both the effectiveness and efficiency of meaningful learning experiences [137].

LMS (LMS) is the software that automates the administration of education. LMS registers students, tracks courses in a catalogue, records data from learners, and provides reports to management. LMS is typically designed to handle courses by multiple publishers and providers. It usually doesn’t include its own authoring capabilities; instead it focuses on managing courses created by a variety of other learning resources. Sandy Britain and Oleg Liber present a prototypical LMS in [72]. Because learning models can be classified into traditional, distance, and blended learning paradigms; universities and educational institutions can also be classified into three classes that map to each learning paradigm. However, nowadays technology and the great advancement in recent Web 2.0 and informal learning methods allowed the existence of completed education programs and courses to be presented online. So, university model can be either Online University “full distance learning model”, Campus University “the one that supports either the traditional or blended learning paradigm”. Campus Universities used to have managerial aspects that can’t

be neglected and need to be addressed to enable universities to achieve their goals effectively. University Management Information Systems (UMIS) is the software IS responsible for managing universities. UMIS advances LMS with decades. UMIS has achieved mature level of well formed business and performance requirements adoption, and usage. Unlike LMS; which is the newly introduced IS to universities when compared to UMIS. UMIS and LMS needs to be integrated to enable universities present effective and efficient learning.

Architecture is about providing balance in the face of conflicting concerns. Software entities, architectures, have gotten more complex as we have progressed from monolithic to client/server to network-centric architectures and now to SOA. Architectural evolution continues to move toward agility and hence the attention and interest in SOA. At the heart of every new shift in architecture sits a practically or theoretically compelling concept that brings the paradigm forward. Client/server introduced the distributed application architecture that portioned workloads between service providers and service consumers referred to as clients. Network-centric architectures introduced the Internet as a paradigm shift. Now with SOA, services are the architectural game changer, moving the architecture to greater agility.

## 6.2 Problem Definition

Current Blended Learning Model Paradigm faces many challenges. Those challenges are mainly pedagogical. No one can deny that today's technologies advances by decades technologies used in teaching and education less than decade ago. However, a number of recent articles have commented that science education is no better today than it was fifty years ago. National Assessment of Education Process (NAEP) shows that in most areas today's students are achieving at about the same levels as students tested in 1971 [69]. e-Learning researchers know very well that this problem is not caused by technology; because as mentioned before technology has advanced greatly. This pedagogical issue is the result of:

- The attempt to use whatever technology currently available or becomes available in the near future without pedagogically considering student or the learning process.
- Allowing technology to stand against the learning process, because no matter what advancement we have achieved, technology is still limiting our dreams (because we still dream about things that we can't yet achieve technologically), and this situation takes place with e-Learning nowadays.
- The poor evaluation that is available for many of the innovations. Most of the required evaluations are either inadequate or doesn't exist at all.

Alfred Bork et al. argue that some of the reasons why technology has not led to improvements in learning globally are [69]:

- Grabbing Onto Each New Technology: The belief that each new technology will enhance learning needs more arguments about efficiency than just belief.

- Failure to Continue Successful Development: funders often prefer to look for something new rather than follow up on successful approaches because they want to make a mark by being in the forefront. Funders want to make a statement, and following up on someone else's work doesn't provide them with the credit or "name" they desire.

Pedagogically, most training methods and technologies produce, at best, "trained novices". That is, they introduce facts and concepts to students, present them with relatively simple questions to test this new knowledge, and provide them with a few opportunities to practice using this knowledge in exercise or scenarios. However, becoming proficient requires extensive proactive solving realistically complex problems in wide range situations, combined with coaching and feedback from managers, more experienced peers, or other types of experts [215]. Most evaluations of today's presented technologies in the learning model focus on technology aspects of the solution while ignoring the pedagogical aspect; almost at all. The result of using technology, particularly computers, in learning has so far not been impressive. A variety of studies and opinions have questioned the use of technology to improve learning. Although it has been many years since computers have begun to be used in learning environments, there is little improvement in learning, with or without technology. Although the use of technology in learning shows no significant difference, that is, computer learning is no better than traditional instruction, learners have been provided with the convenience of any time, any place learning. Students' understand and retention improves when students learn by experience. Technologies such as collaboration, interactivity, modelling, simulations, virtual reality interfaces, and gaming will help students experience the skill being taught, but they have not helped students that far yet [69]. Besides, students lack of awareness of different e-Learning technologies stand up against the presentation of effective e-Learning model.

We started coding in objects after procedural programming and structured design had evolved. Object-oriented programming gave way to the need to design using objects, and then a plethora of object-oriented analysis and design methods and techniques ensued: Rumbaugh, Jacobson, Coad, Wirfs-Brock, and others. Eventually, patterns evolved that took smaller units of programming and built micro architectures that were dubbed design patterns. One of the programming notions that developed into a best practice in object orientation was "program to interfaces, not to implementations." This separation of concerns allowed greater underlying flexibility in development: both at design time (classification and class associations) and runtime (polymorphism). With the evolution of objects came the need to group them into larger-grained entities and to build larger-grained components. Component-based development, and later component-based design, created larger structures that were more closely mapped to business intent and needs. When Web services—with the promise of looser coupling and greater standardization than the Common Object Request Broker Architecture (CORBA) or DCE came to the forefront and promised to make distributed computing more accessible, programmers started to build remote procedure calls with XML-based objects. As the momentum grew, Web services architecture gave way to SOA.

## 6.3 Proposed Solution

Adaptive learning for students with many different backgrounds, learning styles, and interest is almost a must. Educational psychologists by and large agree that students differ greatly in the ways they learn and very few teachers or professors can adapt learning to each student in the typical large classes, the costs associated with delivering different instruction for varied learning styles is prohibitive [69]. Benjamin Bloom (1984) showed twenty-five years ago, as reported in his 2 sigma paper, that almost all students can learn to the mastery level, given the right learning environment [209]. In Bloom's experiments, the most successful learning strategy was tutoring. Adaptive e-Learning that is supported with intelligent techniques and methods is one of the ways to support personalized e-Learning, and so authors believe that it will be the way to solve many of the limitations and today's e-Learning challenges. Authors will review the different adaptive and intelligent e-Learning systems proposed over a long e-Learning research era, highlighting the maturity of adaptive and intelligent features and the e-Learning researchers attempt to introduce services based e-Learning systems in order to overcome many of the e-Learning challenges, like reusability, scalability, integration, and interoperability. Finally, authors will conclude with the need to present a new learning model that attempts to use the best of what was presented before, and avoid all the challenges and mistakes. Though authors believe there's no single unified learning model can be the only right model, hopefully this work will be a step towards a better learning model supported with the appropriate technologies.

## 6.4 Adaptive e-Learning Systems

Computer-based learning systems are criticized by many researchers for their limited ranges and adaptability of teaching actions compared to rich tactics and strategies employed by human expert teachers [205]. Many universities in the developing countries, started to adopt e-Learning by modifying their network infrastructure, establishing new labs, providing internet connection, and purchasing different tools for creating e-Learning courses and using different LMSs. However, these modifications and supplement were not enough to ensure successful e-Learning outcomes because other important elements for e-Learning success were missing such as flexibility of the system, adaptability towards students needs, reusability of learning objects, interoperability between different LMSs, effective and official design of e-content [200]. Adaptive e-Learning systems would be a good solution for better e-Learning. The absolute majority of Web-enhanced courses rely on LMS because they are powerful integrated systems that support a number of teachers and students needs. Though LMSs look surprising, indeed for every function that a typical LMS perform there is an Adaptive Web Based Educational System (AWBES) that can do it much better [75]. Adaptivity is the ability to modify e-Learning lessons using different parameters and a set of predefined rules. Researchers differentiate slightly between adaptivity and adaptability by thinking about adaptability as the possibility for learners to personalize an e-Learning lesson by themselves. These two approaches go from machine centred (adaptivity) to learner centred (adaptability).

In practice, it is quiet difficult to isolate one from the other due to their close relationship [77, 248]. Adaptive e-Learning is often meant to be new or in an early development stage [205]. Christian Gutzl defines adaptive e-Learning systems as an environment of software modules, which comprises a set of features for adaptivity and adaptability [146]. Important factors for adapting to student needs and desires include [69]:

- Each Student Should Move at a Unique Pace: Given all the variations between student backgrounds, interests, and abilities, it is highly desirable to allow each student move at a unique pace in the learning units.
- Adaptation Should be Very Frequent: Changes based on occasional exams are inadequate. Learning activities should adapt to each student on a moment-by-moment basis. Students should feel that the adaptive program is responding to them as individuals.
- Each Student Should Be Successful in Learning: a major advantage of adaptive variable placing is that the students can continue to learn in a given area until they have learned the material. We know from Bloom's research that almost all learners can succeed and achieve mastery, but some learners need more time and more practice than others.
- When Something Is Successfully Learned, the Learner Should Move On: Often in classroom learning, after a student has learned something, the class continues working on the topic, boring the student. This will not occur in a fully adaptive learning environment.
- No One Should Be Taught Something S/he Already knows: By assuring learner competencies, avoiding unneeded instruction, and moving each student forward when ready, we expect to achieve a major reduction of learning time, but this cannot be verified empirically until we have a full range of computer-based adaptive learning units.

The provision of static learning material will not meet the requirements of the users. Adaptive e-Learning enables personalizing learning process to individual learners via adapting some parameters; like identifying, analyzing and monitoring relevant aspects of instructions, such as different velocities, paths, or strategies of learning can be personalized. Performance improvements within the learning process can be gained via adaptive e-Learning systems [146]. Adaptation and personalization will improve the learning process, therefore, a paradigm shift from the consumption of static learning contents to well tailor and highly personalized learning sessions is needed.

#### 6.4.1 Adaptive e-Learning Approaches

Four main approaches which are used to give a historical overview of adaptive e-Learning can be identified [205]:

- Macro-Adaptive Approach: Addresses adaptation of instructions on a macro-level by allowing different alternatives in selecting a few main components such as learning objectives, levels of detail, delivery system, etc. In this approach, instructional alternatives are selected mostly on basis of the student's learning goals, general abilities, and achievement levels in the curriculum structure.
- Aptitude-Treatment Interaction (ATI) Approach: This approach treats adaptation of instructional strategies to specific student's characteristics. This strategy proposes different types of instructions or even different media types for different students. The most important classes of learner characteristics can be summarized with the following ones: intellectual abilities, cognitive styles, learning styles, prior knowledge, anxiety, achievement motivation, and self-efficacy. One aspect of the ATI approach is the user's control over the learning process according to the abilities of the students by giving them full or partial control over the style of the instruction or the way through the course. Level of control can be one of three levels: complete independence, partial control within a given task scenario, and fixed tasks with control of pace.
- Micro-Adaptive Approach: Addresses adaptation of instructions on a micro-level by diagnosing the student's specific learning needs during instruction and providing instructional prescriptions for these needs. Researchers have attempted to establish micro-adaptive instructional models using on-task rather than pre-task measures. Monitoring the user's behaviour and performance, such as response errors, response latencies, emotional states, etc. can be used for optimizing instructional treatments and sequences on a very refined scale [146].
- Constructivistic-Collaborative Approach: Follows the constructivist pedagogical approach. An important element of this approach is the usage of collaborative technologies which are considered often on essential component of e-Learning. Adaptive system enables learning by focusing on how knowledge is learned and should consider the context, learning activities, cognitive structures of the content, and the time extension. Some new adaptive e-Learning systems take account of students' motivational factors combining the instructional plan with a "motivational" plan.

Over the last decades, various types of adaptation systems and possible areas for their applicability have been identified, thus leading to the emergence of specialized research fields, like Adaptive Educational Hypermedia Systems (AEHS), Computer Aided Instruction (CAI), Computer Managed Instruction (CMI), Recommender Systems, Intelligent Tutoring System (ITS), Personalized Systems of Instruction (PSI), and many others. Adaptive multimedia systems as an improved learning environment is well documented in the research work of Christian Gutl et al. [146].

#### 6.4.2 AEHS Overview

Brusilovsky thinks about Adaptive Hypermedia Systems as systems that refer to all hypertext and hypermedia systems which reflect some features of the user in the

user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system; it should have a user model; it should be able to adapt the hypermedia using this model [248].

### **Application Areas of AEHS**

Educational Hypermedia Systems are the most popular area for adaptive hypermedia research, hence come the term AEHS (AEHS) and incorporates applications like:

- Adaptive Textbooks: Adaptive Textbooks use history-based, knowledge-based and prerequisite-based adaptive annotation of links to suggest to the individual user an appropriate path through a learning space by adapting the navigation links [111]. Adaptive textbooks can help students learn faster and better [75].
- Adaptive Quizzes: Quizzes and Tests are assembled by the system from a pool of questions using student and domain models that are manipulated with automated reasoning machines. Such forms of evaluation are adaptive in a per-question basis. Also it would be desirable to have the questions themselves be adaptive [124].
- Adaptive Class Monitoring: Class Monitoring tools are important to give the teacher much better chances to notice the students lagging behind [75]. Adaptive Teaching Systems can help instructors adapt lectures and educational contents to fit students.
- Adaptive Collaboration Support: Intended to capture adaptive support in learning processes that involve communication between multiple persons (and, therefore, social interaction), and potentially, collaboration towards common objectives. Adaptive techniques can be used in this direction to facilitate the communication/collaboration process, ensure a good match between collaborators, and other activities [218]. Adaptive Collaboration Support can enforce the power of collaborative learning [75].

### **Modeling the User**

Three categories of user related data can be defined and modeled in AEHS [248, 179]:

- User Data: Refers to information about personal/characteristics of the user. User features that can be stored in AEHS include knowledge, goals/tasks, background, hyperspace experience, preferences, interests and individual traits, and cognitive and learning styles.
- Usage Data: is related to information on the user's interactive behaviour that cannot be resolved to user characteristics. This category include data like: selective actions of the user (clicking on links, scrolling and enlarging operations for hypermedia objects, audio control operations), temporal viewing behaviour rating (users are required to explicitly rate objects, links, WebPages), and purchases and purchase-related actions.

- Environment Data: comprises aspects of the user environment that are not related to the users themselves. This kind of adaptation has evolved due to Web-based systems. Adaptation decisions may depend on spatial-temporal location of the user, the user platform (such as hardware, software, equipment - computer, PDA, cell phone, digital TV), and network.

## Adaptation Taxonomy

In 1996 Brusilovsky developed a taxonomy providing a way to segment adaptive research into two categories, which was labelled “adaptive presentation and adaptive navigation support” [74]. Figure 6.1 presented in page 125 shows an updated taxonomy of adaptive hypermedia technologies [76]. This taxonomy has been illustrated and studied by many researchers [248, 58]. Whereas the adaptive presentation changes the content presented in the documents, the adaptive navigation support changes the structure of links between the documents and how this navigation structure is presented to the user.

### Adaptive Presentation

The goal of adaptive presentation is to adapt the content of a hypermedia page to the user’s goals, knowledge and other information stored in the user model. This area concerns the selection and composition of information fragments to be later presented to the user. For example, the system may decide to provide additional explanations as well as to hide content from a novice user, while an expert user in the domain will receive more specific and more detailed information. In a hypermedia system the content of a “page” may not only be text, but it may also contain various items of other media types. The main benefit of the adaptive presentation is that it tries to reduce the amount of presented information to the most relevant information for a particular user, solving the information overload problem of the classic hypermedia systems.

- Adaptive Multimedia Presentation: the selection of different multimedia fragments (in practice this technique also includes the adaptation of multimedia content, just as the generation of shortened versions, but currently most examples of adaptive multimedia simply select between prefabricated content).
- Adaptation of Modality: choice among different types of media to present to the user, related to the same content; e.g., for a video we can have full video, still image, text description, or use of them in parallel.
- Adaptive Text Presentation
  - Natural Language Adaptation: Natural language adaptation cannot be classified further at the moment. Many natural language generation systems do, in fact, make use of fragments (and even paragraphs) of canned text, but distinction here is made between those systems that use natural language technology as a foundation and those that do not. Refinement of this distinction might be needed at a later stage.

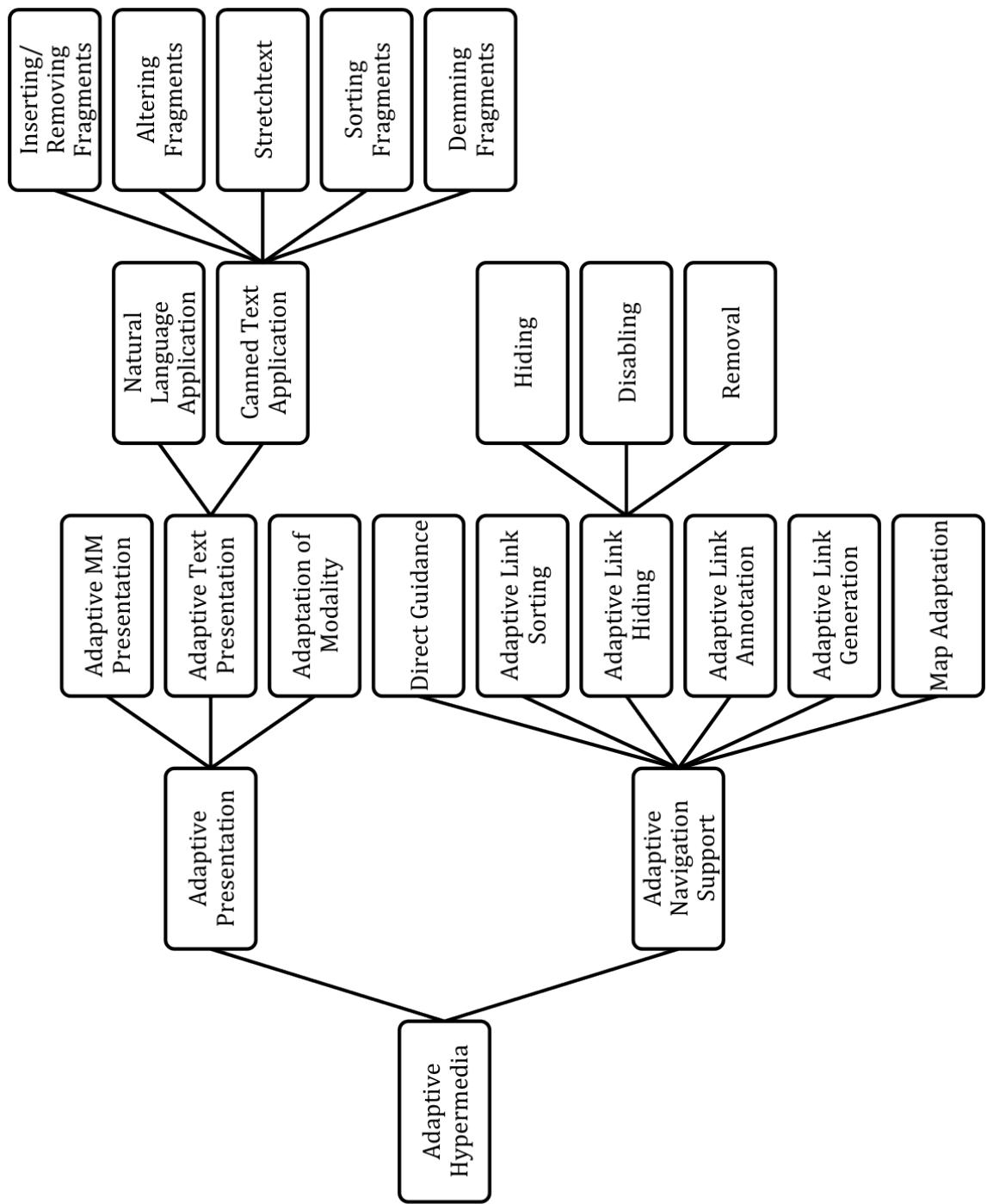


Figure 6.1: Updated Taxonomy of Adaptive Hypermedia Technologies

- Canned Text Adaptation: This term includes fragment processing, such as:
  - \* Inserting / Removing Fragments: Information related to a concept is broken into several fragments of text (or multimedia content). With each fragment a condition is associated on elements of the user model. When information related to a concept is presented, the system selects only those fragments for which the condition is true. This technique can be used to implement the methods of additional, prerequisite and comparative explanations.
  - \* Altering Fragments: Can be used for the implementation of the explanation variants method. The AEHS stores several variants of the same information fragment, and selects the variant to display based on the user model. For example, each variant is created for different group of users such as beginner, intermediate, and expert.
  - \* Stretch Text: Fragments are embedded in a webpage and initially shown or hidden from the user depending on conditions on user model data. AEHS can show or hide these fragments also upon explicit user requests. This technique is useful for implementing additional, prerequisite or comparative explanations, but less for explanation variants and not all for sorting.
  - \* Sorting fragments: This technique is used for the sorting method and presents a set of fragments to the user, ordered from the most relevant to the least relevant according to some criteria based on user's goal, background, knowledge, etc.
  - \* Dimming fragments: Used to dim, shade or deemphasize (in some way) a fragment to indicate that it is not (or at least less) relevant for the user.

## Adaptive Navigation Support

Adaptive Navigation Support modifies or augments the existing set of hyperlinks shown to the user to aid them in finding relevant information with the goal to help users find their way through hyperspace by adapting link presentation and functionality to the goals, knowledge, and other characteristics of an individual user. The goal is to guide the user towards relevant and interesting information and to advise the user not to follow navigation paths that lead to irrelevant information. The main benefit of the adaptive navigation support is that it simplifies the rich link structure and solves the “lost in hyperspace” problem while maintaining the navigation freedom that is typical of hypermedia systems. Adaptive navigation support contains several different techniques that can be used individually or combined to provide navigational support:

- Direct guidance: AEHS outlines visually one of the links on the page showing that this is the best link to follow or generates an additional dynamic link which is connected to the recommended “next” page to visit.

- Adaptive Link Sorting: AEHS uses sorting mechanisms; like ordering links from most relevant to lowest relevant to recommend links to users.
- Adaptive Link Hiding/Disabling/Removing: AEHS tries to prevent the user from following links that are not relevant at the moment. There are several ways to achieve this goal. A link can be hidden by changing the color of the anchors to that of normal text or removing the anchor. Also, link functionality can be disabled and so the link is removed.
- Adaptive Link Annotation: AEHS augments the links with some form of comment, which can tell the user more about relevance of the nodes behind the annotated links. These annotations are usually provided in the form of visual cues such as icons, font colors, sizes, and types.
- Adaptive Link Generation: AEHS may discover new useful links between pages and add them by using previous navigation (by an individual user or a user group) or page similarity to add links.
- Map Adaptation: This technique consists of a combination of the other techniques, the only difference being that it is applied to a graphical visualization of the navigation structure. The map is usually presented in a separate frame or window.

### AIEHS Overview

Enhancing AEHS with methods and techniques from ITS creates the Adaptive Intelligent Educational Hypermedia Systems (AIEHS). ITS form an advanced generation of Computer-Aided Instruction (CAI) systems with their ability to provide a user-adapted presentation of the teaching material. ITSs have been developed and evaluated for many years in the field of artificial intelligence in education. The emergence of WWW increased the usefulness of such systems. Web-based learning systems should be more intelligent through adoption of artificial intelligence techniques [197]. Intelligent e-Learning systems include utilizing artificial intelligence techniques, such as decision making, machine learning, planning, scheduling, and cognitive science in e-Learning platforms [203]. The objective of intelligent e-Learning systems, as is typically conceived, is to provide highly structured lessons that are to a large extent under automated control. Within this framework, the intelligence of the system often appears in the form of adaptive, sequencing or personalization of the course material, adaptive guidance for navigation, or interactive problem solving support. The most interesting opportunities to develop intelligent functionality are related to facilitating collaboration rather than adapting the learning material [152]. There has never been a clear distinction between intelligent and adaptive e-Learning systems. AI techniques have been widely used to support adaptive e-Learning features. Christopher Paul Bailey presents an agent-based framework to support adaptive hypermedia [59].

### 6.4.3 Adaptive e-Learning Applications' Examples

N. Matar et al. presented what they called “adaptive e-Learning system” to address different adaptive features of e-Learning [200]. N. Matar et al. argued that the proposed system can help universities; especially newly established ones come over limitations and shortages by presenting a global course repository. Despite the obstacles that can come against such a solution; starting from the required infrastructure and ending with the instructors’ choice to refuse submitting the learning material they have created, centralization as a solution is not the most suitable solution anymore. A. M. Riad and Haitham A. El-Ghareeb argues that shareability among different LMSs is the solution to unlock course repositories and presented a SOA based CMS that integrates different LMSs [113]. Besides, N. Matar et al. talked about the ‘Unified Adaptive e-Learning System’ as the proper solution to adaptive e-Learning implementation ignoring the fact that adaptive e-Learning solutions doesn’t come in the ‘one solution fits all scenarios’. Adaptive e-Learning solutions differ from different cultures, and different learning scenarios. Instead of attempting to unify adaptive e-Learning, integrating different adaptive e-Learning modules together to establish an adaptive upon request e-Learning solution is the best solution.

## 6.5 Intelligent e-Learning Systems

Artificial Intelligence (AI) utilizes programming algorithms to simulate thought processes and reasoning that produce behaviour similar to humans. A successful implementation of AI could be tested using a Turing Test approach, in which a human interacts with an interface that could have either a human or computer on the other end. The test is considered successful if the human is unable to determine whether there is a computer or a human on the other end. The applications of AI within e-Learning can produce the potential of creating realistic environments with which students can interact. The student essentially would interact with the IAs which in turn perceive changes in the simulated environment. The IAs would then communicate perceived changes in the environment back to the student who then makes decisions based upon their own perceptions of the environment [17]. Current learning technologies can help create trained novices more efficiently, but they are really not up to the job of creating true experts. For example, multimedia Computer Based Training (CBT) systems are good at presenting information and then testing factual recall using multiple choice or fill-in-the blank questions. However, traditional CBT systems are incapable of providing intelligent, individualized coaching, performance assessment, and feedback students need to acquire broad and deep expertise [215]. Employing the state-of-the-art artificial intelligence (AI) technology in current e-Learning systems can bring personalized, adaptive, and intelligent services to both students and educators. Although we have seen more and more successful applications of AI in e-Learning, most of them have not yet been expanded to or adopted in widely used e-Learning systems, especially open-source LMSs (LMS) such as Modular Object Oriented Dynamic Learning Environment (moodle), Sakai and so on. This observation takes us to the analysis and discussion of the current work in both

LMS and applied AI. The findings include that current intelligent LMS systems are still in their early stage, while AI applications need to handle some problems or to be modified before applying them into the LMS systems, and AI technology also needs to be brought to open source communities [263].

### 6.5.1 Evolution of AI in e-Learning

Koschman views the evolution of Computer Based Instruction (CBI) as a series of paradigmatic shifts in the field of Instructional Technology. According to Koschman, the initial CBI paradigm, entitled Computer-Assisted Instruction (CAI) emerged in 1960 with the release of Coursewriter I [249]. Koschman argues that the next paradigm shift ITS was founded by members of the AI community who had migrated to the educational community. Their motivation was as follow: “Research in AI is founded upon the conjecture that cognition is, in some sense, a computational process than can be studied through the construction of “intelligent” systems that serve as functional models of the otherwise inaccessible processes of the human mind. If machines can be programmed to display intelligent behaviour, there is no reason, at least in principle, that systems could not be designed to assume the role of a skilled teacher. Since one-on-one tutoring is commonly considered the gold standard against which other methods of instruction are measured, the paradigm is founded on the proposition that education could be globally improved by providing every student with a personal (albeit machine-based) tutor” [249].

### 6.5.2 AI Methods and Techniques

AI methods and techniques enable computer programs and software applications to think and act intelligently and rationally. To achieve this goal, AI utilizes methods and techniques, like: Neural Networks, Genetic Algorithms, Reinforcement Learning, and Fuzzy Logic.

#### Neural Networks

Neural Networks do not rely on ruled-based programming for their performance. Instead, neural networks use learning algorithms to “tune” outputs to inputs. The technology finds use in situations in which rules are not explicitly available, and in which “tuning” inputs to outputs is easier than analyzing the internal reasoning process. Currently, DM uses neural networks to analyze the large volumes of data [40]. J.E. Villaverde et al. presents a feed-forward neural network to recognize automatically the learning styles of individual students according to the actions that he or she has performed in an e-Learning environment [257].

#### Genetic Algorithms

Genetic Algorithm is the population containing a number of trial solutions each of which is evaluated to yield fitness and a new generation is created from the better of them. The process is continued through a number of generations with

the aim that the population should evolve to contain an acceptable solution. Mu-Jung Huang et al. presents an approach to adaptive e-Learning based system that utilizes Computerized Adaptive Testing to identify the learner applicability for a learning topic, and then the Genetic Algorithm and Case-Based Reasoning (CBR) are employed to construct an optimal learning path for each learner. Mu-Jung Huang et al. presented a genetic-based curriculum sequencing approach that will generate a personalized curriculum sequencing [162]. Anna Hovakimyan et al. presents another Genetic Algorithm based e-Learning system that is based on so called “teaching-scenario” and capable of creating the appropriate sequence of the teaching resources from the set of all possible teaching resources based on quantity and quality [161].

## Fuzzy Logic

Fuzzy set theory provides a formalism in which the conventional binary logic based on choices "yes" and "no" is replaced with a continuum of possibilities that effectively embody the alternative "maybe". Formally, the characteristic function of set  $X$  defined by  $f(x) = 1$  for all  $x$  in  $X$  and  $f(x)=0$  for all  $x$  not in  $X$  is replaced by the membership function. J. Vrettaros et al. presents a fuzzy logic based correction tool that can be utilized in e-Learning systems. Test correction tool utilizes fuzzy logic rules in classifying students based on their answers and rigidity of correction levels [260]. Y. Chen uses feedback and fuzzy self-organizing map to improve adaptive e-Learning model via continuously updating teaching material by tracking learner grades and achievement level [88]. K. K. Loo presents genetic algorithm utilization in SmartTutor that can identify the weaknesses of a learner in an e-course and advice the learner how to do revision efficiently [195].

## SW Agents as AI Implementation

Software Agent is a program that performs some information gathering or processing task in the background. Typically, an agent is given a very small and well-defined task [255]. Software Agent is a physical or virtual entity which:

- Is capable of acting in an environment.
- Can communicate directly with other agents.
- Is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to optimize).
- Possesses resources of its own.
- Is capable of perceiving its environment (but to a limited extent).
- Has only a partial representation of its environment (and perhaps none at all).
- Possesses skills and can offer services.
- May be able to reproduce itself.

- Its behavior tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representation and the communications it receives.

### SW Agents Characteristics

Agents are capable of acting, not just reasoning. Actions affect the environment which, in turn, affects future decisions of agents. The main attributes and characteristics of Software Agents are [255]:

- Autonomy: A key property of agents is autonomy. They are, at least to some extent, independent. They are not entirely pre-programmed but can make decisions based on information from their environment or other agents. Agents have capabilities of task selection, prioritization, goal-directed behavior, decision-making without human intervention
- Mobility: Although this property is optional, it certainly characterizes some agent systems. It is interesting to note that in the natural world, agent intelligence is always associated with agent mobility (animals). Other living things (plants) have no intelligence.
- Persistency: code is not executed on demand but runs continuously and decides for itself when it should perform some activity.
- Social ability: agents are able to engage other components through some sort of communication and coordination; they may collaborate on a task.
- Reactivity: agents perceive the context in which they operate and react to it appropriately.

An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives. Software Agents are [136]:

- Clearly identifiable problem-solving entities with well defined boundaries and interfaces.
- Situated (embedded) in a particular environment; they receive inputs related to the state of their environment and they act on the environment through effectors.
- Designed to fulfill a specific purpose; they have particular objectives (goals) to achieve.
- Autonomous; they have control both over their internal state and over their behavior.
- Capable of exhibiting flexible problem solving behavior in pursuit of their design objectives; they need to be both reactive (able to respond in timely fashion to changes that occur in their environment) and active (able to act in anticipation of future goals).

## Intelligent SW Agents

IA is a software agent that exhibits some form of artificial intelligence. While the working of software agents used for operator assistance or DM (sometimes referred to as bots), are often based on fixed pre-programmed rules, "intelligent" implies the ability to adapt and learn.

- Ability to Adapt: Adaptation implies sensing the environment and reconfiguring in response. This can be achieved through the choice of alternative problem-solving-rules or algorithms, or through the discovery of problem solving strategies. Adaptation may also include other aspects of an agent's internal construction, such as recruiting processor or storage resources.
- Ability to Learn: Learning may proceed through trial-and-error, then it implies a capability of introspection and analysis of behavior and success. Alternatively, learning may proceed by example and generalization, then it implies a capacity to abstract and generalize.

In some literature IAs are also referred to as “autonomous IAs” [10], although autonomy is generally assumed for agents. IA systems should exhibit the following characteristics:

- learn and improve through interaction with the environment (embodiment)
- adapt online and in real time
- learn quickly from large amounts of data
- accommodate new problem solving rules incrementally
- have memory based exemplar storage and retrieval capacities
- have parameters to represent short and long term memory, age, forgetting, etc.
- be able to analyze itself in terms of behavior, error and success

## SW Agents Challenges

When several agents (inter)act they may form a multi-agent system. Characteristically such agents will not have all data or all methods available to achieve an objective (this can be referred to as "limited viewpoint") and thus will have to collaborate with other agents. Also, there may be little or no global control and thus such systems are sometimes referred to as swarm systems. As with distributed agents, data is decentralized and execution is asynchronous. Earlier, related fields include Distributed Artificial Intelligence (DAI) and Distributed Problem Solving (DPS). Multi-Agent architecture standards were attempted in order to force MAS standardization and global integration. Knowledge Query and Manipulation Language (KQML) was presented in order to support knowledge sharing among agents [130]. Knowledge Interchange Format (KIF) is a computer-oriented language for

the interchange of knowledge among disparate programs developed by the ARPA-sponsored Knowledge Sharing Effort [140]. The OMG group proposed a reference model as an attempt to standardize the development of agent technologies [258]. KAoS is described as "an open distributed architecture for software agents." The KAoS architecture describes agent implementations (starting from the notion of a simple generic agent, to role-oriented agents such as mediators and matchmakers), and elaborates on the interactive dynamics of agent-to-agent messaging communication by using conversation policies [70]. The Foundation for Intelligent Physical Agents (FIPA) is a multi-disciplinary IEEE standardizing group pursuing the standardization of agent technology. FIPA's approach to MAS development is based on a "minimal framework for the management of agents in an open environment" [24]. Unfortunately, as a result for all the standardization effort, there were no universally accepted commercially supported standard yet.

### **SW Agents based e-Learning Systems**

Utilizing Software Agents in e-Learning systems have been known to e-Learning researchers very well. Alfredo Garro and Luigi Palopoli [138] proposes an XML based Multi-Agent System to support Chief Learning Officers in defining roles, associated competencies and knowledge level required; manage the skill map of the organization; measure human resources competence gaps; support employees in filling their competence gaps as related to their roles; enrich a given courseware or create personalized learning paths according to feedbacks user provides in order to optimize the acquisition of needed competencies; and assist Chief Learning Officers in choosing the most appropriate employee for a given role. A strong well presented utilization of Multi Agent Systems in performing different tasks. Using software agents as recommenders to students has been presented by O.R. Zaiane [274]. Zaiane presented a recommender system in an e-Learning context that tries to "intelligently" recommend actions to a learner based on the actions of previous learners. This recommendation could be an on-line activity such as doing an exercise, reading posted messages on a conferencing system, or running an on-line simulation, or could be simply a web resource. The use of web mining techniques to build such an agent that could recommend on-line learning activities or shortcuts in a course web site based on learners' access history to improve course material navigation as well as assist the online learning process was presented.

### **ITS**

In the early 1970s a few researchers defined a new and ambitious goal for computer-based instruction. They adopted the human tutor as their educational model and sought to apply Artificial Intelligence techniques to realize this model in "intelligent" computer-based instruction [94]. ITS are complex, integrated systems that apply Artificial Intelligence (AI) techniques to problems of education and training. ITS uses Artificial Intelligence techniques to customize instructions according to the student's needs. ITSs strive to optimize learning of domain concepts and problem solving skills by means of adaptive, individualized coached practice on domain tasks [235]. The goal of ITSs would be to engage the students in sustained reasoning activ-

ity and to interact with the student based on a deep understanding of the students' behaviour. If such systems realize even half the impact of human tutors, the payoff for society promised to be substantial [94]. The closest computational analogy to an ITS is an ES, while the closest educational analogy is a Computer-Assisted Instructional (CAI) system [199]. ITS can be thought of as Intelligent Computer Aided Instruction (ICAI). ITS is an educational software containing an artificial intelligence component. The software tracks students' work, tailoring feedback and hints along the way. By collecting information on a particular student's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work.

### ITS Modules

ITS rely on three types of knowledge, organized in three separate software modules namely Domain, Pedagogical, and Student modules as depicted in figure 6.2 presented in page 135 [215, 235].

- Domain Knowledge Base Module: is the set of questions being taught. Represents subject matter expertise and provides the ITS with knowledge of what it is teaching. This module let the ITS know what it is teaching.
- Pedagogical Modules: contains the methods of instruction and how the knowledge should be presented to the student, enables ITS to know how to teach, by encoding instructional strategies used by the tutoring system. This module lets the ITS know how it is teaching.
- Student Module: Represents the student's knowledge, skills, and other attributes that affect how the student should be taught. This module lets the ITS know who it's teaching.

### ITS Limitations and Challenges

A variety of AI techniques are used to capture how a problem can be solved. Major ITS systems capture subject matter expertise in rules. Though rules enables ITS to generate problems on the fly, combine and apply rules to solve the problems, assess each learner's understanding by comparing the software's reasoning with theirs, and demonstrate the software solutions to the participant's, B. H. Sreenivasa Sarma and B. Ravindran argues that Rule based ITS challenges include [235]:

- There are many rules which a system can use to teach efficiently, but which are very difficult to encode.
- It is difficult to incorporate the knowledge that human teachers use, but cannot express. The machine tutors have a different set of data available than the human tutors, so the knowledge that could improve the tutor's performance is ignored.
- Rule-based systems are not adaptive to new student's behavior.

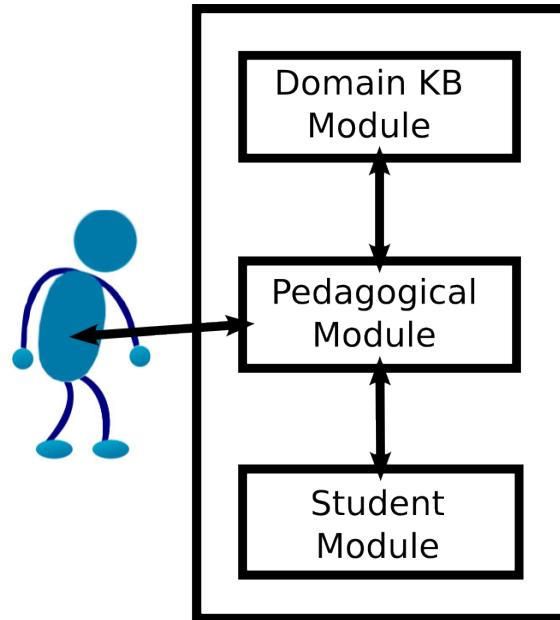


Figure 6.2: ITS Block Diagram

While there has been a sustainable research effort in the application of artificial intelligence to education over the past three decades with some notable success stories, intelligent tutoring has had relatively little impact on education and training in the world. There are several reasons for this lack of penetration [94]:

- ITSs are expensive to develop and until relatively recently, the necessary computing power was expensive to deploy.
- Ontogeny of the field and the consequences for software evaluation. The creative vision of intelligent computer tutors has largely arisen among artificial intelligence researchers rather than education specialists

### Examples of ITSs

Edward R. Sykes and Franya Franek present a Web-Based Architecture of an ITS for Remote Students Learning to Program Java that is called JITS [110]. JITS utilizes an Intelligent Module to define errors at student submitted code. Student code is passed to Java Parser; that checks student code for errors. When errors found, a fuzzy logic based semantic engine is used to define error spots and notify students to correct them, and the process takes place till code is verified. Besides, JITS contextually offers students' help by giving hints about the proper programming method that shall be used (for example: for loop instead of if-then statement). JITS is a Web-based solution that utilizes distributed architectural components. Most ITSs lack evaluation results; and JITS is one of them. Though JITS utilizes Web based architecture, it is always a challenge to integrate Agents based systems with external systems. It is believed that this integration challenge limits the widespread utilization and usage of software agents in real life world scenarios.

## 6.6 SOA in e-Learning Systems

SOA (SOA) is a design pattern that presents IT infrastructure and ISs architecture as loosely coupled, fine granular services that can address system requirements once they are presented either by adding new services or modifying existing ones. SOA also addresses enterprises information systems' inefficiency by enhancing reusability, thus theoretically, shortening ISs development time and effort required. Besides reusability, interoperability and integration are other main driving forces for adopting SOA in e-Learning systems. W3C defines Service as 'A Component capable of performing a task'. Service is 'A vehicle by which a consumer's need or want is satisfied according to a negotiated contract (implicit or explicit) which includes Service Agreement, Function Offered and so on'. SOA is the design pattern that utilizes services concept to achieve architectural advantages. W3C defines SOA as 'A set of components which can be invoked, and whose interface descriptions can be published and discovered'. This definition can be expanded to include the science, art and practice of building applications, so SOA can be defined as 'The policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer. Services can be invoked, published and discovered, and are abstracted away from the implementation using a single, standards based form of interface' [97].

### 6.6.1 Webservices as SOA enabler

Web services are applications that use standard transports, encodings, and protocols to exchange information [239]. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. W3C defines Web service as "A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a format that machines can process (specifically WSDL), Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with XML serialization in conjunction with other Web-related standards" [67]. Web service can also be defined as 'A programmatic interface to a capability that is in conformance with Wsnn protocols'. Wsnn protocols are present efforts in the W3C and more recently in OASIS to reach a Web service maturity model. Wsnn protocols include WSDL, SOAP, and XML [247]. SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment [144]. XML solves a key technology requirement that appears in many places. By offering a standard, flexible and inherently extensible data format, XML significantly reduces the burden of deploying the many technologies needed to ensure the success of Web services [169]. Web services is a general framework that expedites the sharing of heterogeneous data and software resources dispersed on the internet. The standard-based resource sharing and platform-neutral characteristics of Web services have motivated many organizations to apply the technology in diverse areas, such as SCM, virtual enterprise, homeland defence, e-government, and e-business [276].

## 6.6.2 SOA and e-Learning Systems

Many researchers have realized the importance of utilizing SOA in e-Learning systems. Integration, interoperability, scalability, and reusability are the main axis that SOA based e-Learning researchers attempt to address, solve, and enhance e-Learning systems. Though some researchers ignored the pedagogical features of e-Learning systems, others considered those features as the main motivator of adopting SOA in e-Learning. Declan Daggar [95] argues that e-Learning is moving from passive to active; because the learner is getting more involved with the learning experience. Besides, technology emerging from the adaptive hypermedia, semantic web, mobility and distributed computing communities are being widely employed in on-line learning, and steps towards Service-Oriented eLearning platforms needs to be taken. Declan Daggar et al. classifies LMSs into three generations: First Generation “Monolithic”, Second Generation “Modular”, Third Generation “Service-Oriented”. First generation focused on presenting learning content without tracking learner activities. Second generation enhanced the learning process but cannot stand handling new technologies, interoperability and integration between evolving systems like third generation shall do. Declan Daggar et al. argues that interoperability needs to be on control basis not just on data basis. Service-Oriented LMS technologies include adaptive hypermedia and semantic web. Though Service-Oriented LMS standards are still evolving, attempts to address service-oriented LMSs are many, and include the abovementioned research project, and many other projects like the ones described in the following sections.

### **moodle and Webservices**

Al-Ajlan and Zedan [50] presents a research work that focuses on “assignment” modules based on moodle and the steps to enrich it with Web services. Al-Ajlan and Zedan argues that Web services is important and required in integrating different moodle(s) resources; especially assignments, and to provide the capability for more than one instructor to be working on the same course. Instructor can search for the best assignment within different moodle(s) instances and retrieve it. Al-Ajlan’s proposed work utilizes PHP based SOAP Web services to integrate best with moodle. There is no evaluation presented at his research, however it presents a good utilization of UML in describing Web services specifications. Though the research work title includes SOA, the common misusage of Web services as SOA is clear in this research. Though Web services are main SOA enabler, SOA is not just Web services.

### **Services based e-Learning Systems and Reusability**

One of the good utilization of SOA in e-Learning is presented by Moon Ting Su et al. [251] to address the reusability capabilities provided by SOA. Moon Ting Su et al. argues that Instead of building an e-Learning system from scratch, it can be assembled by choosing the required functionalities from a set of Web services related to e-Learning. Moon Ting Su et al. study results in a set of Web services for the e-Learning domain. The description for each of the Web services is as follows:

- Assessment Web service: This service supports the delivery and scoring of assessment. Lecturers are able to create/edit/delete quiz or assignment, quiz or assignment questions using this service. Students on the other hand are able to take the quiz online at specific time and duration defined by the lecturers. The result of objective-type question will be displayed immediately after the students finish the quiz. Students' answer for subjective question can be retrieved from the server for marking later.
- Grading Web service: This service supports grading based on the marks. Marks submitted by lecturers will be graded automatically based on the grading scheme uploaded by the administrator of the e-Learning system. Feedback can be submitted together with the grade.
- Marking Web service: This service provides the automated marking functionalities for multiple choices assessment. For subjective assessment, the marking scheme is defined using rubrics. Rubric is a guide for assessing a piece of student's work and aids in achieving consistency in marking.
- Course Management Web service: This service allows lecturers to manage the courses (uploading, editing and deleting of lecture materials). It also allows the users to manage their favourites' resources. Users can also search, view and download the learning resources. The searching is done based on the Learning Object Metadata (LOM) metadata elements provided by the creator of the learning resources.
- Metadata Web service: This service is developed based on LOM standard. It allows users to upload the learning objects together with its metadata. The metadata is useful in managing, searching and locating the learning object.
- Registration Web service: This service allows administrator of an e-Learning System to create new users, edit and delete the existing user profiles and also delete particular users.
- Reporting Web service: This service generates the performance report for quiz by using Crystal Report.

In the abovementioned study, there are different points of strength and excellence. The main functionalities of typical e-Learning systems are built using Web services. The strength of this approach is reusability and interoperability. Developing a new e-Learning system will involve assembling the required Web services. The e-Learning system can be developed and run on different hardware and software platforms. However, the disadvantage of this approach is if the server that hosts the Web services is off-line, the e-Learning system which depends on these Web services will not be operational. Another important feature of this study is the use of rubrics in guiding the instructor to evaluate subjective assessment. Rubrics improve consistency in the evaluation of students' work especially when there is a large number of student and multiple instructors are involved in the evaluation. Another good point of the proposed research is the commitment to E-Learning Framework (ELF) and e-Learning standards as depicted in the architecture presented at the study. However

and to repeat from almost all services based LMSs, SOA is not Web services. This chapter presents a bunch of Web services that collectively do not form SOA in any manner. Our experience in evaluating SOA in e-Learning highlighted that problem. Because we started by building services for every component without SOA governance and without defining what are the system features that needs to be exposed as services, we found ourselves building the entire system as services from scratch, and it seems this is the case in this research too. Besides, there is no indication of implementation at all, and so evaluation was not mentioned at all. It is really rare to find evaluation for SOA based e-Learning systems. May be because SOA adoption in e-Learning systems is new to some extent, or may be because it is really too much easier to present frameworks and architectures than it is to implement them.

### Services based e-Learning Systems and Scalability

Another SOA based framework for e-Learning Systems is presented by Chih-Ping Chu et al. [90]. Chih-Ping Chu et al. addresses the scalability issues of educational institutions and propose a SOA based e-Learning framework to address scalability capabilities of SOA. This research handles the educational institutions incapability of maintaining large scalable LMSs by grouping LMSs services (based on Advanced Distributed Learning (ADL) Seven services as depicted below) into local and global groups. Local services group which constitutes the “Local LMS” are provided and maintained internally while other services can be spread over the internet. ADL modularize LMS into seven services: Tracking, Delivery, Learner Profile, Course, Sequencing, Content Management, Testing/Assessment. Chih-Ping Chu et al. argues that tracking and delivery services shall compose local LMS while rest of the services can be spread globally. Authors argue that maintaining learning material might exhaust educational institution resources, so Local LMS will not store learning material, however it will acquire it from different places. LMS is described as a highly generalized model consisting of seven parts that provide the functionality of learner/learning tracking, content delivery, course import/export and content sequencing:

- Tracking service: takes a learner request and traces the learning status of the learner. It is the LMS entry point for a user to request learning content, and invoke related services, according to that request. It also has the responsibility of tracking the learning process and the learner’s keystrokes or mouse clicks and whether they abort the learning process.
- Delivery service: delivers the learning content to the learner. It is better from performance view to keep both tracking and delivery services both on the same server. Besides, static binding between services will enhance the performance of the learning process.
- Learner Profile service: sharing user profile between different LMSs is a goal here (because users can be learning via different LMSs). Standardization of learner profiles is important. Authentication between LMSs is required before receiving user profiles.

- Sequencing service: during the learning processing, selects the appropriate content from either the local content repository or a remote content provider. Remote content providers register their content information with the Register server. In proposed framework, there is no local content repository, and all content is supplied by content providers. The sequencing service retains information about the content provider and learning process, since there is no guarantee that a tracking service will bind with the same sequencing service as the learning progresses.
- Course/Content Management service: selects content and provides it to the sequencing service.

Chih-Ping Chu et al. presents a great work from our perspective. Though proposed framework combined content with course services to be almost one service, and though they did not mention Assessment/Test service at all, they still presented a SOA based e-Learning framework that included modifications to the Original ADL LMS model to fit with the research objective.

### **Services based e-Learning Systems and Integration**

Zhengfang Xu et al. [272] propose a Web services oriented framework for e-Learning systems aimed at providing a flexible integration model in which all the learning components and applications are well defined, effectively discovered, and loosely connected through using Web services as the main system component to achieve that objective. However, Zhengfang Xu et al. argues that there is a need for a standard mechanism for supporting complete automation through all aspects of end-to-end learning process that includes finding suitable learning components or learning management services, getting information about their services and invoking their services. Though automation of complete end-to-end learning process might seem to be a goal to achieve, doubts about the effectiveness of such a learning approach is under questions. Arguments that prove the efficiency and effectiveness of pure e-Learning programs are not completely confirmed. There is a need of human tutor to support blended learning (at least at some level of the learning process). This research did not include any details about evaluation of the proposed work. Experience proved to us that SOA is not just Web services. Evaluation of utilizing SOA in integrating LMS and UMIS is available in details. Though we don't completely agree with this research project, however it still confirms the need to present Web services to LMSs to gain advantages like flexibility, and accessibility, and overcome challenges like integration and interoperability. Another SOA based e-Learning research project is the one presented by A. M. Riad and H. A. El-Ghareeb [113] that presents a SOA based CMS to address different integration challenges. One of the critical limitations of a newly established educational institution is the lack of available well prepared courses. It is more applicable to use widely available courses that might be higher in quality than preparing new courses. Current CMSs do not exploit courses shareability. To address this shortage, a CMS is presented to highlight automated discovering and importing of courses maintained and managed by external CMSs. Proposed CMS architecture utilizes SOA as a design pattern

to integrate different CMSs on service level. Proposed CMS consists of two layers: Presentation layer and Service Layer. Presentation layer is responsible for interacting with user either via displaying information or receiving user inputs. Service layer contains core system services. Service layer is divided into three sub layers: orchestration layer, application services layer, and agents layer. Orchestration layer holds business logic required by system processes. Application services layer contains set of stateless services that are capable of performing certain tasks. Agents layer presents the suggested required software agents to serve the overall system. Suggested agents are: Discoverer, Ranker, Tracker, and Analyzer software agents. Integration between software agents and Web services is achieved by utilizing SOA. Proposed CMS facilitates integration between different CMS in order to share resources of educational institutions. Other SOA based e-Learning research projects that concern integration is the one presented by Michel Derntl and Juergen Mangler [101]. Michel Derntl and Juergen Mangler argues that Web services have drawn the attention of learning technology researchers and practitioners, e.g., for decentralized, integrated support of Web-service-based agents, for contract-based provision and discovery of distributed Reusable Learning Object (RLO) repositories, or for enhancing the functionality and interoperability of existing learning technology applications, to mention a few. All of these approaches employ Web services to increase extensibility and flexibility of existing solutions and to foster standards-driven development, dissemination, and usage of desired functionality. They presents an open, standards-based learning technology architecture that uses distributed Web services to support a broad variety of blended learning scenarios and administration asks. This project addresses blended learning where; according to the author, not much projects address efficiently. SOA has been adopted in different ways in different e-Learning projects and from different perspectives, and there is much research to be done regarding this point to highlight the advantages and shortages of SOA within e-Learning domain.

### **Services based e-Learning Systems and Interoperability**

“Scalable Adapter” design pattern constitutes a software architecture that can be used to create interoperability between differently targeted educational tools. The key idea behind the pattern is to add a small “data adapter” to each learning environment. The adapters can then access arbitrary (scalable) parts of the data of “their” learning environment and exchange this data with other adapters. These changes are not costly and usually easy to create since the existing systems do not have to be changed but merely need to be extended. This design pattern is used mainly to provide interoperability features between different e-Learning tools used in educational institutions. Advantages of this design pattern are many and three different case studies presented in this research. However, integrating different applications on data level has never been the appropriate solution for all system integration problems. Andreas Harrer et al. presented an interoperability approach based on data [150]. There are driving forces for applying that approach, include: easier exchange of data, reduced development time, and reduced maintenance costs. Architects and programmers do not have to change current already running systems; instead they need to add new adapters that connect systems to-

gether. However, connecting systems together via adapters is not “always” the optimum solution. Data based interoperability lacks application logic interoperability and might stand as an obstacle against Business Processes adoption within educational institutions. However, data based interoperability cannot be ignored. Another SOA based interoperability research project is the one presented by A. M. Riad and H. A. El-Ghareeb [114]. This research project presents a SOA based Assessment Management System (AMS) to address Mobile Assessment as one of the e-Learning activities. Mobile Learning (M-Learning) is an approach to e-Learning that utilizes mobile devices and is strongly recommended to be enabled by LMS. Assessment is one of the learning activities that can be achieved electronically and via mobile devices. Mobile assessment refers to the capability of conducting assessments via mobile devices. Mobile assessment relies on external services that are not part of the LMS. Providing interoperability between different external systems and services to be virtually part of the educational institution LMS is one of integration and interoperability challenges. Authors presented an extension to the SOA based LMS developed in the faculty of computers and ISs at Mansoura University to address mobile assessment. Proposed architecture consists of two layers: Interface layer, and Service layer. Interface layer interacts with instructors, learners, and business managers via human interface (portals), and with external organization services via machine interface (Web services). Service layer contains core LMS services and has three sub layers: Orchestration, Application Services, and Agents layer. Orchestration layer holds business logic presented by system processes as executable services. Application Services layer contains set of stateless Web services that are capable of performing certain tasks related to system entities. Agents’ layer presents the suggested required software agents to serve the overall system. Agents’ layer presents Tracker software agent; which is responsible of tracking students’ non-conducted assessments and taking appropriate actions to inform them. Other interoperability approach that acts on “service basis” is presented by Pedro J. Muñoz-Merino et al. [177]. Pedro J. Muñoz-Merino et al. highlights the importance of standards and its role in interoperability between different applications. Also, they argue that layering is one of the patterns that are important while considering e-Learning solutions. Also, LMSs’ future tendency is to be service-oriented. In this scenario, LMSs are based on modular components and they can support different services that do not stick to a specific platform. They envision that the ideal scenario is one in which all the different educational services can be interoperable among different LMSs; and in which the entire design of different LMS courses can be done off-line (outside of LMSs) in an easy way for teachers without high technological knowledge using proper authoring tools’ and next these courses can be imported within the different LMSs. Based on Open Knowledge Initiative (OKI) architecture depicted in figure 6.3 presented in page 143, they present their architecture.

Infrastructure layer represents the final resources of an institution, such as file systems or databases. The Common Services are services that are used by several educational applications, such as authorization or authentication. The Educational Services are specific educational modules like assessment or Course Management. Finally, the Educational Applications are the applications a user directly interacts with and these educational applications can use the implemented educational and

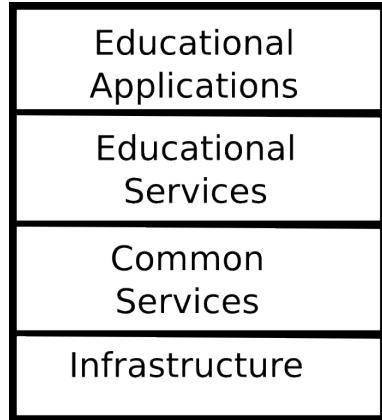


Figure 6.3: OKI Layered Architecture

common services. The IMS Abstract Framework architecture is very similar to the one defined by OKI, and a perfect relationship among layers of both architectures can be established. Both architectures capture the strong importance of LMS services. Besides interoperability, Reusability of courses, services, and all other materials is one of the main motivators of researchers to highlight the importance of layering and standards to achieve interoperability and reusability between different distance educational systems.

### **Services based e-Learning Systems and XML**

Theodore K. Apostolopoulos and Anna Kefala propose an e-Learning framework based on the design and implementation of a middleware [57]. Authors ought to adopt technologies that are standardized and widely deployed in both e-Learning systems and network infrastructure layers. The general architecture of the e-Learning management scheme is based on the configurable component-based middleware architecture for deploying e-Learning services. Each component has one or more agents, which maintain a local XML-based Management Information Base (MIB), and communicate with manager residing at the service or session management component. The agent communicates with the middleware via a light protocol such as SOAP. Another research project that utilizes XML in e-Learning is the one presented by Wan-Jen-Chang et al. [85]. Wan-Jen Chang et al. proposed an e-Learning platform with the required functions to provide information to everyone in anytime and at anywhere. The three functionalities are:

- Learning contents presented by vision (i.e., Web browser) and voice (i.e., voice device, phone, cell phone) to provide learners who are less technology-literacy and visually disabled to join learning.
- KMLS associates with collaborative learning (i.e., learning communities) to transform tacit knowledge into explicit knowledge by consultation, assessment, discussion and sharing experiences.

- The subject of life education is everyone needs to face and to develop unrestrained by age, gender, and avocation.

Utilizing VoiceXML to support Voice User Interface (VUI) was a powerful idea and the integration presented between different system components via XML utilization was a good contribution.

### Comments on Services based e-Learning Systems

SOA utilization from different researchers' perspectives leads to different SOA based e-Learning systems architectures. From full SOA functionalities architectures to just Web services enabled architectures are available by different researchers. SOA based e-Learning systems frameworks are available; without much evaluation and further analysis of points of strength and weaknesses. Architectures that tend to support fully automated learning process and architectures that tend to support blended learning are available. SOA based systems that support point-to-point integration and interoperability, and architectures that support Middleware based integration and interoperability also exist. Each approach has its own points of strength and points of weakness. Point-to-point integration enhances performance. On the other hand, middleware based architectures are more flexible, scalable, and fault tolerant. Evaluation of utilizing SOA in e-Learning systems need to be more studied to highlight advantages and shortages of utilizing SOA in e-Learning, and still SOA based architectures vary a lot from a research project to another.

## 6.7 Summary

In 2003, Charles A. Shoniregun, and Sarah-Jane Gray [240] wondered if e-Learning is a risk or a future, and it is believed that e-Learning is an important part of the future that depends on how e-Learning researchers and application designers will present to support and enhance. By surveying e-Learning research activities in the last decade, it is clear that e-Learning has been presented to everyone, and challenges have moved to a next step: the step of efficiency and effectiveness. Either e-Learning has achieved the required results or has not yet, that only means e-Learning researchers are facing new challenges. Adaptive e-Learning is one of the proposed solutions to exploit the unlimited advantages of e-Learning. Tutoring is a solution to learning limitations and challenges and ITS have been a reputation research subject [86]. e-Learning can present fined adaptive e-Learning solutions supported by the intelligent methods and techniques presented over long time of research. Services based learning systems represent the third generation of learning systems that supports integration, interoperability, and scalability. Services based LMSs are recommended in order to fulfil e-Learning systems architectural requirements. Adaptive e-Learning that is supported by intelligent techniques is the proposed solution to present efficient and effective learning. To the knowledge of the authors, there is no article have been found to merge adaptive and intelligent techniques with SOA in the subject of e-Learning. As a result, we were concerned with our activity to fill this gap.

## 6.8 Selected Topics

### 6.8.1 Deploying Services in the Cloud

The “cloud” refers to a virtual infrastructure in which dynamically scalable, virtualized resources are provided as a service over the Internet. With cloud computing, organizations choose between renting an application, process, service, or infrastructure or completely outsourcing one or more of these aspects to a cloud provider. Loose coupling is a key litmus test for services being deployed in a cloud. That is, the optimal scenario for moving services to a cloud is when one can bundle five fundamental elements of SOA and deploy them as an independent unit. These five elements are a set of services, components, processes, data, and rules that are closely aligned with one another and are collectively more loosely coupled and less dependent on other groups of services, components, processes, data, and rules. One of the key principles and value propositions of SOA is its geographic independence. This geographic autonomy implies that the services clustered together are not in themselves bound to a data center or geographic location. Instead, they can indeed be deployed within a cloud that may span multiple smaller cloud structures, starting from inside a private cloud and moving into a public cloud and hybrid cloud architecture to suit security, performance, and scalability requirements and service-level requirements or agreements. Cloud computing provides new architectural deployment options for services. The adoption of services requires increased thinking about security, and the use of cloud services further increases the risk of having non-secured services. The Cloud Security Alliance (CSA) has created a controls matrix to address the security risks by a cloud provider. Services in the cloud must be secured.

### 6.8.2 Cloud Computing as SOA Infrastructure

Cloud computing is Internet-based computing where resources such as software, storage, services, applications, and development tools are shared. The intersection between cloud computing and SOA lies in what resources that live in a cloud are suitable for sharing in the infrastructure. A main characteristic of a cloud environment is that it provides elasticity and dynamic capacity on request. Many of the capabilities of a cloud environment are expressed in terms of infrastructure-level services (e.g., platform as a service, infrastructure as a service, application as a service, or even business processes as services). These infrastructure-level services create an optimal deployment environment or operating environment for your infrastructure or subsets of your infrastructure. Technologies and vendors can be identified who provide an SOA runtime in the cloud. This is augmented with appropriate security restrictions and access restrictions that occurs when multiple organizations run their applications in the same cloud. Some organizations may choose to implement private clouds. Private clouds are often used to describe a cloud environment contained within the firewall of an enterprise, contrasted to a public cloud accessible outside the organizations firewall. Using the cloud as a run-time environment for SOA is increasingly an option as service-level characteristics such as security and agreements are provided. These will motivate more corporations to consider running

parts of their runtime infrastructure in a cloud environment (whether private, public, or a hybrid). Cloud computing is a deployment architecture, whereas SOA is an architectural style. Cloud uses the principles of SOA to build services that will be deployed in the SOA infrastructure. Elasticity is a unique feature of the cloud that can be viewed as the amount of strain the cloud infrastructure can withstand while either expanding or contracting to meet the demands placed on it. These aspects of cloud computing, along with other provided attributes such as partitioning and separation of concerns, are attractive infrastructure features for service consumers and service providers. A key attribute for SOA and its ecosystem is to promote change as business demands changes, and leveraging cloud functionalities may allow organization to change SOA infrastructure faster. Security continues to be the biggest impediment to using cloud services, which is not ameliorated with SOA.

## 6.9 Review Questions

1. What are the limitations of current LMSs?
2. What are Adaptive e-Learning Systems?
3. What is AEHS? What are its main features?
4. What is Intelligent e-Learning Systems?
5. How can we utilize SOA in e-Learning Systems?
6. How can Cloud Computing be related to SOA?

## 6.10 Exercises and Labs

In Labs, you shall be getting familiar with Java programming language, as it is the programming language we will be using to build, deploy, and consume different online Webservices and SOA systems presented in this book.

# Chapter 7

## Course Management Systems

### 7.1 Introduction

This chapter presents the proposed CMS. Proposed CMS facilitates integration among different CMSs. An automated course search, import, and deposit process is presented. This automated process requires governance of business rules. Business rules might be limiting system efficiency, so they must be monitored and modified when needed. Manage business rules process is presented to achieve this goal. Utilizing SOA to integrate Web services and software agents in CMSs highlighted the unlimited advantages of Web services and its capabilities to facilitate software agents' integration within systems.

### 7.2 CMS Analysis

Proposed CMS addresses two processes namely Search and Manage Rules as depicted in the use case diagram at figure 7.1 presented in page 147. Search process enables instructor and CMS to search for courses within course repositories managed and maintained by internal and external CMSs, import, and display those courses. There are managerial issues that should be addressed to enable this process. Managerial issues are managed at Manage Rules process.

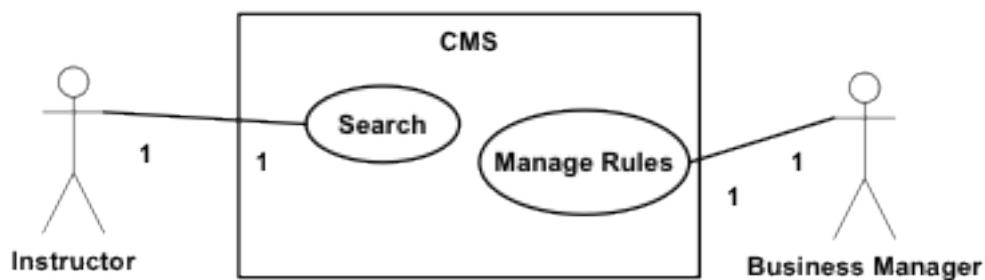


Figure 7.1: CMS Usecase

## 7.3 Proposed CMS Architecture

Figure 7.3 presented in page 149 depicts the proposed CMS architecture that enables automation of courses integration. From architectural perspective, proposed system consists of two main layers that present its frame: Presentation layer, and Service layer.

### 7.3.1 Presentation Layer

It is the layer that presents user interface. System's scope is limited to instructors and business managers. Those are the end users of the system. Display service is the service that is responsible for preparing dynamic output prepared to users, calling the suitable web page to display data. More than web page is prepared to address different functionalities. Display service passes suitable set of data to appropriate pages, and acquires input from users incase there are. Separation of interface design and implementation from business logic has proven many advantages.

### 7.3.2 Service Layer

Service layer consists of three sub layers: Orchestration Layer, Application Services Layer, and Agents Layer.

- **Orchestration Layer:** Orchestration layer manages interaction details required to ensure that service operations are executed in a specific sequence. Sequences are determined based on processes supported by system. Orchestras within this layer are: Course Manager, Rules Manager, Exception Manager, Financial Manager, Mail Manager, Faculties Manager, Instructor's Manager, and Feedback Manager.
- **Application Services Layer:** Application Services are set of stateless services perform certain. Process is the summation of tasks performed by one or more application services layer in the sequence that is maintained at orchestration layer services. Services of Application Services Layer are: Discover/Recommend Display, Check Capability, Import, Insert Rule, Pay, Raise Exception, Manage Rules, Send Mail, Manage Courses, Search Process Exception Manager, Manage Rules Exception Manager, Manage Faculties, Manage Instructors, and Manage Feedback.
- **Agents' Layer:** Specific task agents are required to serve system. Software agents are always the optimum solution for track and analysis tasks. Software agents are: Discoverer Agent, Ranker Agent, Analyzer Agent, and Tracker.
  - Discoverer: It is the software agent that is responsible for preparing list of external course providers and indexing metadata about courses.
  - Ranker: It is the software agent that is responsible for ranking different system's components based on ranking rules. System components that should be ranked are: Universities, Faculties, Instructors, and Courses.

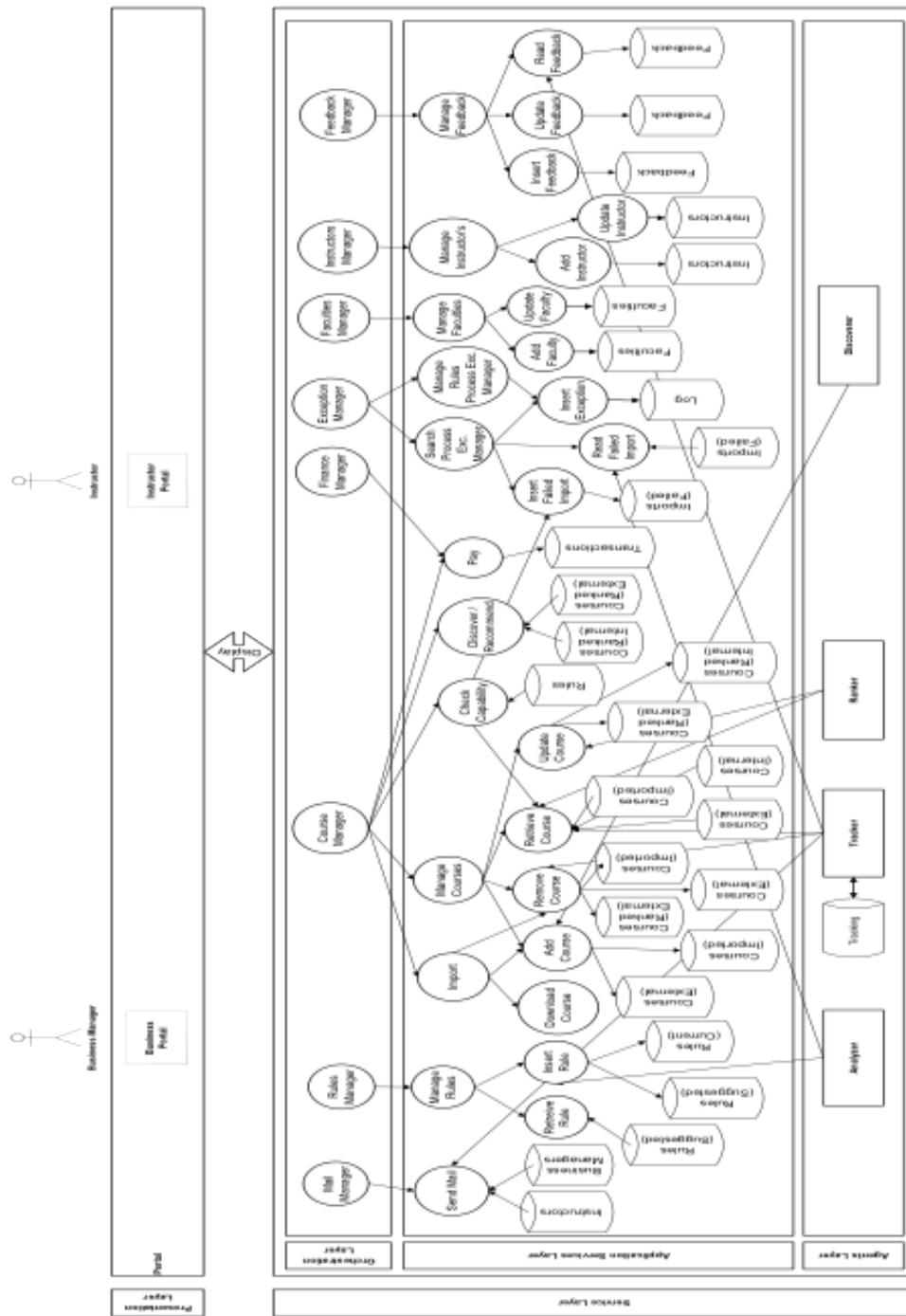


Figure 7.2: Proposed CMS Architecture  
Online: <http://zoom.it/e7X3>

- Tracker: Proposed CMS depends heavily on instructor's feedbacks to enhance system's performance.
- Analyzer: Analyzer is the software agent that is responsible for analyzing previously stored failed imports by detecting the most cause of failed imports and suggesting new conditions to decrease such problem. Suggested rules are inserted temporarily in the suggested rules DB, waiting for the approval/denial of business manager. CMS is one of the LMS components that is interested in automating the activities related to courses discovery, downloading, and tracking.

## 7.4 Processes Analysis

### 7.4.1 Search Process Analysis

Search process is initiated by Instructor, managed and maintained by Course Manager. Figure 7.4.1 presented in page 151 depicts analysis of Search process. Highlighted shapes present activities performed by course manager. Other shapes present non-course manager activities. Non-course manager activities are presented within the architecture as stand alone services that are called explicitly by course manager.

### 7.4.2 Manage Rules Process Analysis

Manage Rules process is initiated by Business Manager, managed and maintained by Rules Manager as shown in figure 4.11. Highlighted shapes present activities performed by rules manager. Other shapes present non-rules manager activities. Non-rules manager activities are presented within the architecture as stand alone services that are called explicitly by rules manager.

## 7.5 SW Agents Analysis

This part presents the design process of the four software agents highlighted in the CMS. The four software agents are: Discoverer, Ranker, Tracker, and Analyzer.

### 7.5.1 Discoverer Agent Analysis

It is the software agent that is responsible for preparing list of external course providers and indexing metadata about courses. Discoverer agent makes use of standard SOAP/XML based Web services that integrates internal and external applications [1]. Discoverer software agent integrates internally with exposed Web services like Course manager in order to insert data and metadata about courses. Figure 7.5.1 presented in page 153 depicts Discoverer software agent architecture. There are three main interfaces for Discoverer agent: Internal interface, external interface, and human interface. Discoverer software agent integrates externally with educational institutions exposed Web services in order to initiate and maintain an internal catalog of external courses providers and courses. Web services technologies

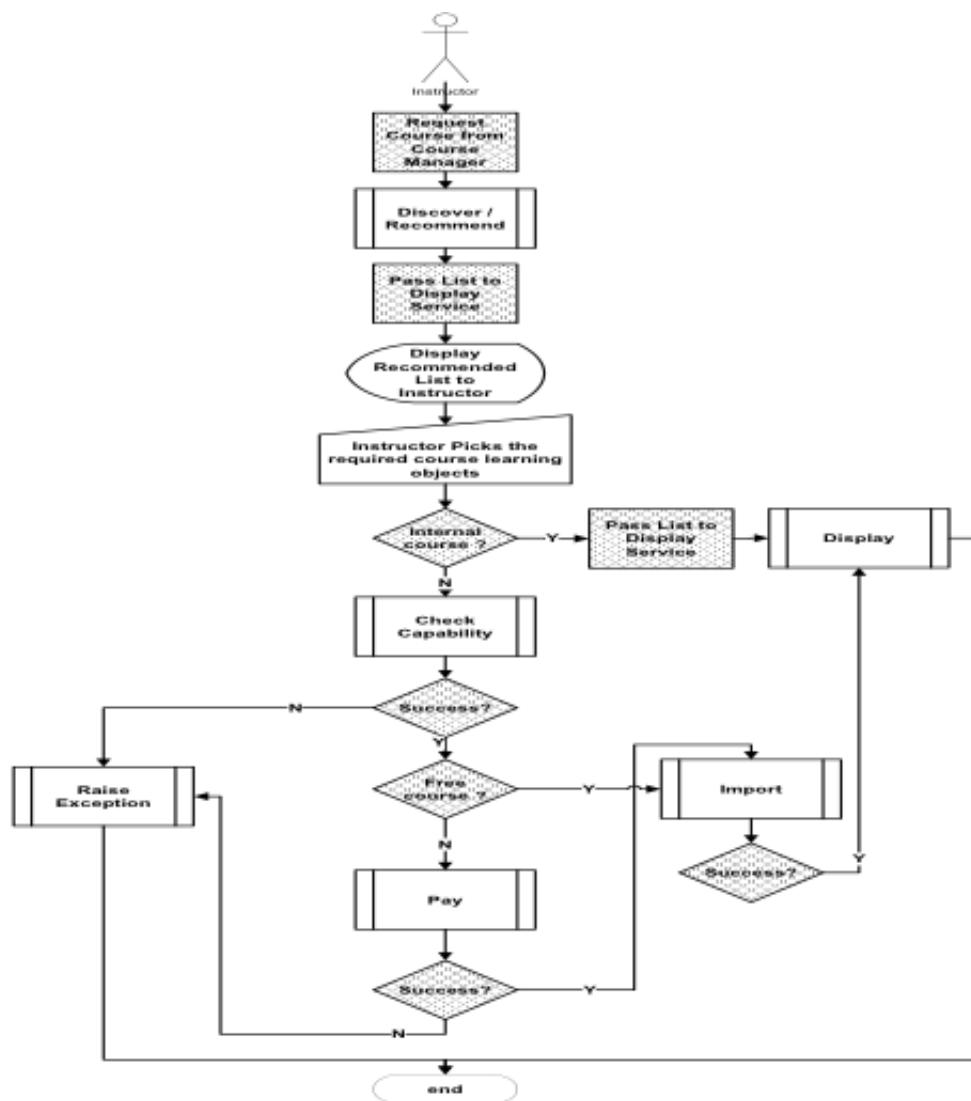


Figure 7.3: Search Process Analysis  
 Online: <http://zoom.it/P5fE>

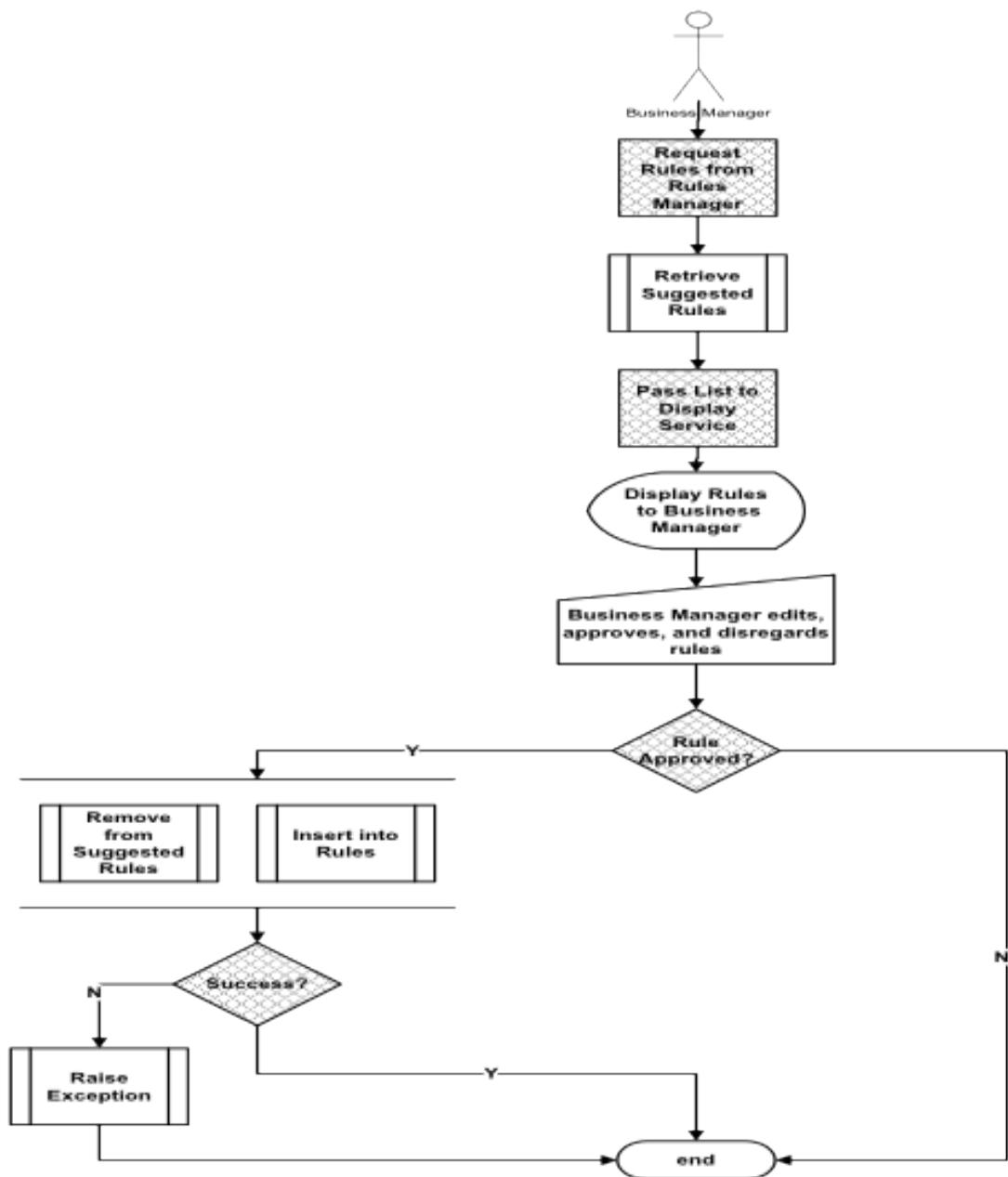


Figure 7.4: Manage Rules Process Analysis

Online: <http://zoom.it/IXdN>

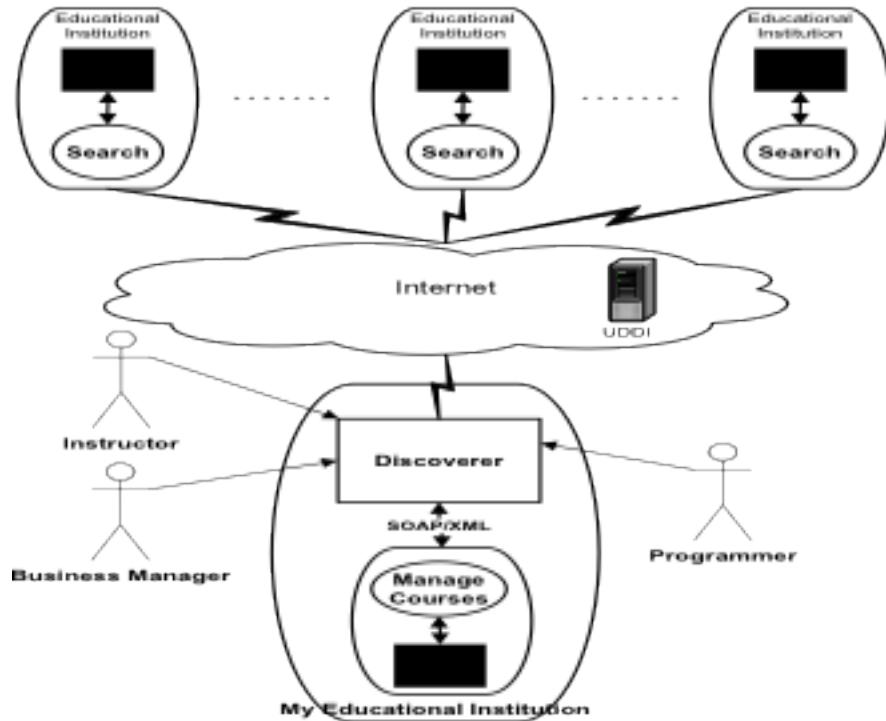


Figure 7.5: Discoverer Agent Architecture  
 Online: <http://zoom.it/apLcP>

enabled integration flexibility and maintainability [1]. Internal interface presents integrating Discoverer software agent with CMS's Web services on service level. Discoverer needs to consume Manage Courses service in order to update courses DB. Discoverer is not directly inserting data into courses DB. All operations have to be done through Manage Courses Web service in order to obtain service level integration benefits. Human interface is required to enable instructor and business manager insert data about courses and courses' providers. Programmer interface is required to inform Discoverer about available external Web services. UDDI provides a platform-independent way of describing services, discovering businesses, and integrating business services using the Internet [30].

### 7.5.2 Ranker Agent Analysis

It is the software agent that is responsible for ranking different system's components based on ranking rules. System components that should be ranked are: Universities, Faculties, Instructors, and Courses. Different formulas used to rank each of them. Ranker agent integrates internally with the proposed system on service level via system's exposed Web services. Ranker agent consumes suitable Web services to insert, update, retrieve, or delete items that to be rank. Figure 7.6 presented in page 154 depicts a Course Ranker process design. Course Ranker consumes two services in the early phase of running, and another two services as depicted.

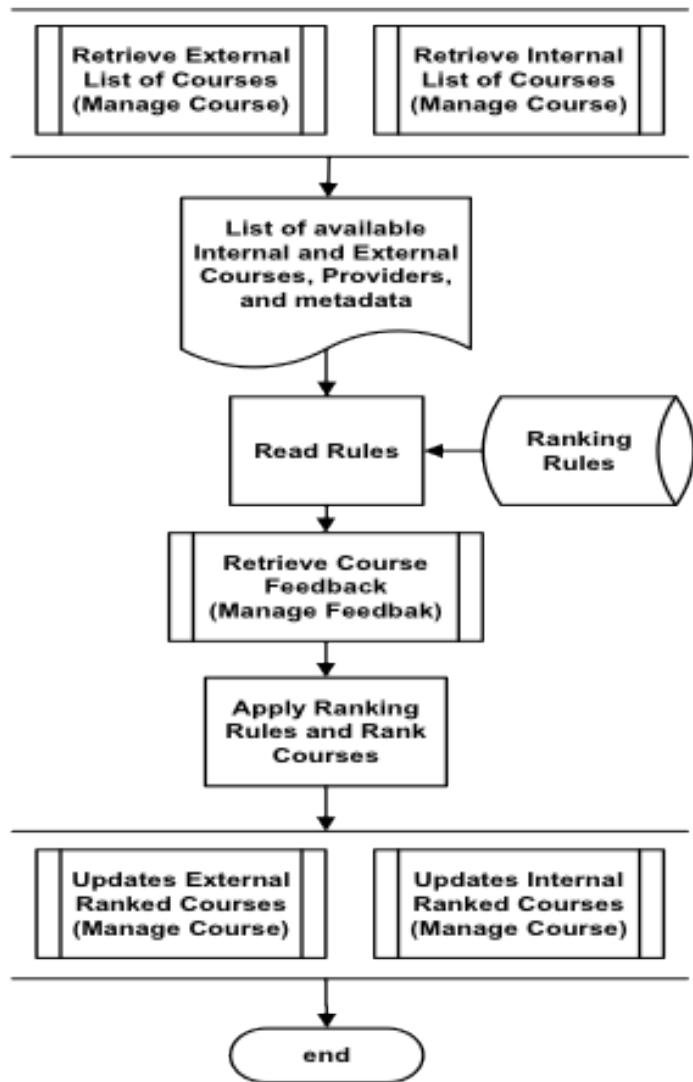


Figure 7.6: Course Ranking Process Analysis

### 7.5.3 Tracker Agent Analysis

Proposed CMS depends heavily on instructor's feedbacks to enhance system's performance. Figure 7.7 presented in page 156 depicts of tracking instructor's feedback of imported courses process design.

### 7.5.4 Analyzer Agent Analysis

Analyzer is the software agent that is responsible for analyzing previously stored failed imports by detecting the most cause of failed imports and suggesting new conditions to decrease such problem. Suggested rules are inserted temporarily in the suggested rules DB, waiting for the approval/denial of business manager. Analysis of failed imports tend mainly to increase system overall performance by adapting system to meet/exceed instructor's expectations. Figure 7.5.4 presented in page 157 depicts Analyzer agent process analysis.

## 7.6 Design of CMS

CMS design include design details of the Search, Manage Rules Process Design, and designing the software agents. Each process design includes defining composing services and designing each of them.

### 7.6.1 Search Process Design

Figure 7.6.1 presented in page 158 depicts Search process design. Search process utilizes more than one service. Services that are utilized by Search process are: Discover/Recommend Display, Check Capability, Manage Courses, and Import.

#### Discover/Recommend Service

It is the service that is responsible for searching internal and external list of ranked courses to prepare a list of ranked courses that satisfies user query. Recommend is the service responsible for ranking results prepared by Discover service by relevance of its keywords and user query keywords. Detailed design for Discover/Recommend activities is depicted in figure 7.6.1 presented in page 158 under Discover/Recommend side title. Figure 7.10 presented in page 159 depicts tables required by Discover/Recommend to achieve tasks. Courses-External-Ranked, and Courses-Internal-Ranked are the tables required by Discover to prepare list of courses. Keywords table is the table required by Recommend to rank list by relevance to user's keywords.

#### Display Service

It is the service that is responsible for preparing data that will be displayed, passing this data to the proper page at presentation layer, and acquiring input from end user. Display service manages different kinds of data. Those data include:

- Display list of courses: at presentation layer, there should be a display course page that is waiting for the dataset that will be passed by course manager.

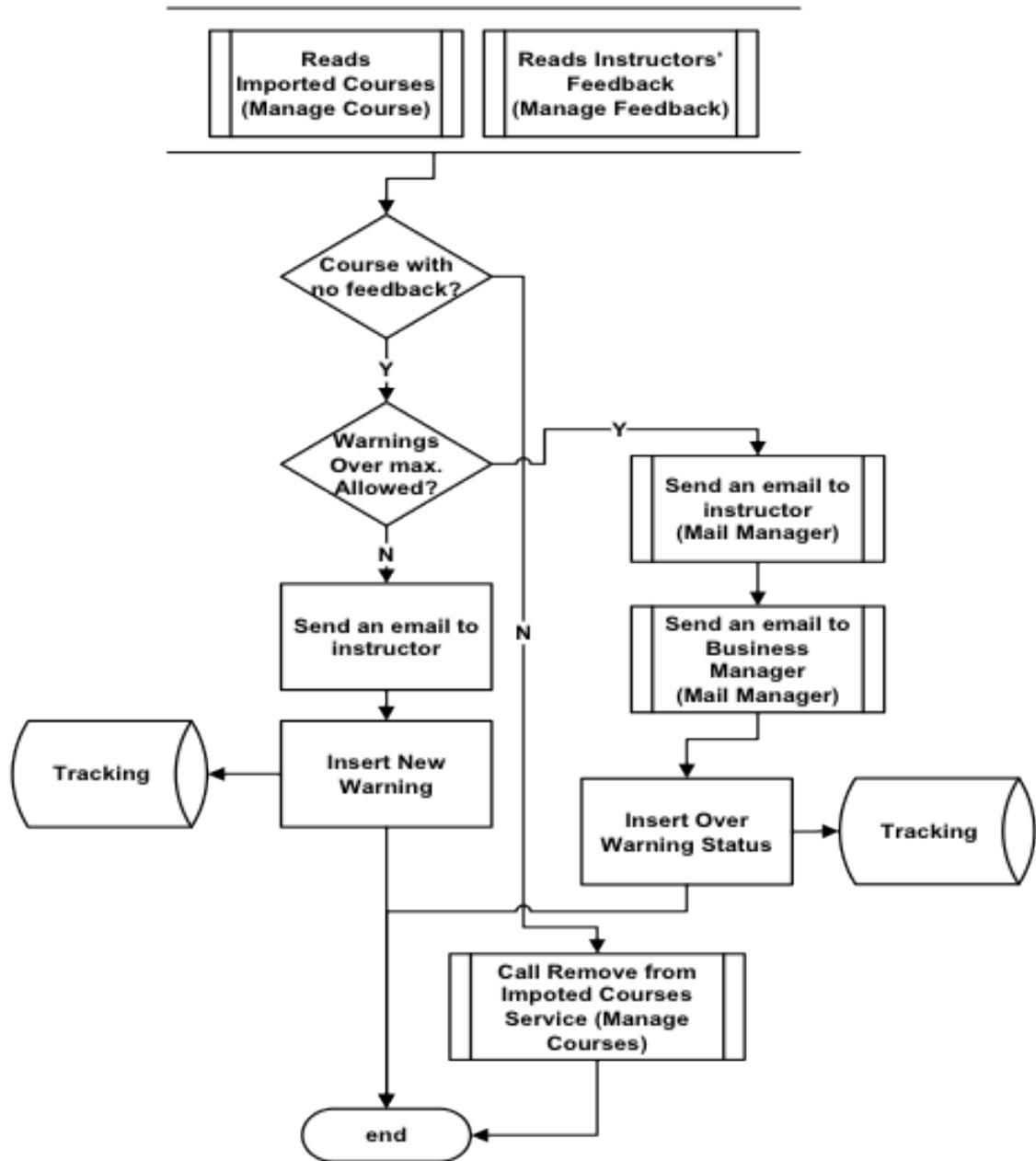


Figure 7.7: Imported Courses Tracker Process Analysis

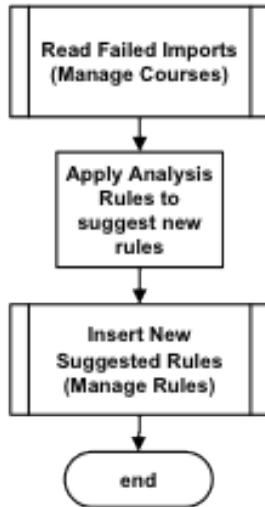


Figure 7.8: Analyzer Agent Process Analysis  
 Online: <http://zoom.it/GPed>

- Display Notification message: about rules, there should be a display rules page that is waiting for the dataset that will be passed by course manager.

Display service takes input from instructor and business manager. Example of instructor input is Course-ID. Example of Business Manager Input is Rules-Approved. Figure 7.6.1 presented in page 158 depicts detailed activities performed by Display service to support Search process.

### Check Capability Service

It determines either the course satisfies organizational rules or not, by doing so, it actually determines if course can be imported or not. Rules not only include financial issues; like limits provided for each instructor, but they also reflect pedagogical aspects of the educational institution. Educational institutions must guarantee certain pedagogical level of courses. Check Capability Service requires DB tables depicted in figure 7.11 presented in page 159.

### Import Service

It consumes other services to achieve the required task, which is download course file and update DB. In order to update DB, import service calls Manage Course appropriate services. Download Course It is the functionality that holds programming logic that is responsible for downloading the course file into FTP server.

### Manage Courses Service

Manage Courses refer to the capability to implement basic databases operations upon certain entity. Basic DB operations are: insert, update, and delete [29]. As

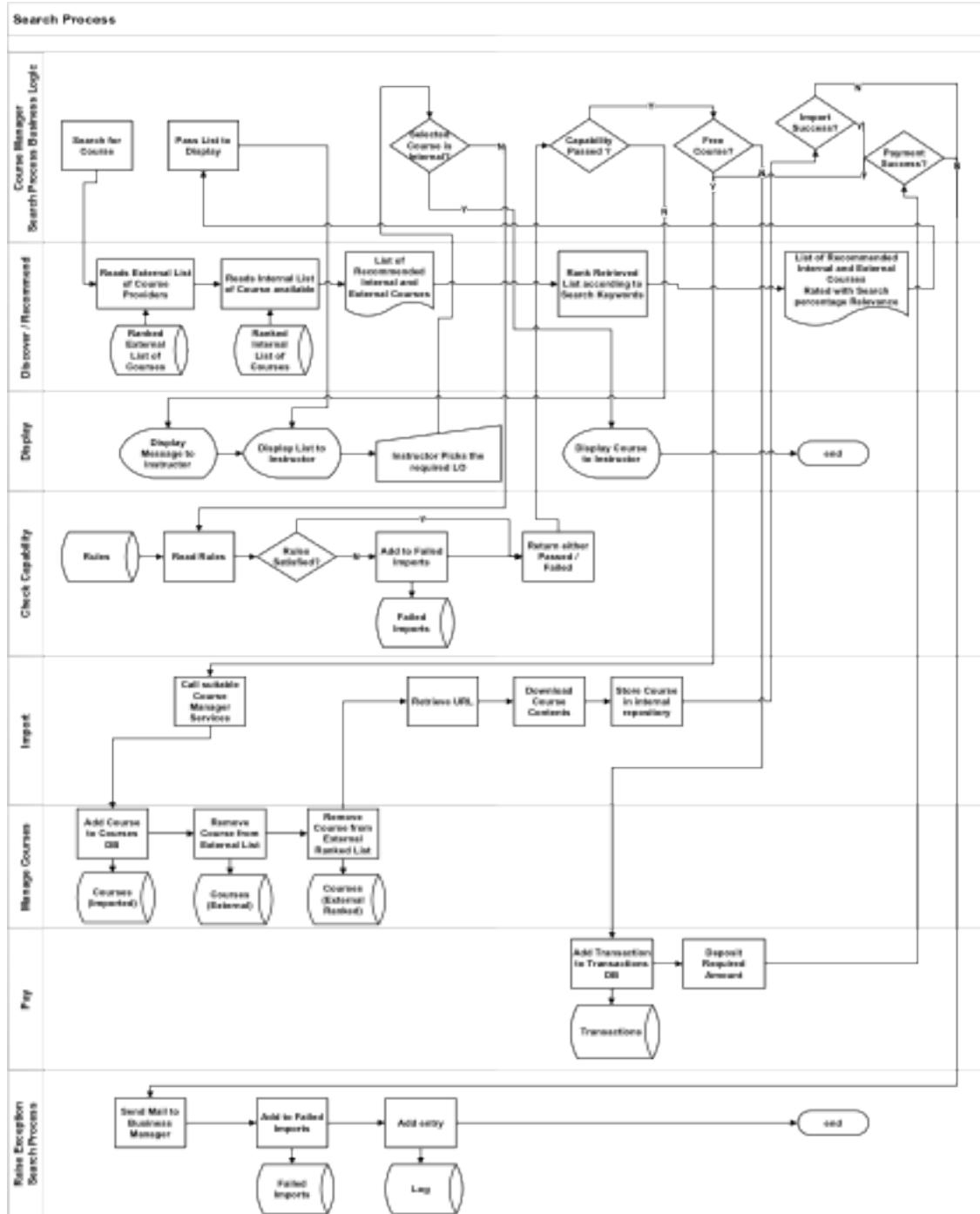


Figure 7.9: Search Process Design  
Online: <http://zoom.it/SCFK>

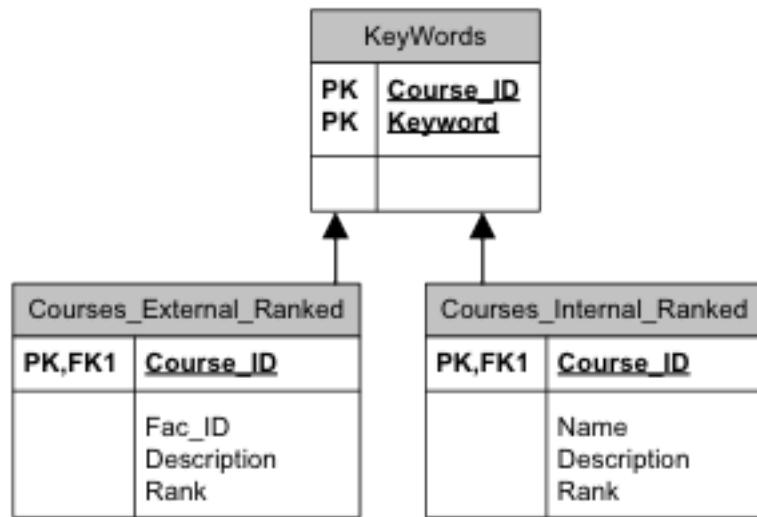


Figure 7.10: Tables of Discover/Recommend Service

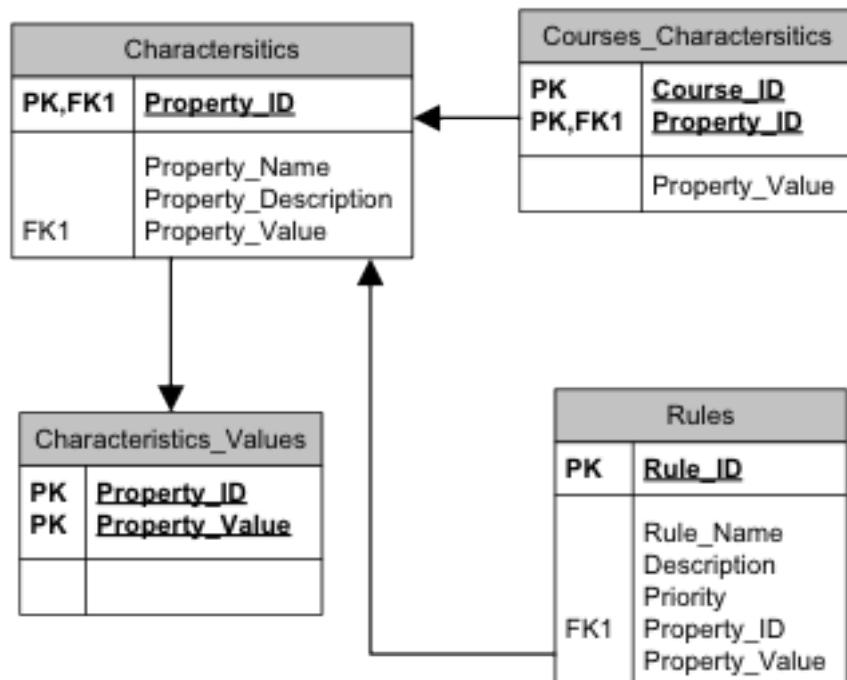


Figure 7.11: Tables of Check Capability Service

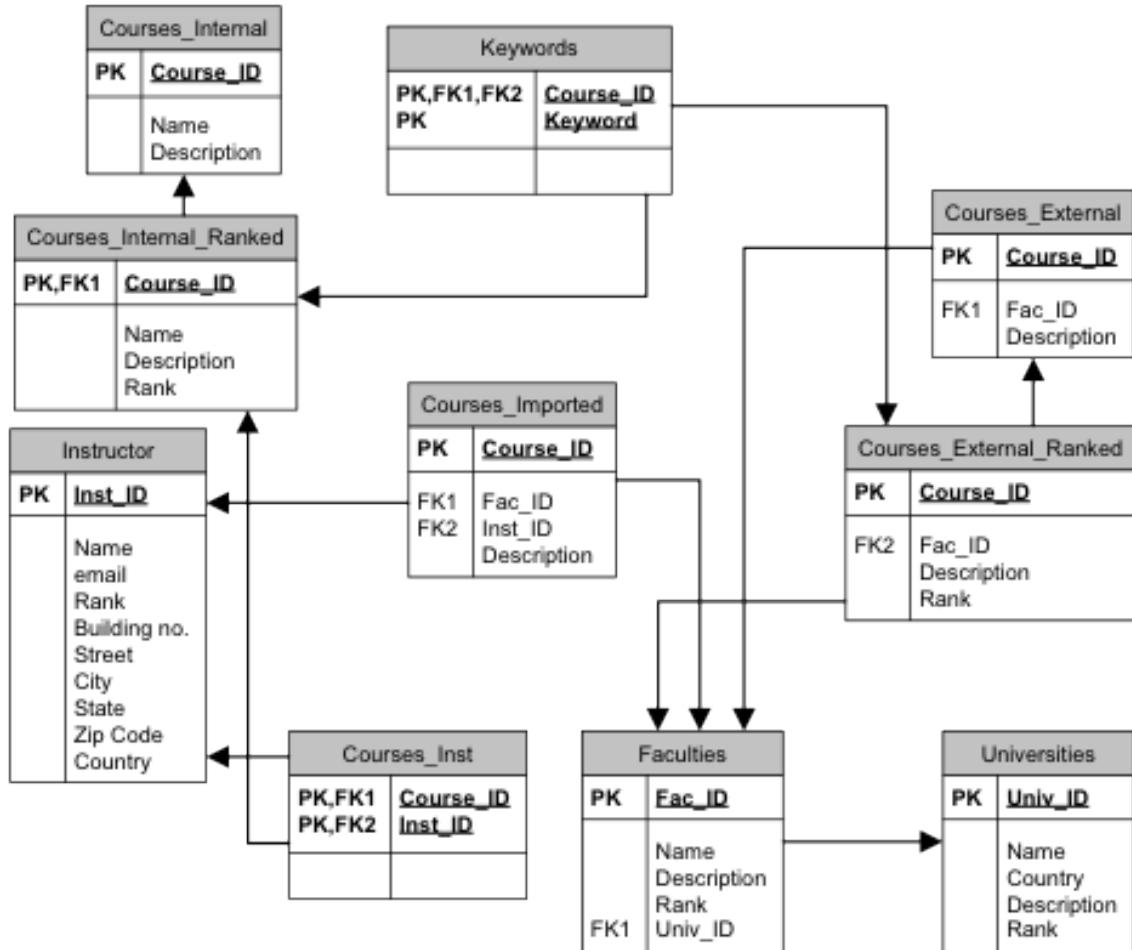


Figure 7.12: Tables of Manage Course Service  
 Online: <http://zoom.it/iOIHh>

courses are the entity of interest to search process, add, remove, and update operations should be supported. Courses are classified into three categories: Internal, External, and Imported. Internal courses refer to courses developed by institution's instructors. External courses refer to courses that are available in other educational institutions and have not been imported yet. Imported courses refer to downloaded courses form external sources. Internal and External courses are ranked in different tables. Manage Course Service is capable of performing basic DB operations on those different tables. Figure 7.6.1 presented in page 160 depicts required DB tables to support basic DB activities.

### Pay Service

Responsible for depositing money into accounts determined by the transaction. Figure 7.6.1 presented in page 158 depicts detailed activities by Pay service. Pay service should manage transactions carefully by adding transaction entry to transactions table before attempting the process for it to be able to rollback non successful transactions. Figure 7.13 presented in page 161 depicts required table to support financial

Transactions	
PK	<u>Transaction_ID</u>
	Inst_ID
	Account
	Amount
	Date
	Time

Figure 7.13: Table of Pay Service

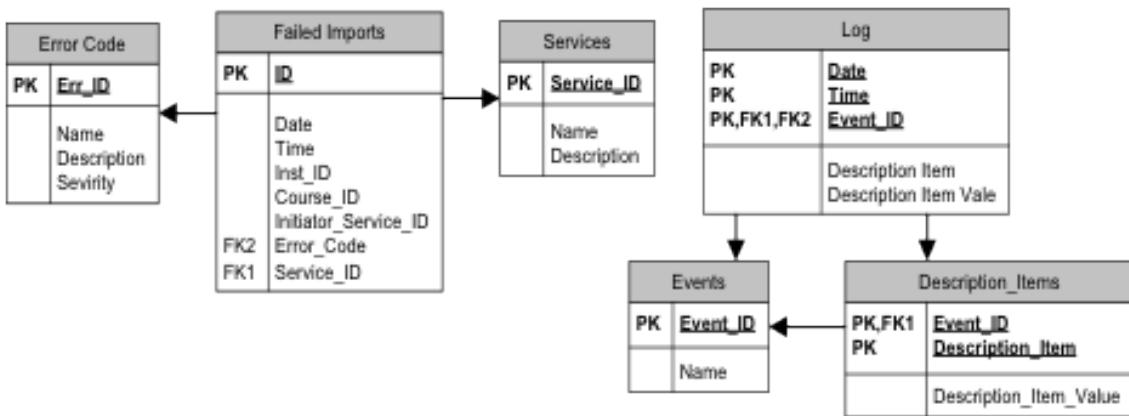


Figure 7.14: Tables of Raise Exception Search Process Service

transactions by the Pay service.

### Raise Exception Search Process Service

Incase any kind of errors happened during the Search process, a Search Process Exception is raised. Raise Exception Service performs two main functionalities: Adds an entry to Failed Imports table, and adds an entry to Log table. Figure 7.6.1 presented in page 158 depicts activities performed by Raise Exception Search Process Service. Figure 7.14 presented in page 161 depicts required tables by Raise Exception Search Process Service.

### 7.6.2 Manage Rules Process Design

Figure 7.6.2 presented in page 162 depicts Manage Rules Process Design. Manage rules process is initiated by Business Manager and managed by Rules Manager. Manage Rules Process utilizes the following services: Retrieve Suggested Rules, Display, Insert Rule, and Raise Exception.

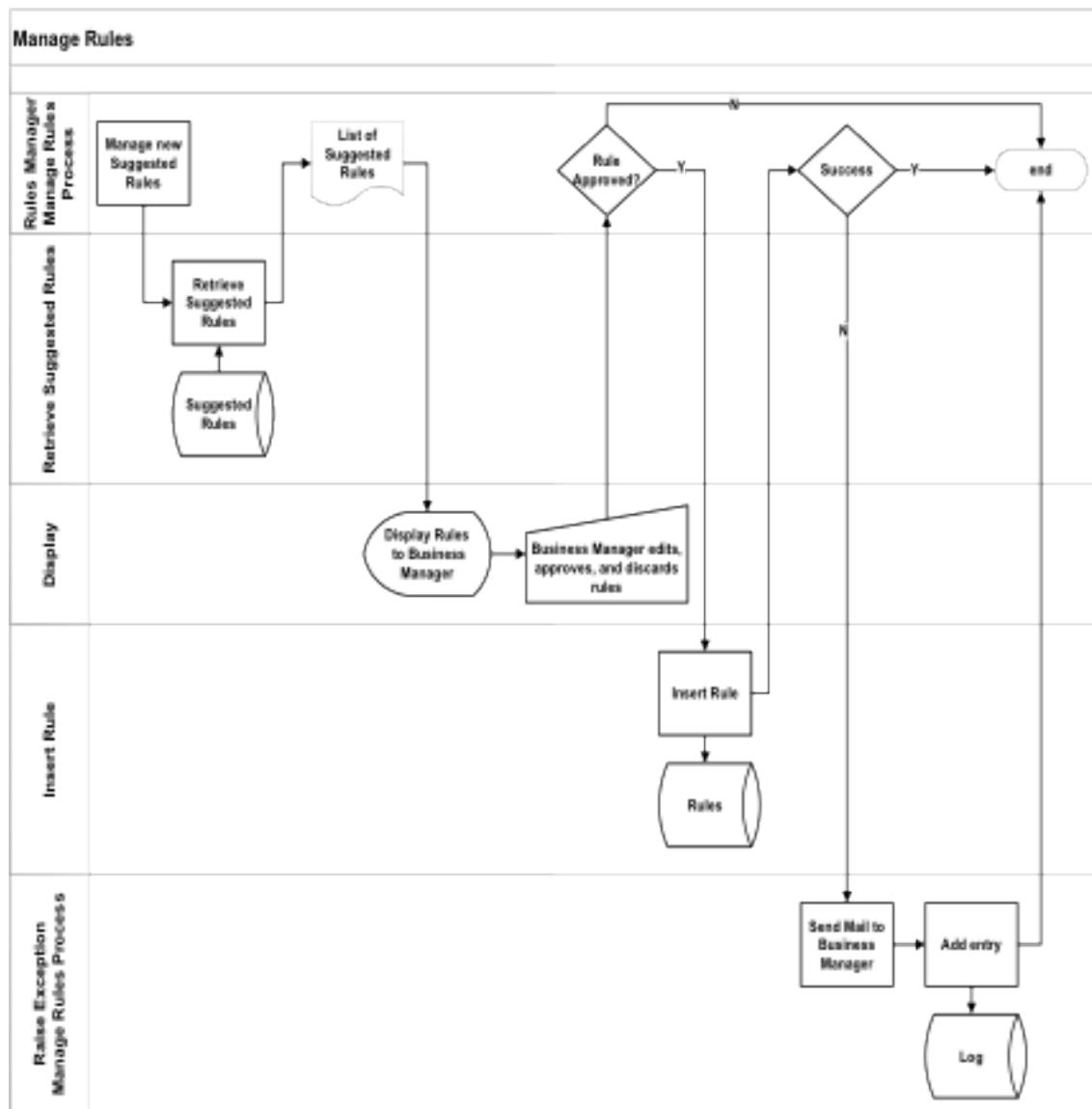


Figure 7.15: Manage Rules Process  
 Online: <http://zoom.it/3GqK>

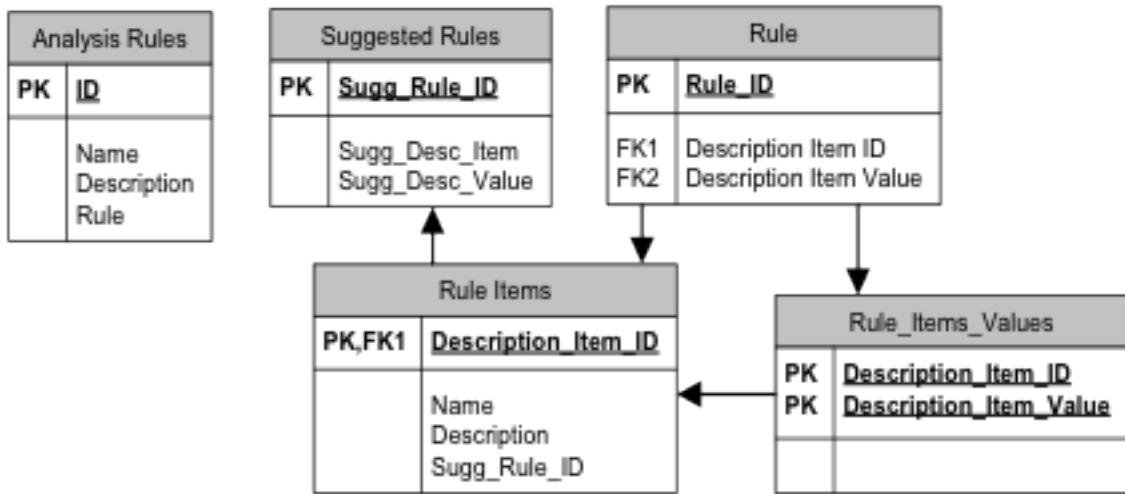


Figure 7.16: Tables of Manage Rules Process

### Manage Rules Service

Manage Rules Service supports basic DB operations on rules. There are three rule's categories: Rules, Analysis Rules, and Suggested Rules. Rules table presents rules of educational institution to rank courses and determining applicability of course import. Analysis rules table is the table that holds analysis rules will be used by Analyzer agent to apply them on failed imports, in order to generate suggested rules. Suggested rules table is the table that holds either new rules or edited current rules suggested by Analyzer agent and waiting for approval by Business Manager. Figure 4.23 depicts tables required by Manage Rules Service

### Raise Exception Manage Rules Service

Activities performed by Raise Exception Manage Rules Service are depicted within Figure 7.6.1 presented in page 158. Raise Exception Manage Rules Service requires only log table in order to add entry. Figure 7.14 presented in page 161 includes Log, Events, and Description-Items tables design.

### 7.6.3 Software Agents Design

This part will cover the design of two of the systems software agents. Those are: Tracker, and Ranker.

#### Tracker Agent

Proposed CMS depends heavily on instructor's feedback to enhance system's performance. In proposed system, tracking focuses mainly on instructors filling their feedback forms. Every instructor imported a course should give her/his feedback for this course. Imported courses feedback tracking requires consumption of three Web services. Figure 4.24 depicts tracking required tables.

Tracking	
PK	<u>Inst_ID</u>
PK	<u>Imported_Course_ID</u>
	Warning_Limit
	Current_Limit

Figure 7.17: Imported Courses Tracking Required Table

### Ranker Agent

Course Ranker consumes two services: Retreive Course, and Update Course. Pedagogical and managerial rules are presented with different priorities to reflect system's interest. Figure 7.18 presented in page 165 depicts ranking rules required tables. Ranking is either based on instructor's feedback, or mathematical formula. Courses, Instructors, and Faculties ranking is based on instructors' feedback for those different items. Several feedback forms with one to five grading are presented to instructors to evaluate different items. Rank items and inserted values are saved, and Ranker gets average of all those items to present final rank of any of the saved items. Thresholds are presented in different tables to enable Ranker to classify ranked item either accepted 'ranking value over threshold' or not accepted 'ranking value is below threshold'. Universities ranking is presented as a formula that is the average of all faculties composing this university ranks.

## 7.7 Summary

This chapter presented a proposed CMS that facilitates integration among different CMSs. An automated course search, import, and deposit process is presented. This automated process requires governance of business rules. Business rules might be limiting system efficiency, so they must be monitored and modified when needed. Manage business rules process was presented to achieve this goal. Utilizing SOA to integrate Web services and software agents in CMSs highlighted the unlimited advantages of Web services and its capabilities to facilitate software agents integration within systems. CMS should be thought of as a collection of stateless Web services. SOA provides a fine granularity and modularity that solves many integration problems, but adds complexity to systems design. SOA is a design pattern that helped enterprises overcome integration obstacles, and gain agile and interoperable advantages within architectures. Pedagogical, social, and managerial advantages of added processes include:

- Overcome lack of internal courses.
- Get use of external, higher pedagogical features courses.
- Shareability among different educational institutions.

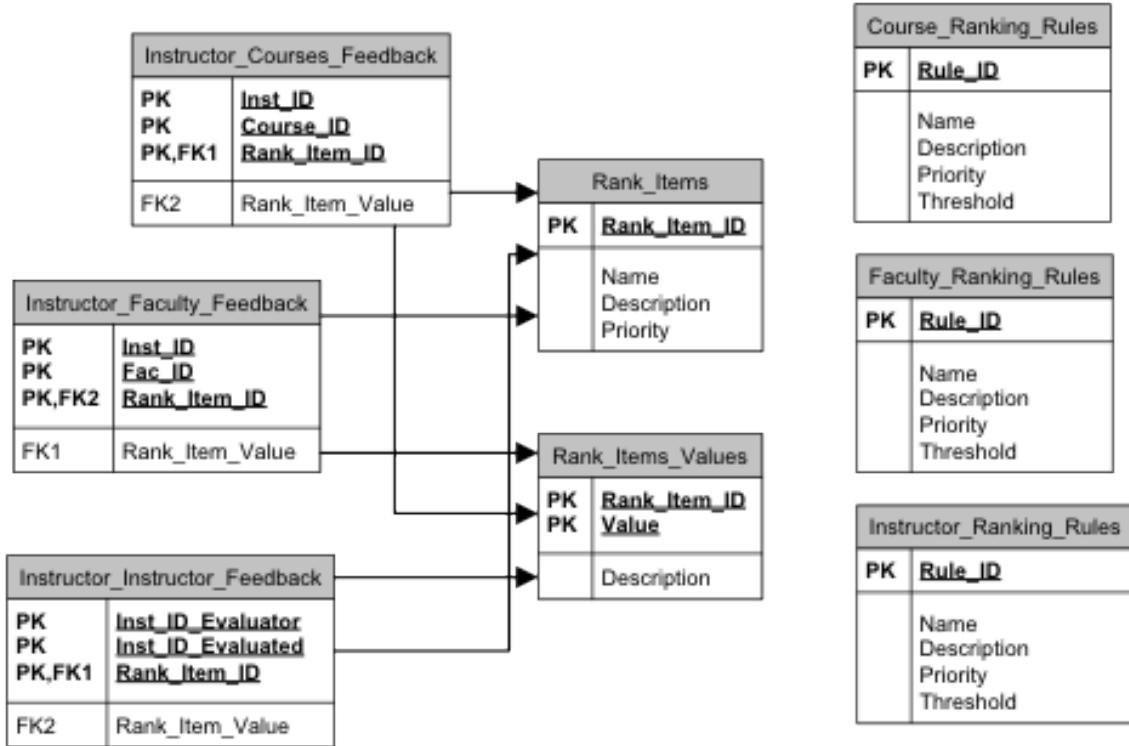


Figure 7.18: Tables of Ranker Software Agent

- Competition increment adds to quality (indirect effect).
- Increase Return-On-Investment by selling courses.

## 7.8 Review Questions

1. What are the main functionality of CMS?
2. List the main processes that must be available for CMS
3. Design a CMS that integrates both SW Agents and SOA.

## 7.9 Exercises and Labs

At this moment, you shall be already familiar with JAVA programming language, the basic building blocks of the language, and how to write a simple GUI application that uses JAVA main building blocks. Right now, you are requested to build an application that simulates some of the functionality of CMS presented at this chapter.



# **Chapter 8**

## **Mobile Devices in LMS**

### **8.1 Introduction**

Assessment is one of the learning activities that can be achieved electronically and via mobile devices. Assessment Management System (AMS) is part of LMS that is responsible for providing, managing, automating, and facilitating the assessment process. Mobile assessment adoption in current LMS requires architectural modifications to reflect interoperability characteristics required by mobile supported processes. Mobile Learning (M-Learning) is an approach to e-Learning that utilizes mobile devices and should be enabled by educational institution LMS. Unfortunately, M-Learning is not the same at all educational institutions. Mobile assessment refers to the capability of conducting assessments via mobile devices. Mobile assessment relies on external services that are not part of the LMS.

### **8.2 AMS Analysis**

Mobile assessment utilizes mobile services architecture to deliver interactive messaging automatically. Interactive messaging include sending assessment questions and receiving multiple responses SMSs. Figure 8.1 presented in page 168 presents Mobile Services Architecture required to enable learners to interact via mobile SMS with LMS that resides in university server. Mobile services architecture can have different implementations. Understanding the mobile service architecture is critical for the design and implementation of LMS.

Learner is connected to her/his mobile service provider via cell or smart phones. Mobile operators implement one or more SMS centers. Those are centers that manage sending and receiving of SMSs. University needs a mediator in the way to SMS center. Mediator lies in the middle between mobile operators and LMS. Mediator connects directly to different SMS centers using Short-Message Peer-to-Peer (SMPP) protocol. SMPP is a telecommunication industry protocol for exchanging SMS messages between SMS peer entities such as SMS center. It is often used to allow third parties to submit messages in bulk. SMPP has been designed to offer services for various cellular networks such as GSM, CDMA, and TDMA. Mediator is connected directly to the University server over the Internet, via standard Web services. Mediator receives and sends SMSs and exposes two Web service that give LMS the capabil-

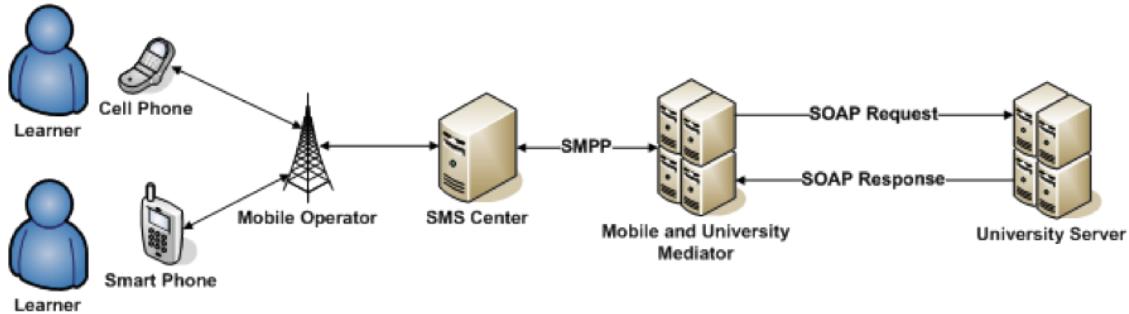


Figure 8.1: Mobile Services Architecture

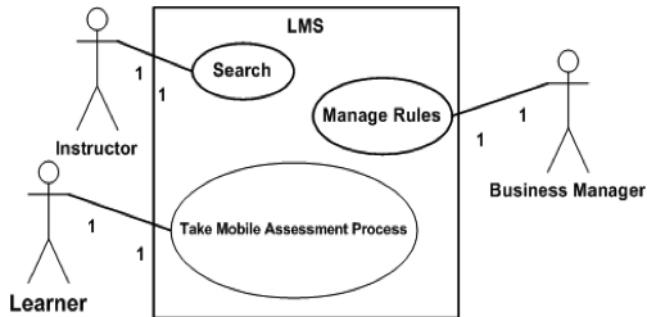


Figure 8.2: AMS Use Case

ity to access SMSs received, and to send SMSs. Via SOAP request, university can receive SMSs aimed to it, and via SOAP response, university can send new SMSs. University LMS should manage sessions with different users, utilizes data within SMSs in managing learners profiles. LMS shall maintain a record of all sent and received SMSs for managerial, financial, and educational issues. Proposed extended LMS addresses a new added process namely Take Mobile Assessment as depicted in the use case diagram at figure 8.2 presented in page 168. Figure 8.3 presented in page169 presents AMS entities.

### 8.2.1 Proposed AMS Architecture

Proposed architecture as presented in figure 8.2.1 presented in page 170 consists of two layers: Interface layer, and Service layer. Interface layer interacts with instructors, learners, and business managers via human interface, that is portals, and with external organization services via machine interface, that is Web services. Service layer contains core LMS services and has three sub layers: Orchestration, Application Services, and Agents layer. Orchestration layer holds business logic presented by system processes as executable services. Business logic refers to different activities that can include Web services invocations, data manipulation, exception handling, and process termination. Application Services layer contains set of stateless Web services that are capable of performing certain tasks related to system entities. System entities reflect distinguishable objects within system that should be reflected

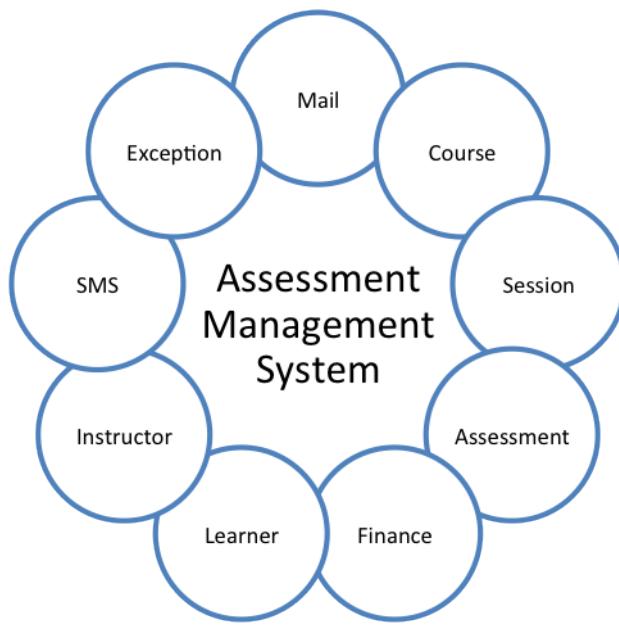


Figure 8.3: AMS Entities

in the LMS, like instructors, courses, and assessments. Web services are stateless because they can not maintain business logic, operation flow, or user state. Agents layer presents the suggested required software agents to serve the overall system. Agents layer presents Tracker software agent that keep track of learner assessments due dates.

### 8.2.2 Take Assessment Process Analysis

Figure 8.5 presented in page 171 presents Take Mobile Assessment process analysis. Take Mobile Assessment process is initiated by learner, managed and maintained by Assessment Manager, and Session Manager. Assessment Manager and Session Manager are both Orchestration layer services. Managing and maintaining take mobile assessment process refers to keeping track of assessment items retrieved, validating learner's answers, handling exceptions, and maintaining actions sequence and Web services invocations. Figure 8.5 presented in page 171 implicitly defines LMS boundary for take mobile assessment process. Out of LMS scope refers to activities fall outside LMS boundary and can not be maintained, neither controlled by LMS. Activities related to mobile operators and mediators are out of LMS boundary. LMS scope activities are handled, maintained, and controlled by LMS. Take Mobile Assessment process begins for LMS once it retrieves SMS that is aimed for assessment. Take Mobile Assessment process is an example of processes that contain activities outside organizational boundaries.

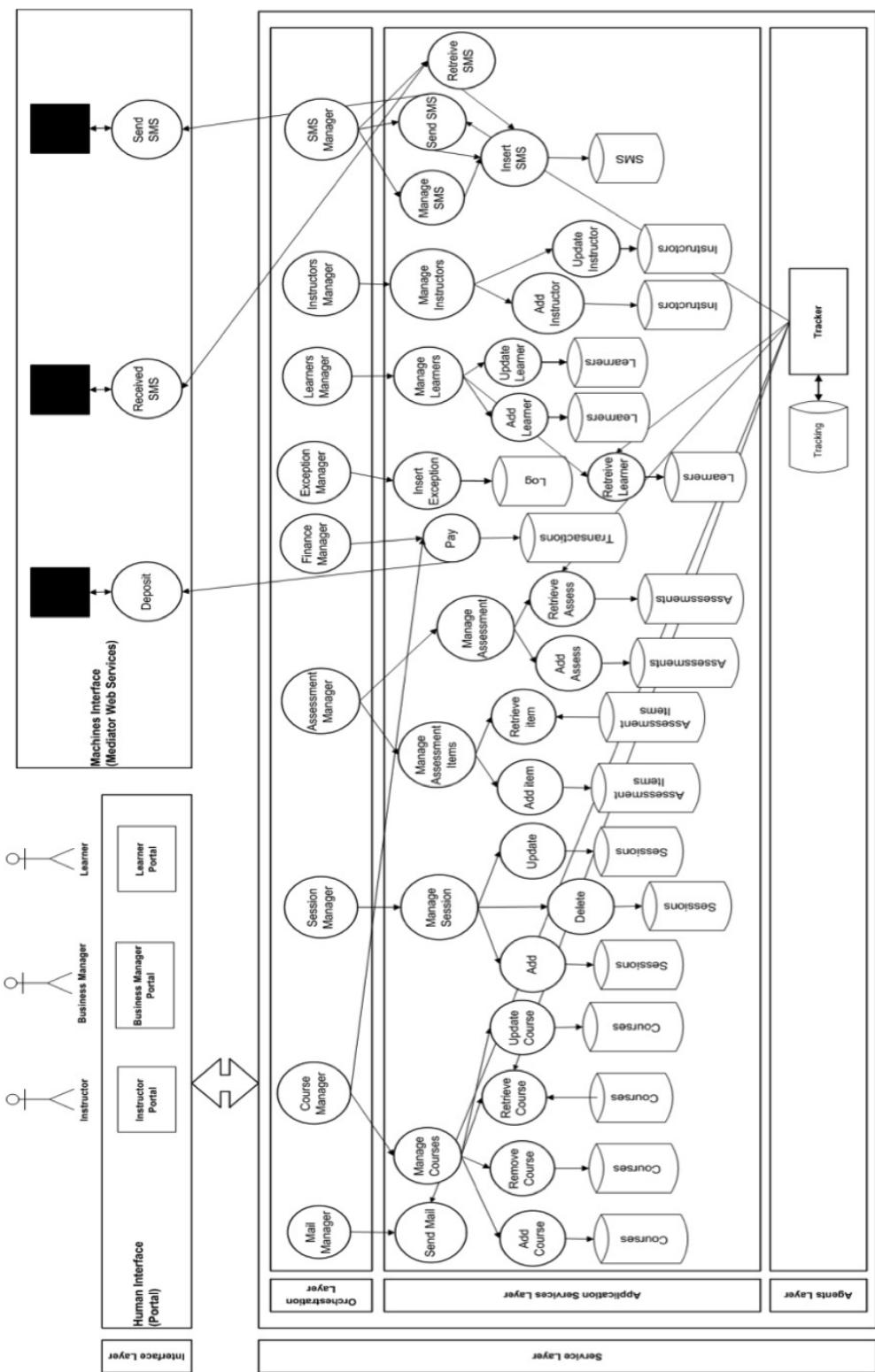


Figure 8.4: Proposed AMS Architecture  
Online: <http://zoom.it/jiNG>

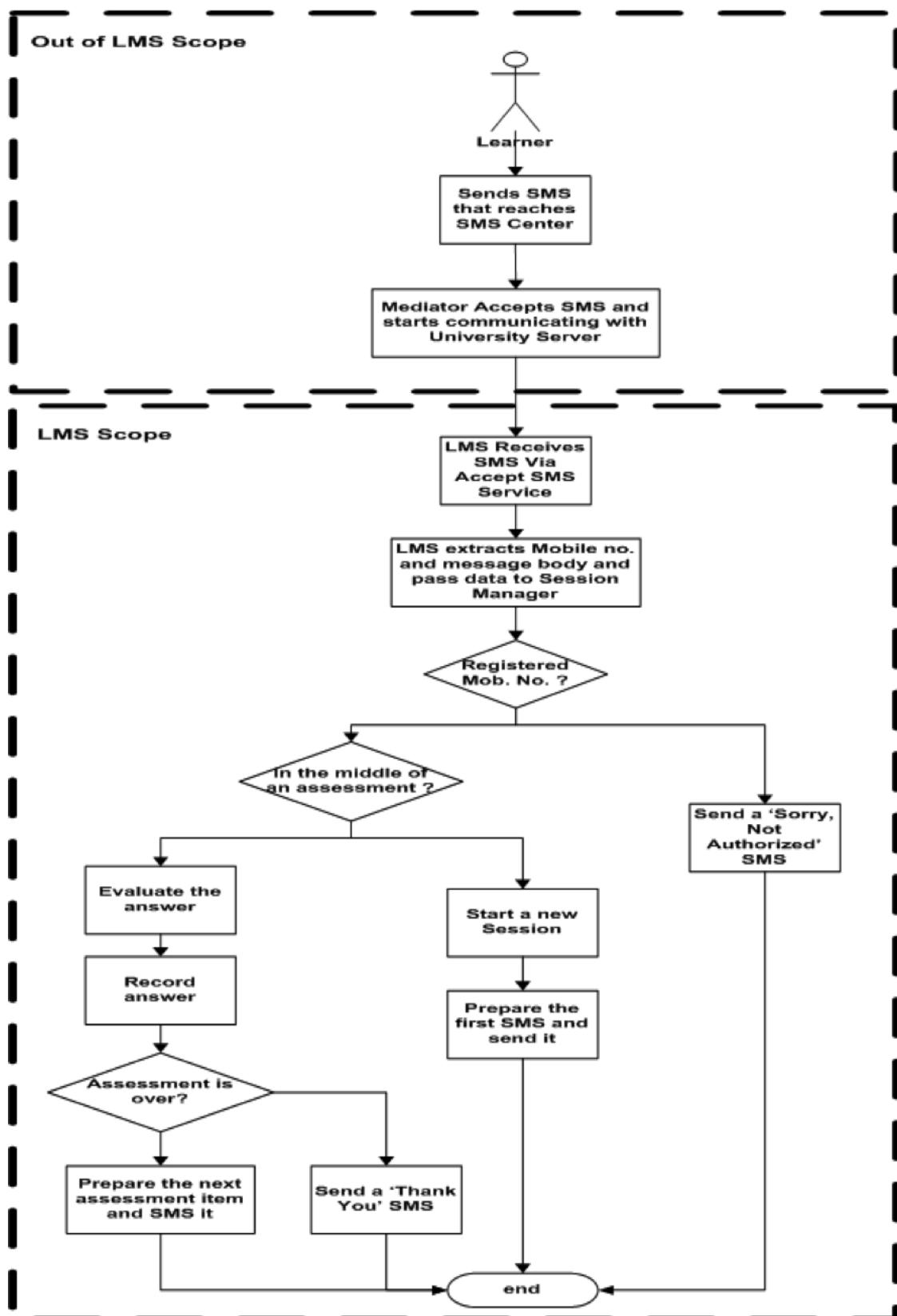


Figure 8.5: Take Mobile Assessment Analysis

### 8.2.3 Tracker Agent

The role of software agents in tracking users' activities has been widely known and accepted. Figure 8.6 presented in page 173 illustrates analysis of tracking process. Tracking process is initiated and performed by tracker agent and aims to track learners missing assessments and take some action to remind students with due dates of assessments. Tracking process consumes five Web services: Read Learner Data, Read Course Data, Read Assessment Data, Send Mail, and Send SMS.

## 8.3 AMS Design

AMS design task includes automating the new added process, and designing the required software agent. Automating take mobile assessment process includes partitioning the process into reusable services, designing services, and designing DB tables required to support each service.

### 8.3.1 Take Mobile Assessment Process Design

Figure 8.3.1 presented in page 174 presents Take Mobile Assessment Process design. Take Mobile Assessment Process is managed and maintained by assessment manager, and session manager. Take mobile assessment process utilizes Web services namely: Manage SMS, Manage Learner, Session Manager, Manage Assessment Items, Manage Session, and Insert Exception. LMS services can be reused to serve different processes. Design of Manage SMS, Manage Learner, Manage Assessments, Manage Assessment Items, Manage Sessions, and Session Manager Services are presented in details.

#### Manage SMS Service Design

Manage SMS presents a collection of services that include Retrieve SMS, Send SMS, Insert SMS, and Delete SMS. Retrieve and Send SMS are two services that map mediator exposed Web services. LMS maps exposed mediator's Send and Receive SMSs Web services into internal Retrieve and Send SMS. When LMS invokes both services, it is actually invoking Mediator's Send and Receive SMS services. This mapping is tended to act as an isolation layer to enable changing mediator upon need. Insert SMS keeps a log of all received and sent SMSs. LMS maintains a record of all sent and received SMSs for managerial, financial, and educational issues. Records can be used to calculate fees to be paid for mediator monthly according to usage statistics. Figure 8.8 presented in page 175 shows the required DB table for Insert SMS service that is invoked every time a SMS is sent or received. Delete SMS utilizes the same DB table and is invoked periodically to clear SMS record from too old SMSs based on date and time basis.

#### Manage Learner Service Design

Manage Learner encapsulates the three primary DB operations insert, update, and delete. Learners' data and profiles are maintained and updated based on attended



Figure 8.6: Tracking Process Analysis

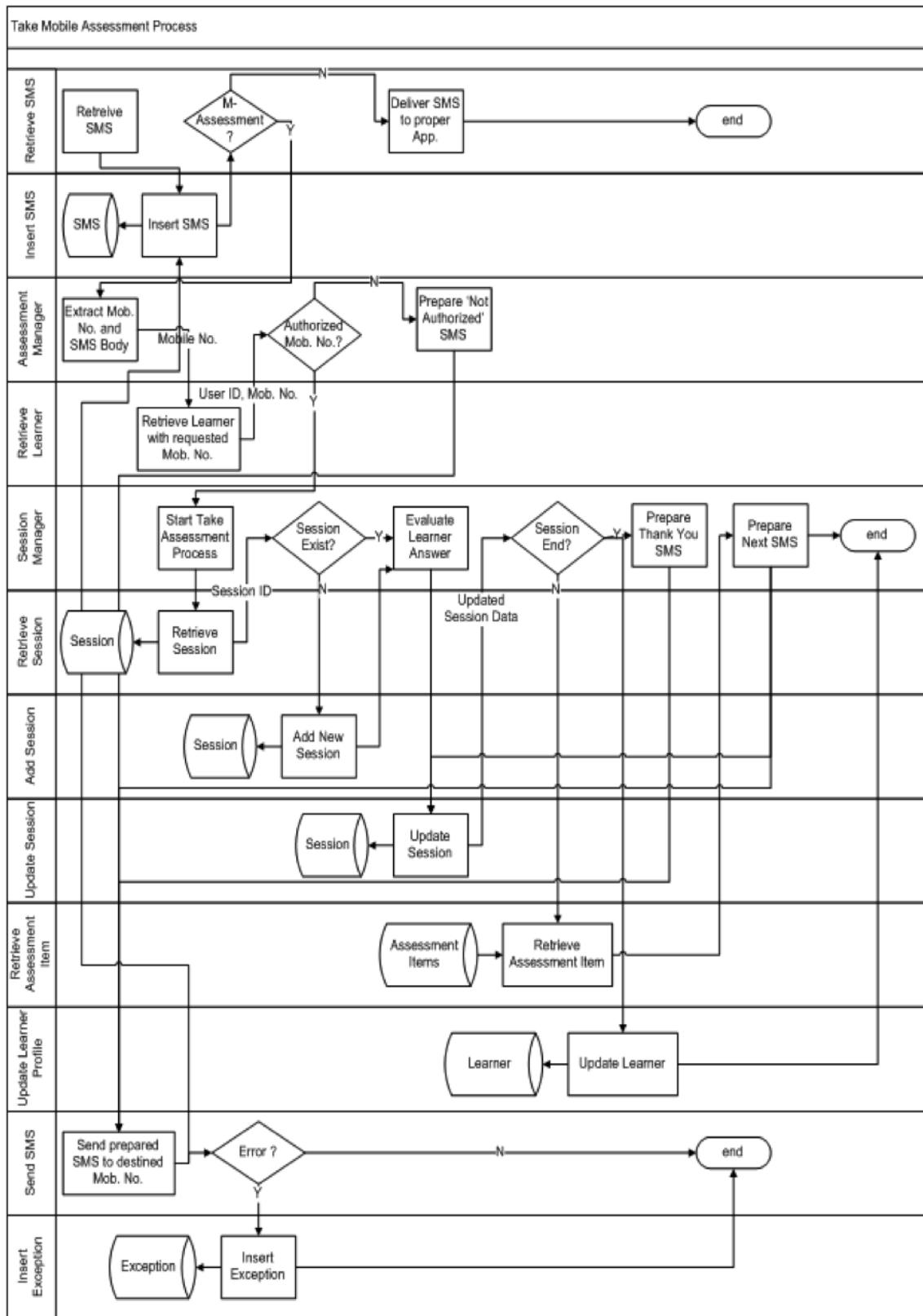


Figure 8.7: Take Mobile Assessment Process Design  
Online: <http://zoom.it/gdPO>

SMS	
PK	<u>Mob. No.</u>
PK	<u>Date</u>
PK	<u>Time</u>
Content	

Figure 8.8: Table of Insert SMS Service

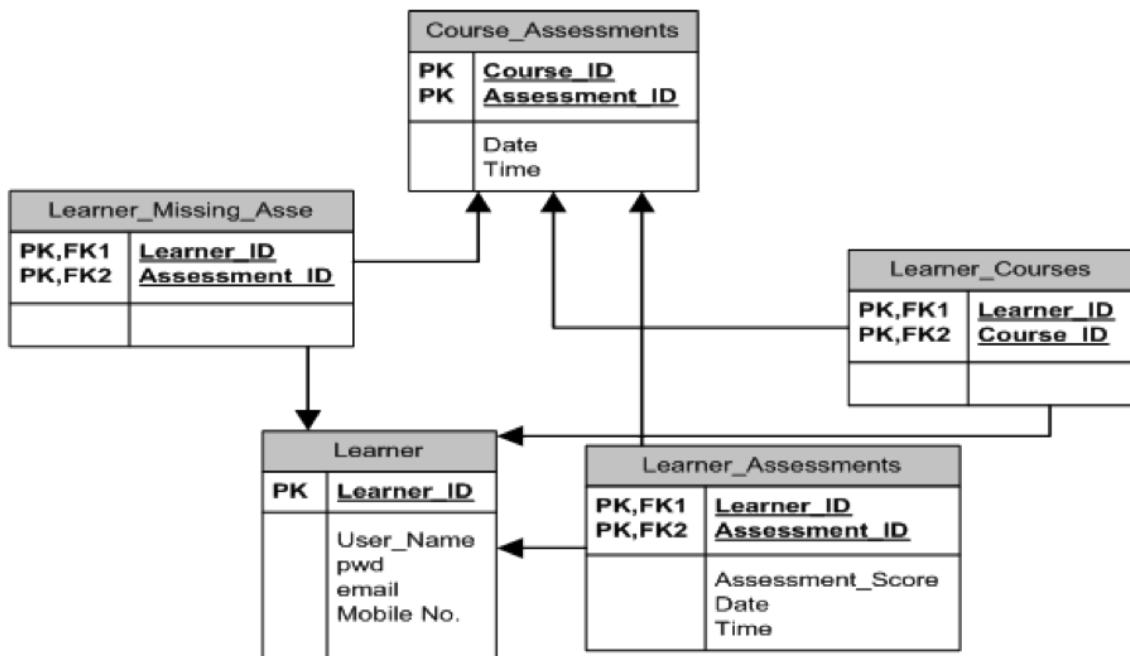


Figure 8.9: Tables of Manage Learner Services

assessments. Figure 8.9 presented in page 175 presents required DB tables to maintain learners' data and profile. Course-Assessments table holds assessments of each course. Learner-Courses table holds courses attended by each learner. Learner-Assessments table holds assessments attended by learner, and related data like student score for this assessment, date, and time student attended the assessment. Assessment dates are stored, so LMS can determine learners missing assessments to be inserted in missing-assessments table. Learner-Missing-Asse table holds assessments that learners should have attended but they did not. Learner-Missing-Asse is used by tracker agent to remind students of missing assessments.

### Manage Assessments Service Design

Figure 8.10 presented in page 176 shows required DB tables to store and manage assessments. Manage assessments service is concerned with the three main DB operations insert, update, and delete assessments themselves, not assessment items.

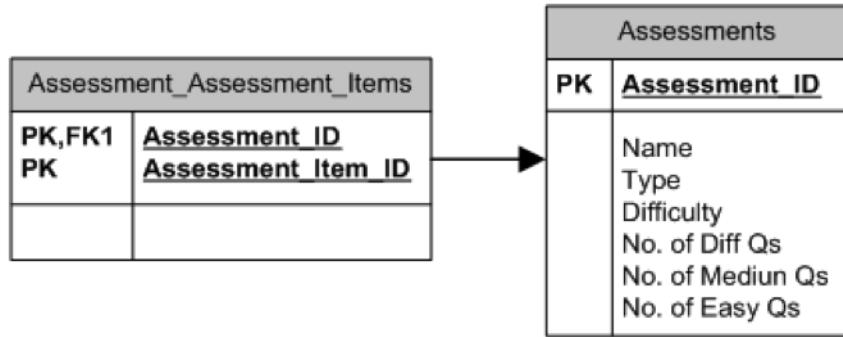


Figure 8.10: Tables of Manage Assessments Services

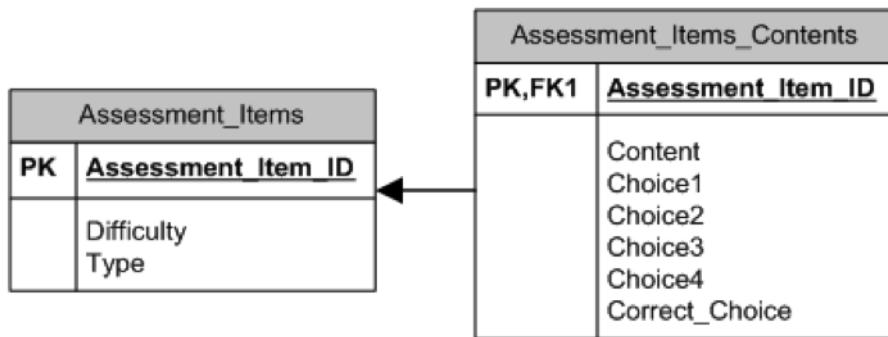


Figure 8.11: Manage Assessment Items Services

Assessments should contain variant difficulty levels of assessment items. Assessments are categorized according to difficulty level into easy, medium, and difficult. Instructor is responsible for categorizing difficulty level of each assessment from educational perspective. An automated assessment categorization can be achieved via calculating percentage of difficult, medium, and easy assessment items composing the assessment.

### Manage Assessment Items Service Design

Assessment items are the main factor of assessments to either achieve required goals or not. Assessment items need to be well prepared to enable assessment efficiency. Assessment items are stored and managed separately to form an assessment items repository. Assessment items repository enable sharing assessment items among different assessments to overcome interference of topics among courses and to enrich assessment items bank. Sharing assessment items among assessments is done under instructors' supervision. MCQs, True/False, and Complete the missing word are examples of applicable mobile assessment items. From assessment items difficulty point of view, LMS has three classes: easy, medium, and hard. Assessment item difficulty level is determined by instructor either while inserting or updating assessment item. Figure 8.11 presented in page 176 depicts the required DB tables for manage assessment items service.

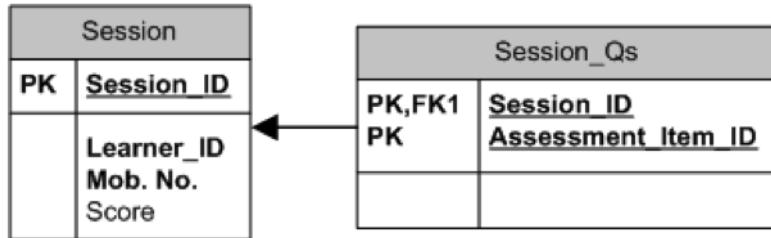


Figure 8.12: Tables of Manage Session Services

### Session Manager Service Design

Session refers to the period of time in which the same user interacts with the system. Mobile user interacts with the system via multiple discrete responses. Web services are stateless, and sessions are needed to enable stateless protocols to track user interaction with the system. In LMS, Session refers to the period of time the learner is identified by the system as in the middle of a process. Session managers maintain users' interactivity data temporarily in the DB. Session Manager Service holds the business logic required to enable the supported processes. Temporary DB accesses are left to manage session service. Session manager activities include recognizing either the learner is in the middle of an assessment or attempting to a new one, and then invoking the suitable Web services based on user state. Some of the Web services session manager invokes are add session, update session, update learner profile, and send SMS.

### Manage Session Service Design

Manage session service is responsible for performing the three main DB operations insert, update, and delete sessions. Figure 8.12 presented in page 177 shows the required tables for manage session services. Session data include a unique session ID that becomes available for reuse when session ends, Learner ID that reflects the learner unique ID stored for each learner, learner mobile number, and score of the learner in the assessment. Score increases when learner correctly answer assessment item. Another DB table is required to keep track of assessment items sent to the learner during the session, so sending of repeated assessment items is avoided.

### Manage Exception

Exceptions refer to non-ordinary flow of events or actions. Well defined ISs should address and handle exceptions carefully. Feed forward ISs needs to keep record of exceptions took place to be analyzed. Software agents can perform analysis functions on recorded exceptions. Figure 8.13 presented in page 178 depicts the DB table required to keep exception log. Exception log is available for business manager and system administrator to maintain system state. Figure 8.3.1 presented in page 178 shows the overall implemented Assessment Management System DB tables.

Exception	
PK	ID
	Description Date Time Severity

Figure 8.13: Table of Insert Exception Service

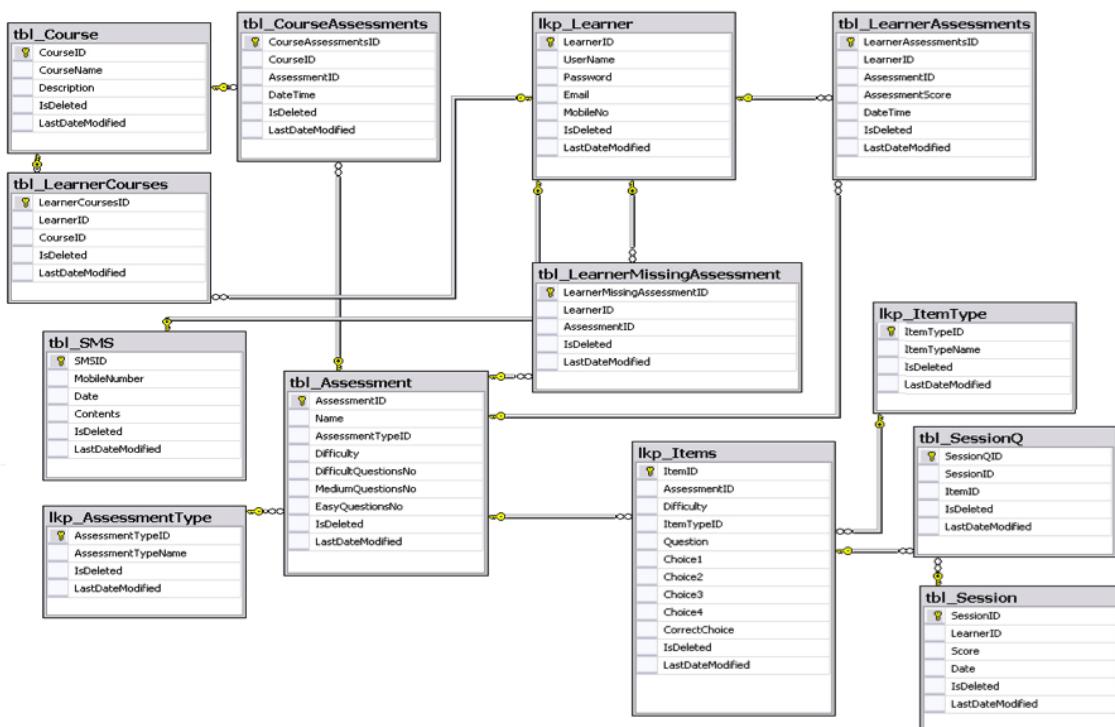


Figure 8.14: Overview of AMS Database Tables  
Online: <http://zoom.it/whne>

Tracking	
PK	<u>Learner ID</u>
PK	<u>Assessment ID</u>
	Warning Limit Current Limit

Figure 8.15: Table of Tracker Agent

### 8.3.2 Tracker Agent

Tracking process consumes five Web services: Read Learner Data, Read Course Data, Read Assessment Data, Send Mail, and Send SMS. Figure 8.15 presented in page 179 presents the DB table required for tracker to keep track of current submitted learners warnings, and the maximum number of allowed warnings. Tracker agent gets information about the learner missing assessments by utilizing the manage learner.

### 8.3.3 AMS Classes

Figure 8.16 presented in page 180 presents Services layer SMS Manager class diagram. Figure 8.3.3 presented in page 181 depicts Assessment Management System class diagram,

### 8.3.4 Mobile Simulator Interface Design

Figure 8.18 presented in page 181 shows the mobile simulator interface available via the portal to test the AMS mobile functionality.

## 8.4 Learner interaction with LMS

Take Mobile Assessment process relies on the sequence of SMSs learner sends and receives with the LMS. Figure 8.19 presented in page 182 shows a classical interaction between learner and Assessment Management System during the take mobile assessment process. Highlighted shapes present SMSs sent by learner, and other shapes present SMSs received. Learner sends SMS with assessment ID, receives the first assessment item, responds with the appropriate choice, and finally receives a summary with the result. The receive assessment item and response sent by learner operation is repeated as many times as assessment items for each assessment.

## 8.5 Summary

Assessment is an integral part of the learning process, and a learning activity that can be achieved efficiently via mobile devices. Due to differences in mobile archi-

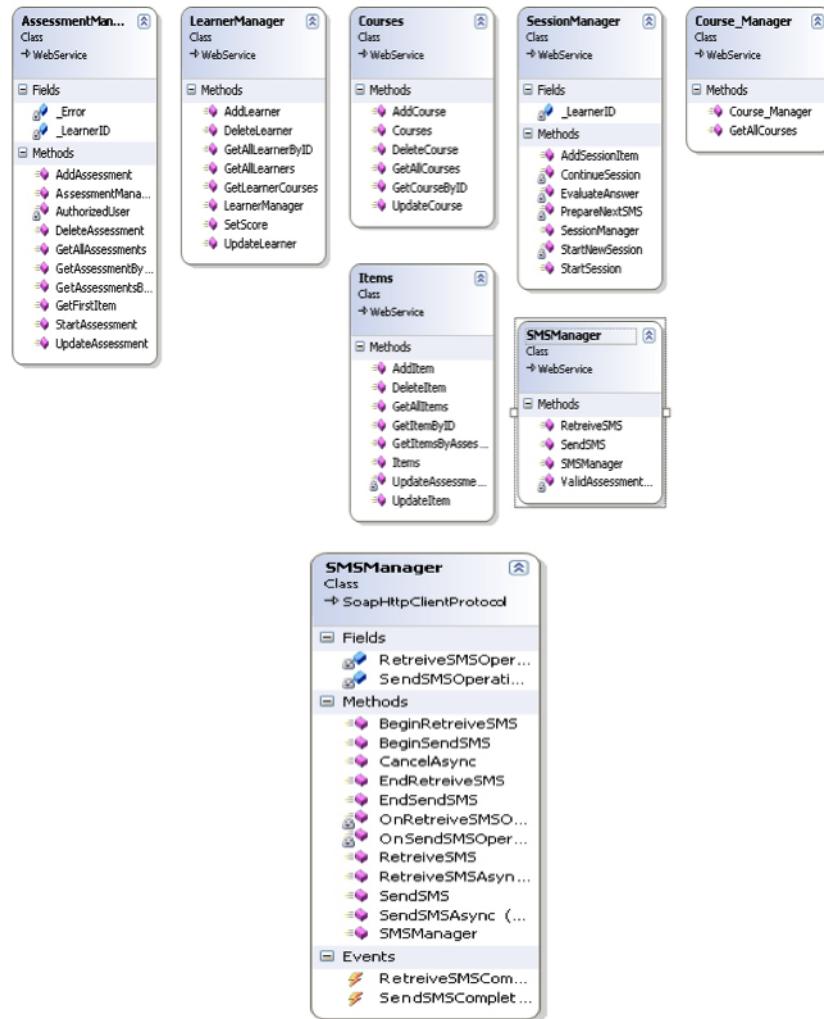


Figure 8.16: Service Layer and SMS Manager Class Diagram

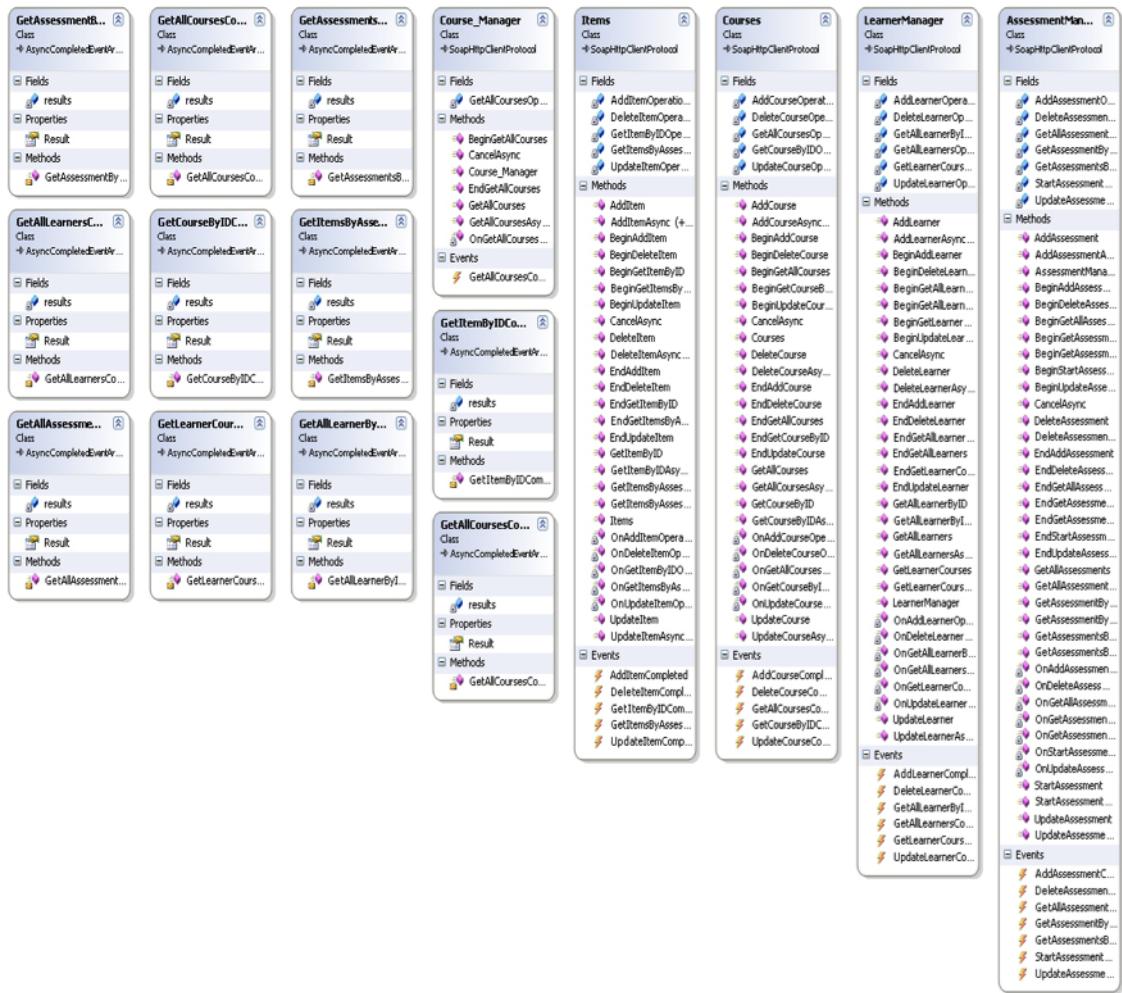


Figure 8.17: AMS Class Diagram  
Online: <http://zoom.it/PbTm>



Figure 8.18: Mobile Simulator

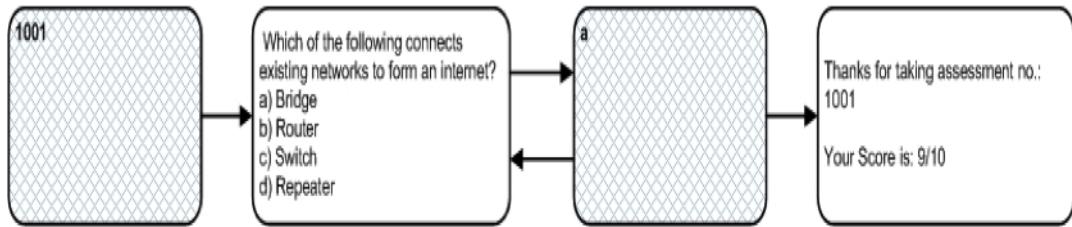


Figure 8.19: Classical Learner – AMS Mobile Interaction

lectures, and as a result of lack of interoperability in current commercial LMS with external systems, mobile assessment may find difficulties in implementation. SOA as a design pattern addresses non functional requirements like interoperability and integration to achieve systems agility. SOA utilizes Web services as main enabler to achieve addressed design goals. Interoperability is a main challenge for systems adopting and interacting with external components that. Interoperability between LMS and mobile mediator is achieved via adoption of Web services based SOA in LMS. SOA provides a fine granularity and modularity via standard interface that solves integration and interoperability challenges, but adds complexity to systems design. Pedagogically, mobile assessment is important due to the assessment importance process to the learning process. Mobile assessment facilitated and encouraged learners to attend assessments and enabled distance education by expanding interactivity tools available to learners to include mobile devices.

## 8.6 Review Questions

1. How can we use Mobile devices in e-Learning?
2. What are the main challenges of integrating Mobile Devices in e-Learning?
3. Design a SOA based LMS that integrates Mobile Devices.

## 8.7 Exercises and Labs

At this moment, you shall be already familiar with JAVA programming language, the basic building blocks of the language, and how to write a simple GUI application that uses JAVA main building blocks. Right now, you are requested to build an application that simulates some of the functionalities of M-Learning presented at this chapter.

# Chapter 9

## Web 2.0 in LMS

### 9.1 Introduction

Using ICT in learning has a long history and was handled in different ways and from different points of views over the years. Though most of the time this diversity has led to the most inappropriate understandings and misleading implementations of the concepts, it has also opened the way for different approaches that has been under validation and evaluation over the years. The learning institution in concern for this chapter is the University. The focus of this section will be on ISs that were presented over the years to facilitate university activities. Then, the attempts of different learning institutions to utilize web 2.0 technologies will be presented. Current widely spread technologies in learning institutions include utilizing ICT in the learning process mainly for two categories: University Management Information Systems (UMISs), and LMSs (LMSs). El-Ghareeb (2009) compares systems' components, scopes, interest, usage, features, and utilization in universities yielding that [112]:

- UMIS: used in handling and managing managerial aspects of the university. They have almost nothing to do with the pedagogical process. However, their existence is almost a must. UMIS exceeds LMS by decades and has reached stable levels of well-defined requirements, existence, acceptance, implementation and performance levels. UMIS include different sub ISs that work to facilitate and manage the activities of different parts of the university as an organization. Examples of those sub ISs are: Student Information System (SIS), Library Information System, Faculty Information System and Finance System.
- LMS: used in handling and managing the learning process. LMS implements technologies that enable virtual/digital university and/or personal learning environments. LMS features can be categorized into four sub systems concerned with courses, exams, assessments and collaborative features. LMS can be thought of as the integration of these four sub systems. Each one presents specific functionalities via specific tools. Providing online lectures via video-conferencing and recording capabilities used to be optional in LMSs.

Both UMIS and LMS have to integrate and operate together to support educational institutions and e-Learning. One of the international initiatives that provide

information to institutions on investing in and using information technology infrastructure is e-Framework “<http://www.e-framework.org>”. The e-Framework for education and research has been established to help the education and research arenas take advantage of the opportunities offered by the service-oriented approach. The primary goal of the e-Framework is to facilitate technical interoperability within and across education and research through improved strategic planning and implementation processes. The e-Framework is a collaborative effort that recognizes greater coherence in development is needed and thus aims to provide an overview of current development and experiences in services-oriented approaches.

This chapter goes as follows: section 2 presents web 2.0 technologies, examples of current available technologies, statistics about their utilization, and how they changed the internet perception for users. Section 3 highlights the transition from e-Learning to e-Learning 2.0 via utilizing different web 2.0 technologies. Section 4 reviews evaluation of web 2.0 utilization in learning institutions from theory and practice. Section 5 presents real world examples of utilizing web 2.0 technologies in learning institutions and summarizes a comparison between them. Section 6 presents taxonomy of web 2.0 technologies that can be used in learning institutions, highlighting its different classes and components. Section 7 takes a closer look on the future of attended classes based on presented facts and highlighted trends. Section 8 presents future research directions for comprehensive utilizing of web 2.0 technologies and a look on technologies that transitions e-Learning into e-Learning 3.0.

## 9.2 Web 2.0 Technologies

Ever since group behavior became an important part of mainstream websites, companies have found numerous ways to exploit the behavior of web surfers. Simple uses of web 2.0 ideas include businesses like amazon.com soliciting product rankings from consumers and offering shoppers hints of what others have bought. Moreover, notions of web 2.0 include the many social networking web sites that have tried to profit from the basic human need to connect with others. Facebook, MySpace, Friendster, Tribe, LinkedIn, Spoke, and countless others have looked to profit on connecting teens, communities, professionals and just about any other type of demographic. The web 2.0 phenomenon is more rightly described as a social and behavioral sea change. Instead of serving up static fixed content to web surfers, the web has become an interactive place for people to congregate and do things together virtually. Web 2.0 is about the way people use web 1.0, not about the web itself [223]. Web 2.0 refers to the social use of the web which allows people to collaborate, to get actively involved in creating content to generate knowledge and to share information online [141].

Most instructors are interested in utilizing whatever their hands reach to in enhancing and supporting the learning process. Instructors are surrounded by acronyms that are completely new to them, and there is not enough time to get used to one of the technologies before another becomes available. Figure 9.1 presented in page 185 highlights instructor distraction between different acronyms that are made available and easy for students to utilize and are under study by e-Learning



Figure 9.1: Instructor Challenges between Students and Researchers

researchers. All those technologies are available, but there are always some questions that need answers: How to utilize them? What are the efficiency and drawbacks on the learning process and on learners from utilizing them? And are those technologies convenient with the learning institutions policy?

Figure 9.2 presented in page 186 presents a collection of different Web 2.0 implemented technologies that can be utilized in e-Learning 2.0. Presented list is not meant to include all presented examples because they are plenty and the list is updated every day. However, it is meant to include the most accepted and highly deployed implementations of web 2.0 technologies. Those examples gained a huge acceptance and access from online users. This chapter does not tend to present a full coverage and/or a list of highlighted web sites' users. The following numbers are just for proof of concept. Table 9.2 presented in page 187 presents some facts about web 2.0 online usage statistics based on Google Ad Planner statistics on November 9, 2009. All statistics are in approximate.

Google Ad Planner is a free media planning tool that can help identify websites where audience is likely to visit. Table 9.2 presented in page 187 Columns details are:

- Cookies: approximate number of cookies on a site over a specific month.
- Visitors: estimated, unduplicated number of people who visit a site over a month.
- Reach: total estimated number of users you can reach on a specific site.



Figure 9.2: e-Learning 2.0 Supportive Technologies Online Implementations

Table 9.1: Statistics about Popular Web 2.0 Sites Usage

Site	Cookies	Visitors	Reach	Page Views	Total Visits	Avg. Visits	Avg. Time
Twitter	120 M	66 M	5.2%	4.4 B	600 M	9	12:10
Facebook	710 M	410 M	32%	220 B	10 B	25	23:20
MySpace	190 M	99 M	7.6%	29 B	1.1 B	12	21:40
WikiSpaces	3.8 M	2.2 M	0.2%	50 M	6.7 M	3.1	8:10
LinkedIn	67 M	35 M	2.7%	1.7 B	190 M	5.5	9:50
Blogger	66 M	42 M	3.2%	1.7 B	210 M	5	10:50
MindMeister	290 K	170 K	0%	3.8 M	760 K	4.6	10:40
LiveStream	1.4 M	740 K	0.1%	15 M	2.9 M	3.9	12:00
Flickr	120 M	65 M	5.1%	3 B	250 M	3.8	9:00
Zoho	1.5 M	830 K	0.1%	51 M	5.6 M	6.8	15:50
SlideShare	16 M	9.1 M	0.7%	66 M	21 M	2.4	6:00

Table 9.2: Five fastest growing "member destinations" in the U.S

RANK	Site	Feb 08	Feb 09	% Growth
1	Twitter.com	475,000	7,038,000	1382%
2	Zimbio	809,000	2,752,000	240%
3	Facebook	20,043,000	65,704,000	228%
4	Multiply	821,000	2,394,000	192%
5	Wikia	1,381,000	3,758,000	172%

- Page Views: total estimated number of times pages on a site have been accessed.
- Total Visits: estimated number of times a site is accessed by unique visitors.
- Average Visits Per Visitor: estimated number of times a unique visitor accesses a site over a specific month.
- Average Time on Site: estimated average amount of time, in seconds, that a unique visitor spends on a site.

Total number of users and visits mentioned in table 9.2 presented in page 187 exceeds billions. What really matters most is the Growth Rate of web 2.0 sites. Twitter for example is a phenomenon that shall be closely studied and made use of in different aspects of twitters users' lives. In December 2008, 11% of online adults said they used Twitter or another service to update their status online [242]. In April 2009 study showed that the percent had increased to 19% of internet users [135]. Table 9.2 presented in page 187 presents a new survey from Nielsen about the five fastest growing "member community destinations" in the U.S.

Table 9.2 presented in page 188 presents a comparison between the top rated technology tools used in conjunction with education from students' perspective, and

Table 9.3: Technology conjunction by Students/Instructors

Technology	Students	Instructors
Social Networking and Sites	52%	14%
Web Applications	31%	12%
iPod / MP3 player	31%	8%
Wikis	28%	11%

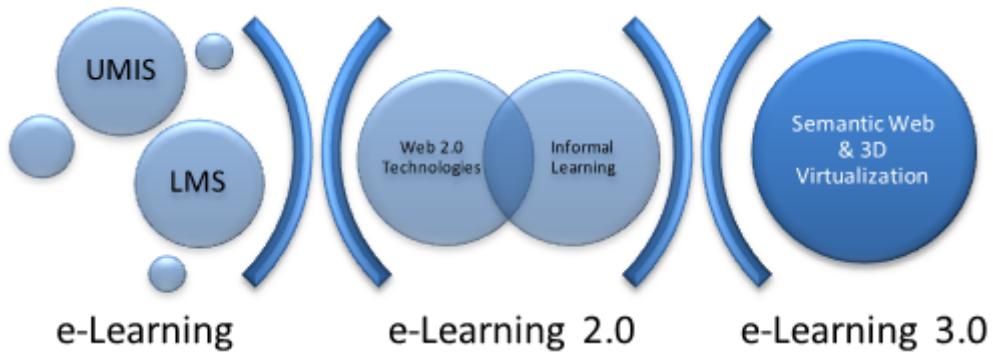


Figure 9.3: e-Learning Transitions

the same tools utilization in teaching from instructors' perspective based on data from the 2009 21st Century Campus Report [82]. There is a huge gap that hopefully will be addressed by different web 2.0 utilization models. Though both use web 2.0 technologies outside the class, instructors and students still rely on traditional methods to communicate with each other.

### 9.3 e-Learning 2.0 via Web 2.0

The web 2.0 effect on e-Learning is transitioning it into e-Learning 2.0. e-Learning witnesses now the middle era that is characterized by the presence of “informal learning”, and that is preparing it for e-Learning 3.0. e-Learning 3.0 is characterized by utilization of semantic web or what is called web 3.0 and 3D virtualization that will play a significant role in this transition in the near future as depicted in figure 9.3 presented in page 9.3. A closer look on e-Learning 3.0 will be presented in future research directions section of this chapter.

While the focus in e-Learning was on presenting ISs that manages the abovementioned activities and tasks in section 1, the change in the internet that is enabled by web 2.0 technologies reflected directly on the learning process. e-Learning 2.0 presents opportunities for instructors to utilize new technologies and overcome challenges and limitations of current learning process, get students in the middle of the learning process, embed technologies that today students live within the learning process hopefully reaching a life-long learning model that seamlessly integrates with students lives. It has been remarkably noticed there are boundaries between

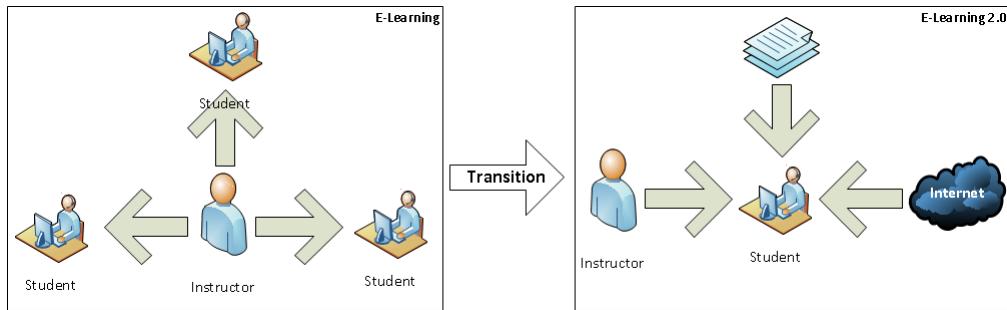


Figure 9.4: e-Learning 2.0 Transition in Information Sources

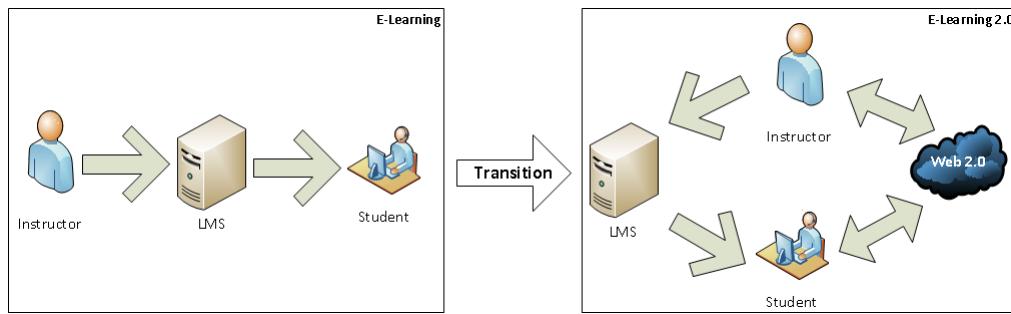


Figure 9.5: e-Learning 2.0 Transition in Feedback

what students utilize during learning and what they utilize during spending time on the internet. e-Learning 2.0 tends to enhance the learning process, make use of informal learning opportunity, and present a model of “life-long learning” via a bulkhead removal. Figure 9.4 presented in page 189 shows the transition from e-Learning to e-Learning 2.0 in information sources. In e-Learning, the instructor was the main source of information for students. He/she was responsible for preparing the learning materials, defining learning plan and other learning activities. Though this fact still applies to e-Learning 2.0, instructors shall not ignore the fact that students have different sources of information. Figure 9.5 presented in page 189 highlights the transition from e-Learning into e-Learning 2.0 by presenting a new informal feedback method. Students might feel more comfortable in responding to instructors and communicating over informal feedback channels more than they do in the formal feedback environment. Instructors can make use of this comfort in conducting informal assessments that help instructors get a closer look and feel of their students. Figure 9.6 presented in page 190 presents one of the transition forms from e-Learning into e-Learning 2.0 from flow of information transition. Students in e-Learning 2.0 play an important role in updating information, not just receiving it. They can participate in updating learning materials based on their perception and understanding of learning topics.

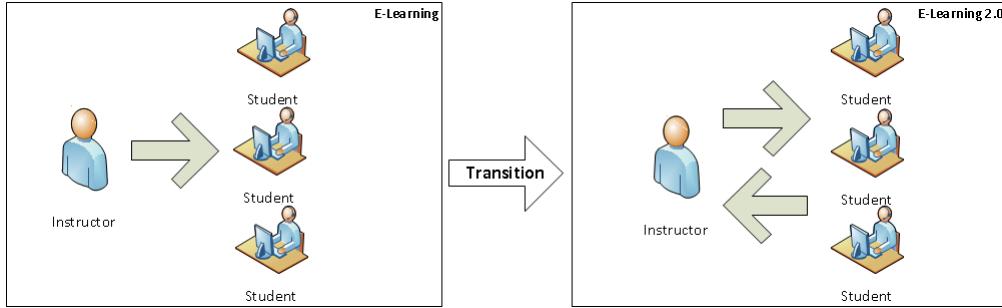


Figure 9.6: e-Learning 2.0 Transition in Information Flow

## 9.4 Web 2.0 in Learning Institutions

Web 2.0 research in e-Learning is a research point of interest. Collaboration, semantic web, ontologies, Web services, SOA and many other technologies have been presented, utilized and evaluated within the learning domain. This section presents different research papers, implementation practices and case studies of adopting web 2.0 technologies in learning institutions. Web 2.0 platforms are seen to have an emerging role to transform teaching and learning [52]. One of the research papers that address different scenarios for utilizing web 2.0 technologies in e-Learning is the one presented in [141]. Some possibilities and examples of using web 2.0 technologies as a support for preparing and collecting didactic materials, evaluating and analyzing the progress made by students, putting together informative and formative presentations, time management, planning the timetable and the calendar of activities, developing projects in collaboration, digital storytelling, students' e-portfolios, and others were rendered. Rendered web 2.0 technologies include: blogging, micro blogging, wikis, photo/slides sharing, video sharing, syndication of content through RSS, social bookmarking, social networking, and other tools. Advantages and challenges of utilizing web 2.0 technologies in the learning process, and the need to invent pedagogy 2.0 that goes along with e-Learning 2.0 enabled by web 2.0 technologies' utilization in the learning process were addressed. One of the available online guides for implementing web 2.0 in e-Learning is the one presented by Downes [106]. This article presents ten things to do via different technologies to make the maximum benefits out of web 2.0. Learning activities include: podcasting, video casting, blogging, slide sharing, Goggling, and commenting on others' posts and activities, socializing via facebook, and other activities.

The three main players in the learning process need to co-operate to adopt and implement web 2.0 technologies in the learning process. They are: learning institution's management, instructor, and student. Learning institution's management presented by Chief Information Officers (CIOs) give high priority for advanced learning features provided by new technology trends. Agee et al. state that teaching and learning with technology is ranked #5 this year, moving up from #9 in the 2008 survey as the top priority of CIOs [48]. CIOs have become crucial to instructional units because they provide leadership in evaluating and supporting the teaching technologies that underlie multiple forms of distributed learning. A growing proportion of

learning takes place outside the traditional boundaries of the classroom facilitated by applications such as social networks and technologies that support a culture in which everyone creates and shares. CIOs are being asked to provide technological directions for cultural transformations such as information fluency that involve library faculty, department faculty, technology specialists and students as co-creators of knowledge. Finding the proper balance between systemic and ad hoc technologies will be fundamental for IT leaders as they respond to a student generation that prefers less passive and more agile learning. These instructional modalities will foster transformational innovations such as the need for e-portfolios in a reflective, contextual, authentic and active learning environment.

On the other hand, Ajjan and Hartshorne assess faculty's staff members' awareness of the benefits of web 2.0 to supplement in class learning and better understand of faculty's decisions to adopt these tools. Findings indicated that while some faculty members feel that some web 2.0 technologies could improve student's learning, their interaction with faculty and with other peers, their writing abilities and their satisfaction with the course; few choose to use them in the classroom [49]. Additional results indicated that faculty's attitude and their perceived behavioral control are strong indicators of their intention to use web 2.0. The results highlight that while a somewhat considerable proportion of the faculty felt that selected web 2.0 technologies would likely provide their students with many benefits, only few chose to use it. The lack of experience with most web 2.0 technologies examined in this study could drive faculty members to avoid their adoption although they realize that this adoption would provide their students with many important benefits.

Küfi and Özgür present an elaboration on the effectiveness of the most recent web based tools from the student perspective [181]. The present study aimed to develop a comprehensive insight into two hundred freshman students' perceptions regarding the use of an interactive web environment in English communication courses offered by the department of General Education at Eastern Mediterranean University. Utilized web 2.0 technologies include: wiki and moodle. Though utilized technologies are really so simple, they achieved the required objectives and provided students with the suitable environment to create their own interactive web environment for their classes. The analysis of data shows that the majority of students is positive about the use of an interactive web environment and finds its use beneficial for their learning. Based on the research results, 50% of students enjoyed learning English by using wiki/moodle, and 52.5% of them agree that the interactive web environment created for English course helped them to improve their English.

## 9.5 Examples of Web 2.0 in Learning

Utilizing web 2.0 technologies in learning institutions is a hot topic and a brand new field that depends mainly on each learning institution's capabilities, infrastructure, instructors' and students' readiness and acceptance of applying those technologies. Real world examples are important in showing what others have achieved, how they achieved it and learning from their watches. Section 9.5.1 presented in page 193 presents summary and comparison between 5 presented real world examples highlighting utilized technology, technology providers, output of utilizing those

technologies and impact on students.

An example of adopting community platform for educators is the one presented by College of Liberal and Professional Studies at the University of Pennsylvania “<http://www.sas.upenn.edu/lps/>”. College sought a platform that would provide a more engaging and participatory environment. Its online courses based on the two utilized open source modules utilized in the College: moodle for managing learning activities, and drupal that is the content management system. GoingOn “<http://www.goingon.com>” is building a platform for a University of Pennsylvania psychology course in the institution’s continuing-education program. The psychology department had graduates who had become psychologists who wanted to learn more to improve their professional practice, as well as learners who wanted to improve their lives. Nearly a thousand student took the course, called "foundations of positive psychology." The interface allowed students to form their own "affinity groups" based on topics of particular interest. The entire platform was able to draw student information from moodlerooms, also built on open-source applications [131].

Another example is Epsilen that is an e-Learning platform built around social networking “<http://www.epsilen.com>”. Epsilen announced a partnership with Sun-Gard Higher Education “<http://www.sungardhe.com>” to draw on student information in a similar way. Epsilen environment lets faculty members use online material from The New York Times both for assignments and to promote discussion. An example of this utilization was achieved by a professor of international affairs who left his Epsilen group open after the class finished and final grades were posted. Months later, there was a spike in online activity and discussions. Students were watching the olympics and had started discussing the athletic competition in the context of the international politics issues raised during the class [225]. Epsilen partners list include Bowling Green State University, and Ohio University.

Utilizing collaborative digital information tracking in Universities is facilitated by iParadigms “<http://www.iparadigms.com>” that is one of the leaders in the field of textual intellectual property protection. iParadigms has announced an update for its peer review Web service named PeerMark to include collaborative features [236]. iParadigms lets instructors create and manage assignments in which students read and provide comments on each others’ work. Faculty use a web interface to set up assignments, provide structured guidance and monitor the results with minimal effort. Students upload, review and comment on each other’s papers, prompted by instructor-supplied questions and guidelines. iParadigms list of users is available at <http://www.iparadigms.com/ourusers.html>.

ConnectYard “<http://www.connectyard.com>” which builds education-related applications has developed a widget that can be added to a CMS for delivering queries to the user via social networking sites and other means. QuickConnect widget works with blackboard, moodle, sakai and other LMS platforms, according to the company. Students post homework and study questions via an HTML form on the QuickConnect widget. Those questions are then delivered to other class members via their preferred method. Both students and instructors have the option of replying to questions via facebook, twitter, text message, or e-mail. All responses are then added to the discussion thread within the CMS for other students and instructors to view and respond to. Threaded discussions can be forwarded to other classes and

groups to expand the conversation.

Purdue University in West Lafayette, Indiana, USA “<http://www.purdue.edu/>” has recently brought social networking to the classroom [202]. Some professors especially those who teach in large lecture halls have come to embrace social networking as an instructional aid via using an application developed on campus. Hotseat “<http://www.itap.purdue.edu/tlt/hotseat/>” allows students to comment on the class and then enables other participants including professors, students and teaching assistants to view those messages. Students either use their twitter, facebook or MySpace accounts to post the messages or log in to the Hotseat web site to send text messages. The application resides on the web so there is no software for professors or students to install. Hotseat was intended as a way to manage the logistics of teaching a classroom of 100 plus students. Hotseat provides a better way for students to engage the instructor and each other in terms of classroom discussion, and to encourage that type of interaction both in and out of the classroom.

### 9.5.1 Summary and Comparison

#### 1. Example 1

- *College/University:* College of Liberal and Professional Studies at the University of Pennsylvania
- *Department / Subject:* Psychology Department, Foundations of Positive Psychology Course
- *Utilized Technology:* Community Platform for Educators “Community Groups”
- *Addresses:* Psychologists who wanted to learn more to improve their professional practice, as well as learners who wanted to improve their lives
- *Affected Students:* Nearly a thousand student took the course, called "Foundations of Positive Psychology"
- *Output:* “Affinity Groups” based on particular interests
- *Impact:* More engaging and participatory environment its online courses
- *Technology Provider:* GoingOn
- *Application Capabilities:* Draw student information from Moodlerooms, also built on open-source applications

#### 2. Example 2

- *College / University:* SunGard Higher Education
- *Department / Subject:* International Affairs Subject
- *Utilized Technology:* e-Learning platform that is built around social networking
- *Addresses:* Current Enrolled Students
- *Affected Students:* N/A

- *Output:* Discussing the athletic competition in the context of the international politics issues raised during the class
- *Impact:* Applying what they have learned in class (months ago) in real-world scenarios “taking what they have learned into action”
- *Technology:* Provider Epsilen
- *Application Capabilities:* environment lets faculty members use online material from New York Times for assignments and to promote discussion

### 3. Example 3

- *College / University:* List of Users is available in Resources Section
- *Department / Subject:* N/A
- *Utilized Technology:* Collaborative Digital Information Tracking
- *Addresses:* Current Enrolled Students
- *Affected Students:* N/A
- *Output:* Instructors create and manage assignments in which students read and provide comments on each others' work
- *Impact:* Students upload, review, and comment on each other's papers, prompted by instructor-supplied questions and guidelines, thus more supervised interaction, engagement, and collaboration
- *Technology Provider:* iParadigms
- *Application Capabilities:* Web interface to set up assignments, provide structured guidance, and monitor the results with minimal effort

### 4. Example 4

- *College / University:* List of Users is available on Web site Home page
- *Department / Subject:* N/A
- *Utilized Technology:* Widgets, Social Networking, Text Messaging
- *Addresses:* Current Enrolled Students
- *Affected Students:* N/A
- *Output:* Threaded Discussions that is integrated in CMS and updated via Facebook, Twitter, Text Message, or e-mail.
- *Impact:* More involved students in the learning process, Utilizing Web 2.0 technologies in the learning process
- *Technology Provider:* ConnectYard
- *Application Capabilities:* Widget that can be added to a CMS for delivering queries to the user via social networking sites and other means

### 5. Example 5

- *College / University:* Purdue University in West Lafayette, Indiana, USA

- *Department / Subject:* Courses with Classrooms of 100-plus students
- *Utilized Technology:* Social Networking
- *Addresses:* Current Enrolled Students (Traditional, Blended Learning Models)
- *Affected Students:* N/A
- *Output:* Comments and Discussions
- *Impact:* better way for students to engage the instructor and each other in terms of classroom discussion, and to encourage that type of interaction both in and out of the classroom
- *Technology Provider:* In-house (application developed on campus).
- *Application:* HotSeat
- *Application Capabilities:* Students can comment on the class via their Twitter, Facebook or MySpace accounts, or send text messages

## 9.6 Web 2.0 for Learning Institutions

Figure 9.7 presented in page 196 presents taxonomy of web 2.0 technologies. Though some of those technologies were available before the presence of web 2.0, however they are meant to be utilized in different ways that will unleash their capabilities in supporting e-Learning. Web 2.0 technologies will be categorized into: in-lecture, and after-lecture. As the category name indicates: in-lecture technologies will be utilized during the lecture, while after-lecture technologies are presented to enable the new e-Learning trend that is based on what students' really utilize, not just what learning institutions used to make available to them. Learning process is not simply and easily divided like mentioned here because a certain overlap between in-lecture and after-lecture should be presented to ensure students' involvement in the learning process, and to provide the most mature learning environment. It is well known that no single model fits all situations, and it is the instructor's responsibility with the learning institution to evaluate their current situation, judge different technologies and decide what technologies to utilize during the learning process. Besides, though the same technology can be utilized in more than one learning activity, it is important to ensure that this technology is used where it fits. Slight pedagogical differences between in-lecture and after-lecture activities should be considered as presented in table 9.6 presented in page 196.

Same polls / surveys technology can be used in-lecture or after-lecture, however it must be clear for instructor that when it is used in-lecture, it is used in an informative way to get an idea on student's following, understanding of the topics and their satisfaction with the learning flow, topics, content and any other instructor's points of interest. However, when the same polls / surveys technology used after-lecture, it is used to give a summative feedback on the learning experience took place during the previous lecture/session. That means, informative knowledge can affect the lecture flow immediately while summative knowledge affects the next time activity. They both differ in nature and importance and can be achieved via same technology.

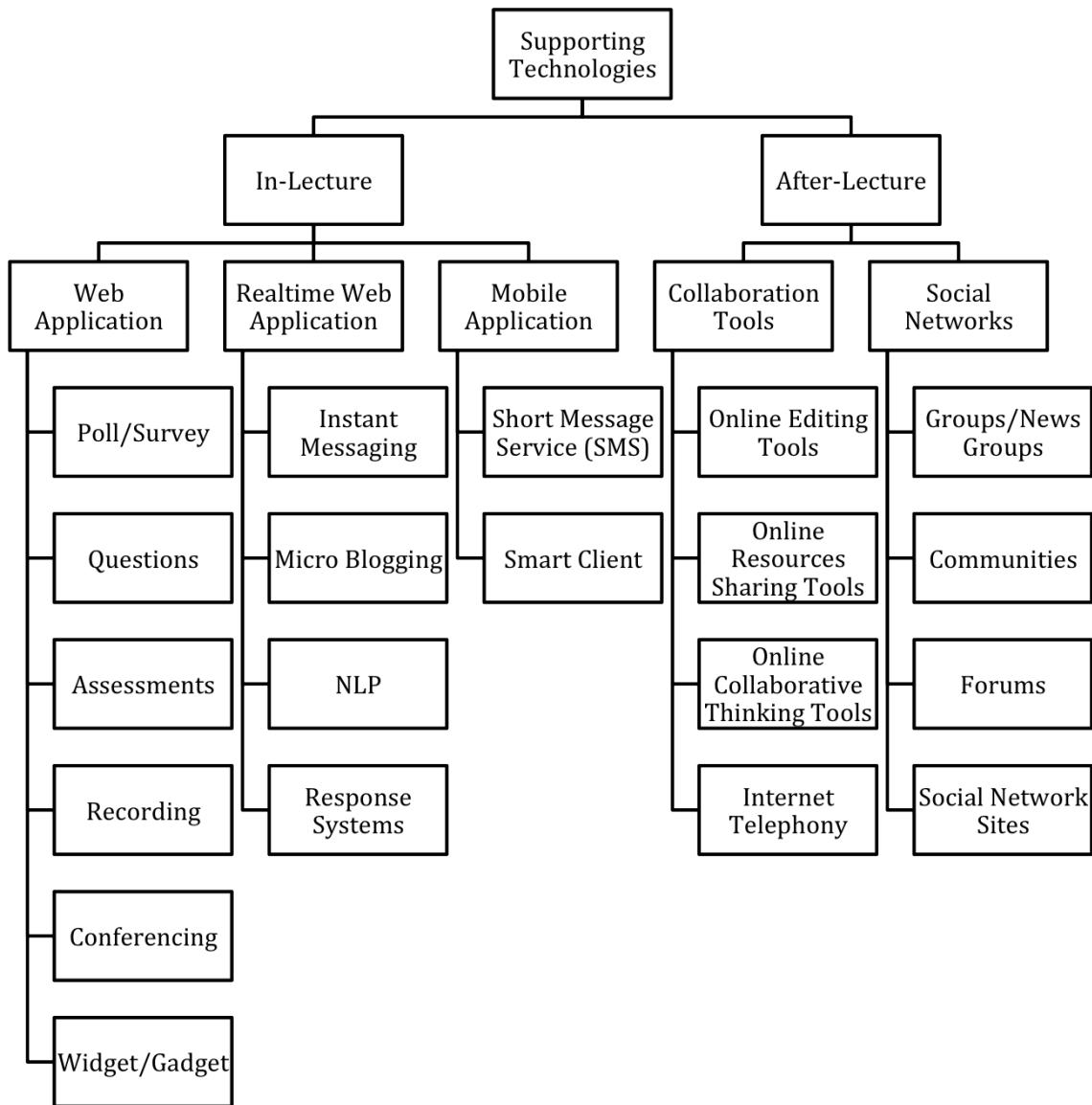


Figure 9.7: Taxonomy of Adaptive Lecture Technologies

Table 9.4: Differences between In-Lecture and After-Lecture Activities

Activity	In-Lecture	After-Lecture
Polls / Surveys	Informative	Summative
Interaction from Student to Instructor	Questions	Discussions
Interaction from Instructor to Student	Assessment for Learning	Assessment of Learning
Student Activity	Simple Tasks	Assignments

## 9.7 In-Lecture Web 2.0

It is important to consider three lecture aspects: basic required functionalities, real-time data, and mobility. Basic required functionalities will be provided via the standard web application, due to high requirements that is still hard to provide over mobile devices. Real-time data plays an important role in monitoring students' status, performance and satisfaction during lecture. Mobility shall be considered, facilitated and managed effectively. Mobile devices include mobile phones, smart phones, PDAs and any other handheld devices.

### 9.7.1 Web Application

Technically, web applications are applications that reside on a web server that is accessible over computer networks via a simple user client that is mainly web browser, and requires no user installations. From user experience, web applications are the most desirable user applications because they take the entire load off the user and provide all the required functionalities. Learning institutions have been using web applications in the learning process for a while. El-Ghareeb presents a list of open source, free and commercial LMSs [112]. Utilizing the following technologies within a pedagogical aspect yields wider effect on the learning process. Different Web application technologies that can be utilized include:

- Polls / Surveys: they are important in the learning process because they give instructors a suitable start point to begin at and an indicator on students' performance during lecture. Instructors can use polls/surveys capabilities to check how many students have prepared for the lecture, have background on the topic to be discussed, interested in learning the current topic and many other criteria that affects the lecture path. Polls/surveys are easily integrated in the learning process from technical point of view, and they produce a high value in the learning process from the pedagogical aspect. They give the lecture a personalized theme that students can feel based on their feedback and responses.
- Questions: instructors might permit questions during lecture, or at the end of a lecture's sections, or at the end of the lecture. This varies based on the instructor's evaluation of the situation and the point s/he is discussing. Students can submit questions while the instructor is discussing the topic. When instructor feels it is appropriate to take questions, s/he reads submitted questions and starts answering them. Different implementations for questions can be presented in the lecture. One of the models includes presenting an (ask question) application for student to utilize during lecture. Students can submit questions online during lecture, and they immediately appear to instructor. It is instructor's choice to answer this question immediately or later. Options like making questions viewable by all attendees, and/or identifying who asked the question are controlled by instructor. Another model might give students the capability to digitally raise hands, so they indicate they have a question without the capability for them to post their questions. A third model might

make use of available teaching assistants and allows them to handle questions during the lecture, so questions don't interrupt the lecture flow.

- Assessments: students' progress measure in learning new topics is revealed by assessments. Different types, standards of quality, and utilization of assessments are available online. Assessment is used mainly to measure student's progress and act as a starting point in enhancing the student's learning experience. Different types of questions are made available via computers. IMS QTI standard "<http://www.imsglobal.org/question/>" addresses 11 types of questions that can be presented and handled via computer technologies.
- Recording: The 2009 21st Century Campus Report (2009) stated that 61% of students identified the need for a video or voice recording mechanism for lectures where only 36% of instructors paid attention to this point [6]. Recording is an activity that produces a digital audio and/or video file that records all lecture activities, and gives students the ability to subscribe to and download via RSS feed to digital devices like an iPod or a desktop computer. A camera needs to be installed in the classroom to record class activities and upload them to the server.
- Conferencing: facilitates students' attendance because they help students overcome challenges that might prevent them from attending sometimes. The 2009 21st Century Campus Report (2009) stated that 70% of students identified the need for distance learning capabilities to connect students in multiple locations, against only 40% of instructors who paid attention to this requirement [6]. Students can join the conference from their locations and it is equivalent to attending the lecture. Seamless integration might be required if instructors/learning institution utilize attendance tracking system.
- Widget / Gadget: they are a piece of reusable code that can be plugged into virtually any website. They have gained popularity because they give different capabilities for internet users like accessing different sites from the same web page. Users don't have to leave current page to check for their social network sites, update their status, send emails, contact using messengers and do many other activities. One of the largest online Widget/Gadget galleries is the one presented and maintained by iGoogle "<http://www.google.com/ig/directory>". Widget / Gadget as a technology that takes the implementation responsibility off the web site owners' shoulder and still presents the capability to integrate different functionalities from different sites can be utilized as both concept and products in the learning portals.

### 9.7.2 Real-Time Web

Different technologies that enable users to communicate in real-time are available, however utilizing those technologies hasn't been addressed in the optimum way. Further advanced research on available technologies, and enhancements to them is required. Real-time technologies that can be utilized in-lecture include:

- Micro Blogging: it is simply about sharing ideas, thoughts, activities, actions and updates in a small number of characters. Twitter is the most famous micro blog available, giving the capability to share whatever user wants to within 140 characters. Micro blogging is a real effective and touching technology. It is an easy way of sharing updates at real time. Students can easily access twitter from lecture and they are intimidating to do so already. It is surprising that twitter users tend to utilize news, so they can consume learning inside lecture as they consume news [242].
- Instant Messaging (IM): is a form of real-time communication between two or more people based on typed text. The text is conveyed via devices connected over a network such as the local institution network, or the internet. IM can be used mainly in the questions activity mentioned in the web application section and of course in many others. IM has taken new shape lately with the presence of converged networks, and the capability of multimedia networks. Voice over IP (VoIP) and multimedia enabled communication devices will revolutionize the way we utilize IM in the near future.
- Natural Language Processor (NLP): field of computer science and linguistics concerned with the interactions between computers and human natural languages. Natural language generation systems convert information from computer databases into readable human language. Natural language understanding systems convert samples of human language into more formal representations that can be altered and processed to generate useful information based on students' submitted data. Further research on mining student's real-time data submitted during lecture can be used in generating useful information about students' satisfaction with lecture instead of relying on teaching assistants for summarizing and analyzing students' feedback.
- Response Systems: allow users to respond with short answers immediately. Response system consists of two parts: the two ways True/False Clicker (TF Clicker) that students can use to answer questions immediately and an administration application that enables instructors to initiate questions and analyzes answers. Figure 9.8 presented in page 200 presents a block diagram of TF Clicker. Students can respond to open-ended questions with alphanumeric answers by typing in text in a way similar to texting on mobile phone [210]. Renaissance Learning "<http://www.renlearn.com>" is one of the leading companies in providing Response Systems.

### 9.7.3 Mobile Applications

Utilizing mobile in learning is a wide area of research that is under evaluation, arguments and challenges. Mobile Learning (M-Learning) is an approach to e-Learning that simply utilizes mobile devices, yet it can also be viewed as a quiet different learning experience [163]. M-Learning has been used as a pre and/or post activity to other types of learning [270]. Mobile phones are widely popular and made

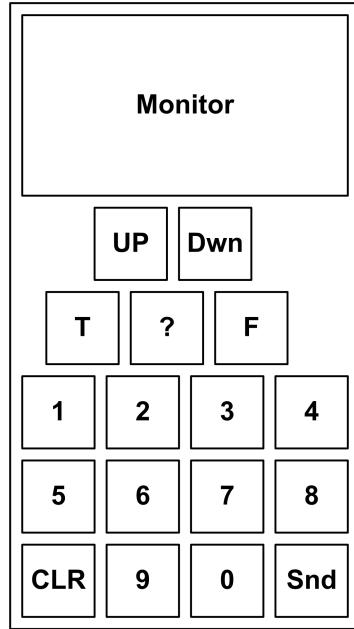


Figure 9.8: Block Diagram of TF Clicker

available to almost everyone in the globe and it is only a matter of time till everyone holds a mobile phone. Users can achieve different activities from their mobile phones, like accessing emails, creating and editing documents, attaching and emailing them, updating calendars and many other activities. More than three-quarters (76%) of twitter users use the internet wirelessly either on a lap top with a wireless connection or via PDA, handheld or cell phone. In comparison, 57% of those who go online but do not use twitter and 59% of internet users as a whole connect to the internet wirelessly [242]. On the other hand, utilizing mobile phones is challenged and limited by mobile computing capabilities, small screen size, low screen resolution, short battery life and other challenges that are still limiting its wide utilization in different life aspects. Riad and El-Ghareeb presented a mobile utilization in the assessment activity as a form of utilizing mobile technologies in the learning process [114]. Mobile applications are enabled mainly via:

- Short Messaging Service (SMS): simplest way of communication using mobile phone. Sending SMS is an activity that includes editing the text message within a predefined number of characters, defining the recipient number and submitting the message. It is possible to force series of interactive SMS exchanges between student and LMS to achieve completion of a task or goal. Student will take part and complete the task [250]. SMS utilization in proposed Adaptive Lecture Model can fit within Polls / Surveys.
- Smart Clients: provide users with a rich and responsive user interface, the ability to work offline, and a way to take advantage of local hardware and software resources. In addition, they can be designed to run on a broad spectrum of client devices, including tablet PCs, and handheld mobile devices such as pocket PCs and smart phones. Smart clients give users access to informa-

tion and remote services within a powerful and intuitive client environment and are an effective solution for flexible user-oriented applications and for increasing user productivity and satisfaction. Smart clients share some or all of the following characteristics: make use of local resources, make use of network resources, support occasionally connected users, provide intelligent installation and update and provide client device flexibility [154].

## 9.8 After Lecture Web 2.0

After-lecture activities are very important in the learning process. These activities ensure that learning is moving from acquiring, gaining and recalling information to utilizing information in real-life situations and scenarios to become knowledge. Discussions, collaboration and socially utilizing information gained in lectures are important in all fields. From author's point of view, collaboration tools and social networks are the most important today's available online tools to integrate the informal learning in the learning process.

### 9.8.1 Collaboration Tools

Utilizing different collaboration technologies that exist will result in pedagogical effect. Educational collaboration ISs are responsible for facilitating collaboration technologies utilization within learning institutions. Collaboration technologies and tools include:

- Online Editing Tools: wikis, blogs, online documents editing, and mind mapping tools are some of the examples of collaboration tools that can be made available to students and to enrich learning process by motivating learning stakeholders to enrich learning.
  - Wiki: is simple collaborative editing software application which encourages users to be not only readers, but also content providers and editors [181]. Wikis have proven gained pedagogical and educational benefits. They encourage collaboration, and enhance writing, editing, discussion and critical skills.
  - Blog: are user journal entries in the form of text, images and links to web content such as websites or other blogs that have a variety of formats and might include the user expressing their opinion about a topic or documenting activities. Blogs are interactive in the sense that other users could provide comments on the information posted by the blog author. Educational applications of blogs include researching, tracking, interpreting and evaluating blogs for political commentary (multiple perspectives), cultural events, business or other news and for examining changes over time [49].
  - Mind Mapping Tools: they are important in putting different ideas into work, then start discussions in attempt to get the best out of them, and

to relate and organize those ideas. They can be best used in problem solving sessions.

- Online Documents Editing Tools: they provide different services that enable more than one to edit online documents and see changes immediately. Other complementary services include the ability to upload files from users' desktops and save them online, editing online files anytime, anywhere without the need to install software other than the web browser, defining who can access those files, storing files securely online, importing and exporting different types of files, publishing those documents online when finished, so students can easily publish online documents that are accessible, enabling readers to leave comments and other features.
- Online Resources Sharing Tools: online workspaces and web drives for sharing files, maintaining versions, managing and arranging files within virtual folders and storing files online. Complimentary services like bulk files uploading, downloading, file sharing, mailing short links to available online files and other services are provided by multiple online resources sharing service providers.
  - Sky drives / Web drives: different web sites provide the service of uploading files and presenting short links for them to be downloaded later. Though this service has enabled files sharing online, obligations, restrictions and considerations about copyrights must be taken into consideration. The most suitable scenario for students is the Workspaces.
  - Workspaces: store online files to be shared between students and combines the functionality of online documents editing tools, so students need a web browser and Internet connection to access the online stored files, open and edit them via Web browser anytime and anywhere, so students can easily collaborate and share files in a productive manner.
- Online Collaborative Thinking Enabling Tools: they are important in sharing resources that students feel important to them in formalizing their ideas about some topics. Students might be working in groups in different topics in their graduation projects / lab projects and they might incorporate an online article that they feel the need for the rest to take a look. Social bookmarking services present this ability. Besides, online mind mapping tools are important in finalizing the overall ideas of the project they are working on, the document they are editing, and whatever collaborative activity they are attempting.
  - Social Bookmarking: allow users to store, describe, and share numerous web addresses with others. Users can explore bookmark collections of others by subscribing to their bookmark pages. If users are interested in a site, they could tag it using few words to help others find it easily. Educators could use social bookmarking to facilitate collaborative information discovery. They could create a social bookmarking page to save important pages about a topic. Students could also collaborate on group projects using bookmarking sites, sharing links and uploading resources discovered while educators could follow their students bookmark pages to gain insight on their research [49].

- Mind Mapping Tools: they don't edit the thing together. They share it for each other and receive comments on it. Generally, they don't need so much collaboration in editing as it needs in discussion.
- Internet Telephony: it has witnessed a huge advancement in the late years as a result of the widespread of VoIP communication devices that utilize internet communication infrastructure in making phone calls. Students can communicate from PC to PC free of charge using Headsets and a free service provider. Also, they can communicate from PC to phone using headsets and a simple charge for the service provider. Students can use their emails as a "voice mail" and leave each other recorded voice files.

### 9.8.2 Social Networks

Social networks are a social structure of nodes that represent individuals (or organizations) and the relationships between them within a certain domain. Therefore, social networks are usually built based on the strength of relationships and trust between the members "nodes" [190]. People have been forming themselves into groups without the gratuitous nature of the internet or social software.

- Groups: they have been a widely spread method to communicate with people that sometimes don't know each other in a productive way to share ideas and resources via sending emails to one email address that is the group email and the group service provider is responsible for delivering those emails to group subscribers based on their registration features. Registration features specify the details of mails to be delivered, delivery rate, preferred format and other features. Students read the emails, follow and contribute to the conversations they feel most important to them and ignore what they want. Though this is not the most perfect way of collaboration nowadays, groups have helped many students and professionals to communicate in an interactive way.
- Communities: different types of communities exist. The most related to interest is Community of Practice (CoP). A group of people in an online group with a common interest who share knowledge and expertise. Following a learning course, the participants form a CoP hosted by a leader to share material related to a shared experience or a course, get help on problems that arise and share best practices [79].
- Forums: online information sharing tool that allows students to submit questions and answer each other. A great tool in discussing updated topics and searching within history of answered questions and sharing experiences. Organizing topics under headers is an available option. Forums don't need immediate responses, so they are an important asynchronous communication form.
- Social Network Sites: A social networking site is an online place where a user can create a profile and build a personal network that connects him or her to other users. In the past five years, such sites have rocketed from a niche activity into a phenomenon that engages tens of millions of internet users [187].

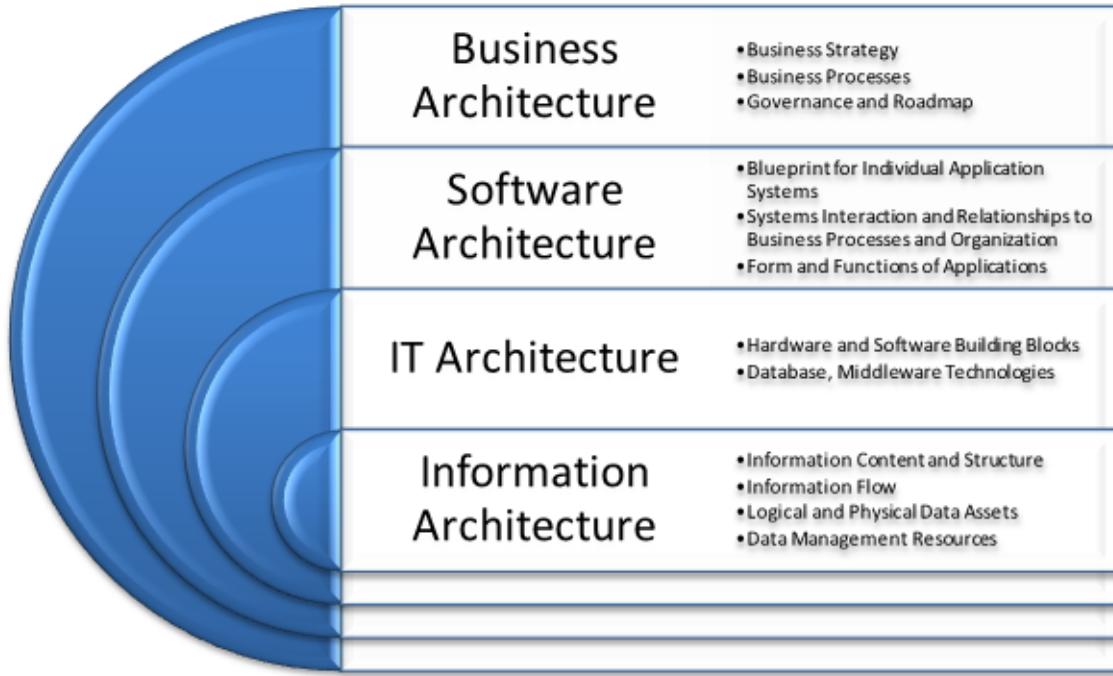


Figure 9.9: EA Dimensions and Roles

## 9.9 Future of Attended Classes

Attended classes will witness changes as a result of web 2.0 technologies presence and the sake of integrating it into current learning process. For this change to achieve success, it needs to include the entire enterprise to define a broader prospective of University architecture. Enterprise in this case is the University, thus EA and university architecture are used interchangeably. EA defines the enterprise's different dimensions, connect between them and present a complete view for the enterprise environment to face the ubiquitous challenges. It has become accepted that there is a clear need for an architectural view of systems [198]. EA dimensions are: Business architecture, Software “Application” architecture, Information Technology (IT) architecture and Information architecture [51]. EA classes utilize each other and build over each other, and there are no clear distinctions between them. Figure 9.9 presented in page 204 presents the EA four dimensions and each dimension role.

Adopting web 2.0 technologies in current attended classes requires certain amount of enterprise agility that is provided by utilizing SOA within the four enterprise dimensions. Utilizing SOA provides educational institutions with both business and technical agility perspectives. Business agility means fast reaction to change and the ability to rapidly implement changes. Business agility needs to be holistic in scope [125]. Technical agility refers to the ability to quickly change the type and flow of information within an organization within enterprise. Enterprises should balance IT to become better positioned and more agile [62, 186]. Services are the building Blocks of an agile enterprise [217]. Figure 9.10 presented in page 206 presents a proposed IT architecture to support web 2.0 empowered attended classes. Educational insti-

tutions differentiate between different components required to support the learning process and are familiar with some technologies like UMIS and LMS, however they are newly acquiring real-time communication services in its new informal form. IT architecture includes the following servers list:

- Firewall: system's entry point and responsible for providing security functions.
- Active Directory (LDAP): single log-in point for the entire system. Helps avoid the repeated log-in process between different applications and servers.
- Collaboration, Assessments, and Assignments: main components of the learning process that are maintained separately to provide greater flexibility and the ability to utilize different technologies.
- Students Data, Student Preferences, Learning Profiles: main asset of LMS that maintains students' data to be utilized by different applications.
- Course Specifications and Instructors Data: data about instructors and courses are stored to enable automation of the communications between different components.
- Real-time and Web 2.0 Communication Server: responsible for providing communications between instructors and students, and students and each other. It also manages online lecture files, desktop, text sharing, other activities and web 2.0 technologies that will be utilized in the informal learning process and informal feedback.
- Analyzer and Report Generator: responsible for analyzing gathered data and generating reports that are used by instructor to identify the sequence of lecture activities, students' learning profiles and preferences and feedback activities.
- Middleware: responsible for managing Quality of Service (QoS) and directing messages between different components of the systems.

Cloud computing is an emerging IT technology that will be utilized in different innovative ways in learning institutions. Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms and services are delivered on demand to external customers over the internet [133]. Cloud computing is a distributed computing paradigm that focuses on providing a wide range of users with distributed access to virtualized hardware and/or software infrastructure over the Internet. Drivers for Cloud computing adoption include: scalability, elasticity, virtualization, cost, mobility and risk reduction [188].

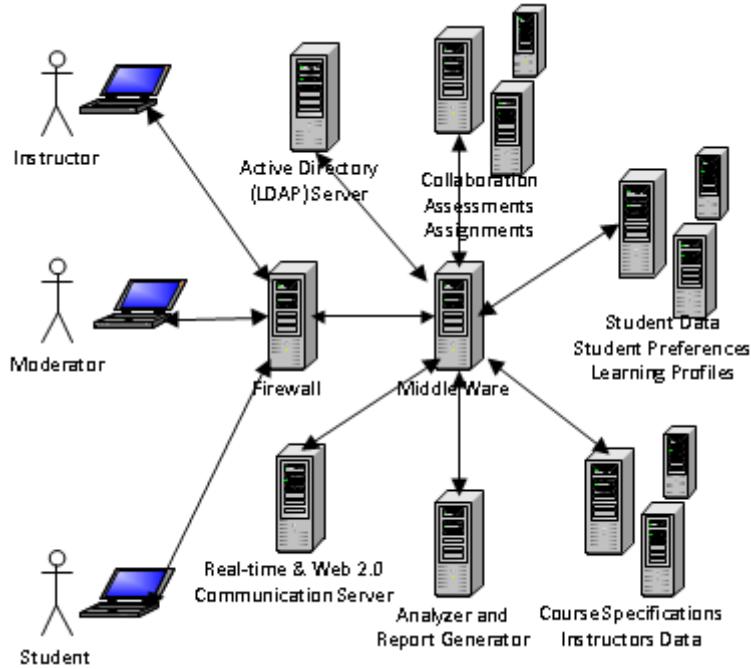


Figure 9.10: Proposed Adaptive Online Lecture Model IT Architecture

## 9.10 Future Research Directions

Future research directions in e-Learning include research in different technologies that shapes the presence of e-Learning 3.0. Technologies include: web 3D, semantic web, mining social networks, collaborative multimedia authoring, and NLP of real-time applications.

- **Web3D:** the next major internet wave [107], Web3D will deliver an interactive, immersive experience much richer than the static, text-oriented or even interactive graphical interfaces of today's web. In the new world of work that web3D will enable; people will be represented visually by avatars that can move in space, communicate with others and interact with objects and information. Yet Web3D won't leave the old world behind. It will integrate with the Web technologies we use today. Workers will use Web3D to teach and learn, innovate collaboratively, communicate and network, interact with and present information and manage real-world systems.
- **Semantic Web in e-Learning:** semantic web is a collection of standards, data structures and software that make the online experience more detailed, intelligent and in some cases, more intense. The goal of the Semantic web is to provide the capacity for computers to understand web content that exists on systems and servers across the internet, ultimately adding value to the content and opening rich new data, information, and knowledge frontiers. The potential of the Semantic web could actually revolutionize the learning experience and help it moving into web 3.0. Web 3.0 entitles better organized and more intelligent use of the knowledge available in the internet that will directly add

potential to the learning process and create new learning environments.

- Mining Social Networks: social networks include billions of terabytes of data, images, videos and many other file types that we need to analyze closely to come closer to newer generations, know how they utilize emerging technologies, how they think, feel and how they can utilize those technologies in a way that changes their lives to the better.
- Collaborative Multimedia Authoring: recorded lectures can be enhanced, updated and collaboratively worked on to present a rich multimedia repositories and learning content to students. An example of this collaboration is the one presented via We-Lcome to enable different students that can speak different languages from providing sub-titles in their languages to be added to the recorded lectures, so those lectures become available in different languages.
- NLP of Real-time Applications Generated Data: imagine an NLP engine that is capable of automatically analyzing students' feedback and providing both detailed and summary analytical reports about their perception of the lecture, and thoughts about the different activities conducted during that lecture till then. This feature can affect the flow of the lecture in a great manner. Such an engine will not enforce students to submit their updates specifically in a certain format, however it will work on their status updates on twitter, face book and whatever social networks and communication services they are using. This automated NLP is like an automatic analyzer of their thoughts and what goes in their minds about the lecture, so instructors can act fast and quickly in response.

## 9.11 SOA Selected Topics

### 9.11.1 WOA and Web 2.0 Effect on SOA

Web-oriented architecture (WOA), like SOA, has varying definitions. Some see WOA as a style of architecture and a substyle of SOA for Web-based applications. Others define WOA as the use of the REST style for building Web services using Web technologies like HTTP and XML documents. Web 2.0 appears to have greater agreement about its definition; and, not as the next version of World Wide Web, but as a platform where software applications are built upon the Web as opposed to upon the desktop. Web 2.0, like WOA, also uses models like REST. Web 2.0 includes mashups and Rich Internet Applications. It's unlikely we will ever see one way to build applications. WOA and Web 2.0 prescribe approaches for building applications, just as SOA. SOA focuses on a different problem domain than Web 2.0, WOA, or situational applications. SOA has a focus on enterprise applications for many of its value propositions—enterprises where there is a wilderness of heterogeneous systems, a forest of silos, and a vast frontier of proprietary databases and applications. Contrast this to Web-based applications, where standards prevail, hungry consumers are in abundance in the ecosystem, and openness is the law. Web 2.0 success stories are numerous; after all, designing and implementing Web-based

solutions is a lot different from designing enterprise solutions. The widespread adoption of application programming interfaces (APIs), services, and applications from Google, Amazon, eBay, and Twitter make many executives wonder why this cannot be replicated in their enterprise with SOA. This leads to inevitable comparisons of Web 2.0, WOA, and SOA. Executives wonder why they cannot build enterprise applications at the same speed of many Web-based applications. Of course, this is an apples versus oranges comparison. The adoption of SOA is not because organizations want to expose more services, make more APIs available, or offer more interfaces to the Web and its vast consumer base. When the focus is on creating a Web-based ecosystem of consumption, growing, and attracting partners, WOA and Web 2.0 are prudent choices. When the choice is more about making strategic assets of the enterprise more efficient and agile, the problem space changes; SOA is the prudent choice with an adoption that addresses more than service exposure.

## 9.12 Summary

The need for utilizing web 2.0 technologies came from the great demand from students to include technologies they utilize in different life aspects within learning and from the instructors' desire to enhance the learning process and overcome challenges. Web 2.0 technologies can be used in delivering learning materials, supporting assignments, enhancing assessments experiences and many other learning activities. In this chapter, taxonomy of different web 2.0 technologies was presented. This taxonomy highlights two main classes: in-lecture, and after-lecture. Web 2.0 technologies were mapped to each of those classes highlighting their powers and usage within learning institutions. Innovative real world examples of utilizing web 2.0 technologies in learning institutions were presented highlighting different advantages and challenges. Web 2.0 technologies utilization is a hot research point that has rich future research directions and touches the way learning institutions manage the learning process.

## 9.13 Review Questions

1. What is Web 2.0? Give some examples of Web 2.0 technologies.
2. How can we use Web 2.0 in e-Learning?
3. What are the most Web 2.0 tools applicable to use in-lecture?
4. What are the most Web 2.0 tools applicable to use after-lecture?
5. What is Web Oriented Architecture (WOA)?
6. What are WOA effect on SOA?

## 9.14 Exercises and Labs

At this moment, you shall be already familiar with JAVA programming language, the basic building blocks of the language, and how to write a simple GUI application that uses JAVA main building blocsks. Right now, you are requested to utilize some of the Web 2.0 provided APIs using JAVA Code. Web 2.0 provided online APIs include APIs presented by: Google, facebook, twitter, amazon, flicker, soundcloud, or any other API you can find and use.



# Chapter 10

## Evaluation of SOA in e-Learning

### 10.1 Introduction

e-Learning has been widely used to refer to computer based systems that not necessarily help main objectives of learning. UMIS is not the appropriate solution to support e-Learning; because it is not based on philosophical theories of learning. Also, LMS shall not focus intensively on managerial aspects of the university and focus on learning objectives. LMS is the software that automates the administration of training, as it registers users, tracks courses in a catalog, records data from learners, and provides reports to management. On the other hand, UMIS is the software that automates Students Affairs and Youth departments' activities in the university. They both complete each other, with the fact that UMIS advances LMS by decades.

Surveying Open source, free, and commercial LMSs yield the fact that LMSs tend to present enough managerial activities to the extent that makes them a standalone solution for universities. While Mansoura University has its own in house developed and deployed solution to manage university activities, adopting LMS as a replacement is not accepted. A solution to integrating UMIS and LMS via portals was presented in [176]. Though university portals can present a solution to the current situation, it is considered a missing solution because systems are not really integrated on application level but are integrated on user level. So, applications are still isolated islands that need to exchange data between each other. There is a real need of loose coupling in LMS and it will be of high importance in the near future [182].

This chapter goes as follow: section two presents a description of the current situation in Mansoura University, concluding with the importance of integrating both UMIS and LMS as the only suitable solution. Section three presents the evaluation of proposed solution from both IS quality parameters and pedagogical point of view. Section four concludes the chapter and presents future work. chapter ends by references.

Table 10.1: Advantages and Disadvantages of Parameterized Queries

Advantages	Disadvantages
Code reusability for different tables, objects, and databases	Hard to debug, and error management becomes more unreliable
Use variable names in statements that require constants	Temporary tables from the main statement cannot be used, unless they are global
Return ROWSETs with a variable number of columns	Algorithm time adds up to the time of dynamic SQL execution
Sorting by any column from a table	Difficulty of maintenance is difficult schema is hard coding in the dynamic code
Return ROWSETs with variable column names	Security can be comprised with SQL injection

## 10.2 Current Situation and Solution

Mansoura University runs its in house developed and deployed UMIS for more than a decade right now. UMIS has reached a stable and well mature state when compared to the newly introduced LMS in the university. To adopt LMS functionalities in the University; without making LMS and UMIS isolated islands, there are three deployment approaches to choose from:

- Approach One (LMS replaces UMIS): University will replace its UMIS with LMS that will perform both educational and managerial functions. Challenges are: UMIS has been customized to fit University rules and regulations and it is not easy to let it go simply, importing current data into LMS might be a challenge, and there is a risk of system inconvenience especially if LMS failed to provide managerial functionalities the way UMIS do. Table 10.2 presented at page 212 highlights advantages and disadvantages of utilizing approach 1 as a solution.
- Approach Two (UMIS takes over LMS): University will add learning functionalities to current UMIS. Though this approach overcomes shortages of previous one but still has some challenges to manage, like time to develop and add the new features while university can make use of advanced features available right now via LMS providers, and dealing with emerging standards. Table 10.2 presented at page 213 highlights advantages and disadvantages of approach two.
- Approach Three (Integrate UMIS and LMS): Neither LMS will replace UMIS nor UMIS will take over LMS. Both UMIS and LMS will exist and interoperate to enable university to achieve its managerial and educational tasks in efficient and effective manner. This alternative avoids all challenges presented in alternatives one and two. It avoids replacing UMIS risks, and provides flexibility to change LMS without affecting UMIS, and provides an immediate solution to make use of current available LMS functionalities.

Table 10.2: Advantages and Disadvantages of Stored Procedures

Advantages	Disadvantages
Improve security of database server	Slower than equivalent application code
Offer a mechanism to abstract data access routines which can improve code maintainability	Lead to application logic fragmentation between database and application tier
Reduce network traffic	Difficult to debug
Implement common routines and make them accessible from multiple applications	Most object-relational mapping systems cannot seamlessly exploit stored procedures
Database centric logic can be isolated	
Improve application portability by moving data logic into stored procedures	

Approach three presents the most suitable solution to the current situation with technical challenges to integrate different ISs side by side while keeping in mind that different ISs require different information presentation for the same entity. Student is an example of the entities that require different information presentations. Student data required by UMIS differ than student learning profile needed by LMS [72]. UMIS student record includes data like ID, Social Security Number (SSN), name, age, gender, address (street, city, country), email, username, password, Date of Birth (DOB), faculty, year, department while LMS student profile include data like a detailed records of what students have already learned at the level of learning objects, a learning preferences profile, and a development portfolio of transferable skills, a history of student interactions with tutors, peers, and other significant learning activities [113].

### 10.2.1 Current IT Infrastructure

Figure 10.1 presented in page 214 presents the current scenario in Mansoura University; where users can be classified into UMIS users; to handle non-educational activities, and LMS users; to handle educational activities. To generate a detailed report of the courses and the learning contents, the user has to go through UMIS to generate the courses IDs and LMS to acquire the learning content. UMIS and LMS are isolated islands connected only via users. The assessment experience by Faculty of Computers and Information Sciences in Mansoura University; highlighted many of the challenges exist in the University. Assessment Management System (AMS) team asked students to register explicitly for the AMS; and that is not accepted. A single-student registration is a must to satisfy all learning interactions with the University.

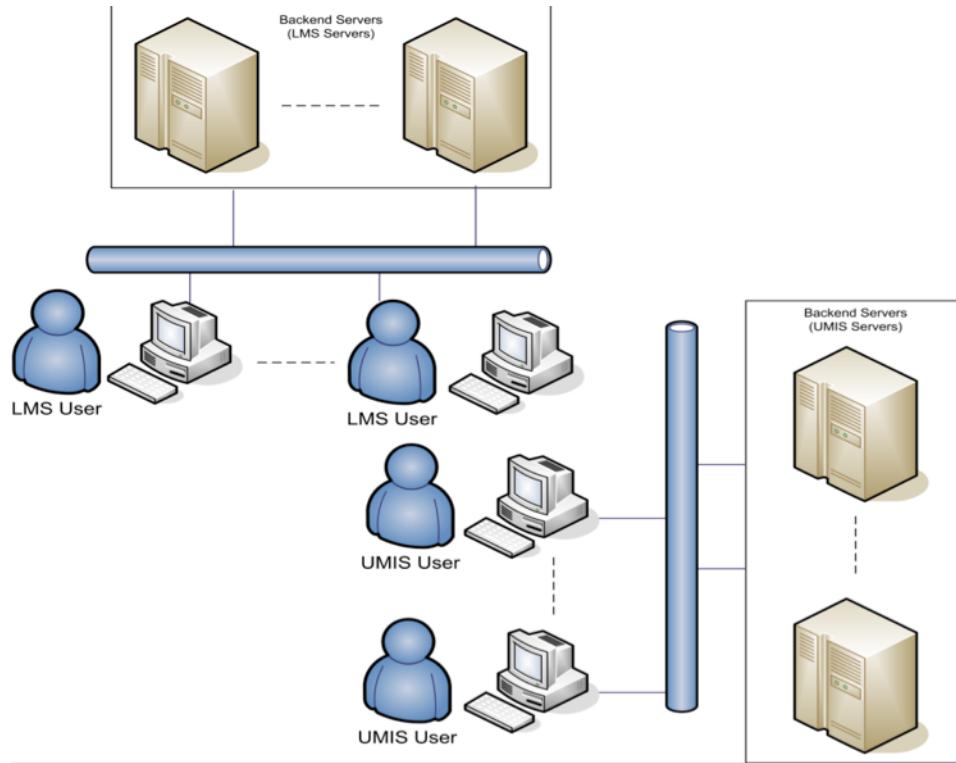


Figure 10.1: Current Scenario (Isolated UMIS and LMS integrated via users)

### 10.2.2 Proposed IT Modifications

Figure 10.2 presented in page 215 presents the proposed solution where a Service Layer shall be added in the middle area between UMIS, LMS, and users. Middle layer facilitates integration between different systems via Web services. Web services are relatively a new technology that have received wide acceptance as an important implementation of SOA [119]. Middle layer can provide portal(s) to unify users' interaction with different systems. More information about proposed CMS can be found in [113] and AMS can be found in [114].

## 10.3 Evaluation of Solution

Evaluation of proposed solution includes ISs, and Pedagogical perspectives.

- **ISs Evaluation:** It is evaluator's responsibility to determine the most valuable architectural aspects to be considered in the evaluation process [128]. ISs quality parameters evaluated in this chapter are: Network Performance, User Perceived Performance, and Functionality. SOA enhances system overall security, replace ability, testability, and both hardware and software scalability [123].
- **Pedagogical Evaluation:** Pedagogically, LMS shall enable universities and educational institutions to provide educational services in an easy, effective, and

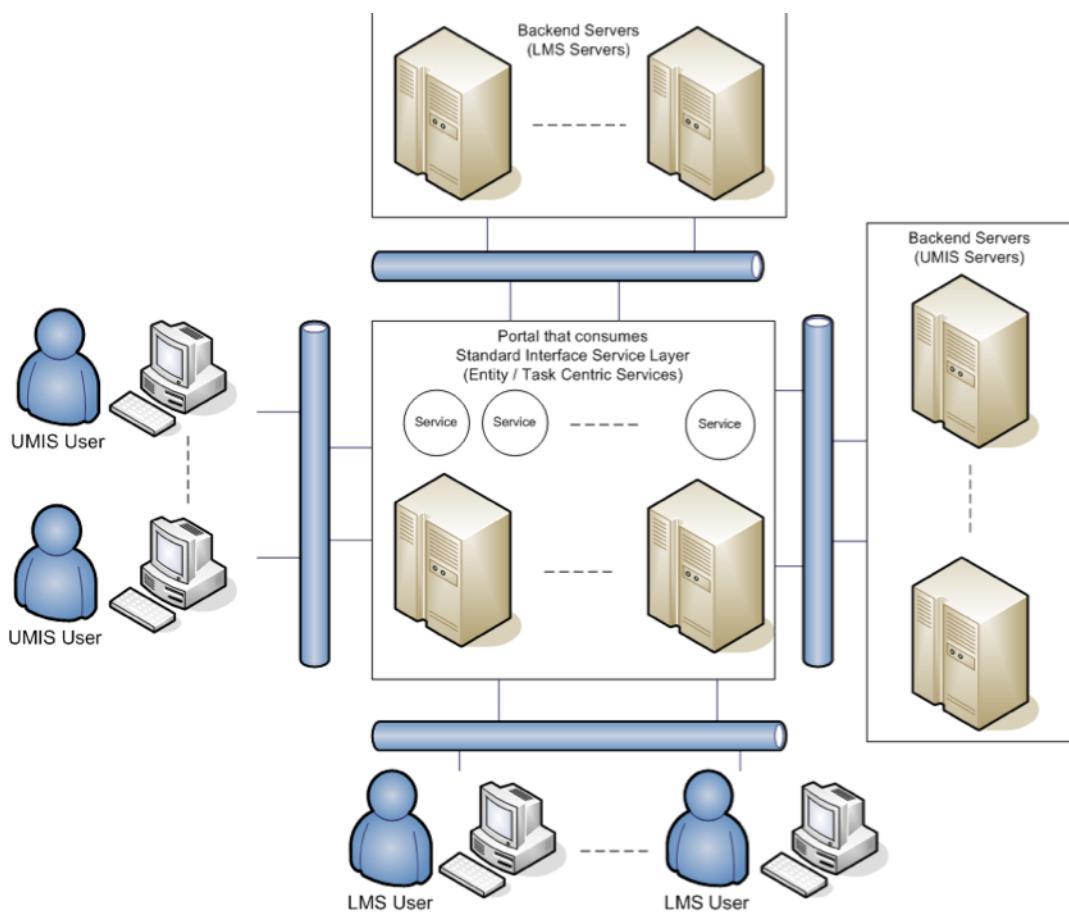


Figure 10.2: Presenting a Service Layer between UMIS, LMS, and Users

efficient manner. LMS providers and evaluators must be aware of pedagogical effects that will affect instructors and students. Current LMSs do not provide the required pedagogical effects [108]. One of the reasons is technology limitations. SOA helped LMS to come over some of the technology limitations.

### 10.3.1 IS Evaluation

From IS perspective, quality parameters like performance, integration and interoperability, compliance, security, maintainability, analyzability, decomposability and modularity, testability, portability via replaceability and scalability, simplicity, modifiability, and reusability shall be addressed. A Comparative performance analysis study is presented to test SOA based systems user-perceived performance against non-SOA based systems.

#### Analysis Tools

More than one analysis tool has been used while evaluating and analyzing proposed architectures performance. Analysis tools include, not only, Firebug; an add-on Firefox. Firebug integrates with Firefox to put a wealth of web development tools while the system developer browses web based systems. Editing, debugging, and monitoring browsed web pages are applicable. Main features of Firebug include the capability to: Inspect and edit HTML, Visualize CSS metrics, Monitor Network activity, Debug and profile JavaScript, measure performance and find bottlenecks fast, Quickly find errors, Explore the DOM, Execute JavaScript on the fly, and Log for JavaScript [20,21]. The most interesting feature of Firebug is the capability to watch the timeline of the web page unfolds. Firebug shows a bar for each file that displays when the file started and stopped loading relative to all the other files. Another interesting feature is Firebug's capability to Examine HTTP Headers that contain important information like the file mime type, web server type, caching directives, and cookie. HTTP headers have been used heavily in Services-based Architecture. Firebug was used heavily in the developing process based on presented functionalities, but in the analysis process it was used mainly to measure the timeline of page unfolds so measure the user perceived performance of the three LIS architectures.

#### Network Performance

SOA based systems relies heavily on messaging. It is clear that SOA based applications need to add extra headers to manage requests and responses in standard format. Header can be classified into two Static Header and Dynamic header. Static header is added once for every time the service is invoked while Dynamic header is added for every record contained within request or response message. Network Delay can be calculated using the formula [183]

Total Delay = Transmission delay + Propagation delay + Processing delay + Queuing delay Because processing and queuing delays are less than micro seconds, they are ignored and the formula becomes:

Total Delay = Transmission Delay + Propagation Delay Because the amount of transferred data within evaluation lab was static, Propagation delay is static, so the

Total Delay becomes the value of Transmission Delay.

$$\text{Transmission Delay} = (M + N - 1) L / R$$

Where:

- M = no. of communication links
- N = no of packets
- L = packet size
- R = Transmission Rate

Transmission delay is affected by file size (F). In the previous formula  $F = N * L$ . By analyzing data in the request and response messages, it is noticeable that there are three data categories:

- Static Header: This header occurs once for each service invocation no matter how many records in the request. There are 463 characters for one of the test headers.
- Dynamic Header (XML Tags): Those tags are the overload of requests and responses. Those tags are named by developer, so they are not static every time, but in the same test message there is 179 characters.
- Actual Data: Those are the record details to be inserted after invoking the Web service.

So, Total extra characters = Static Header + XML Tags. So, file size in SOA will be:  $FSOA = F + SH + RH * R$  Where:

- F = total data size without headers
- SH = Static Header
- RH = XML tags required to represent a single record
- R = no of records

This formula depicts that added extra headers differs according to the no. of records to be handled, and differs from an application to another (because the header used to represent an author might be different from the one used to represent a book) so network performance differs from an application to another. Figure 10.3 presented in page 218 presents a comparative network analysis between the services based system and non-services based system implemented in this case study. It is the system architect responsibility to decrease the transferred data over the network to the maximum extent (so decrease network delay) because it is noticeable that headers needed by SOA cannot be neglected easily.

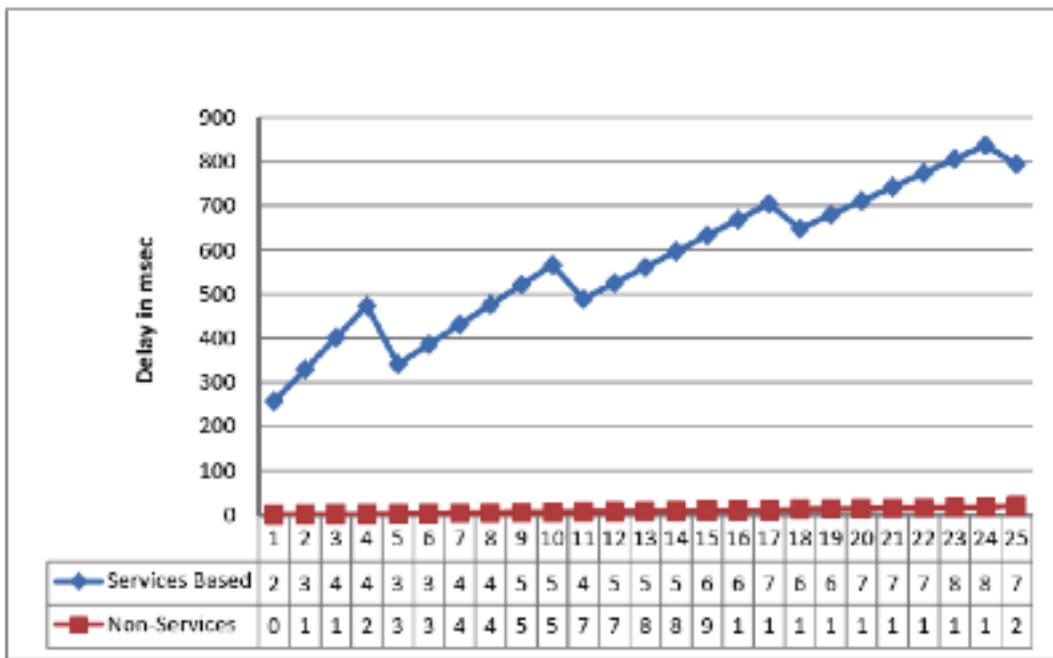


Figure 10.3: Comparative Network Analysis

### User Perceived Performance

Web services are the main SOA enabler. It is expected that utilizing Web services within an application will affect User Perceived Performance. In order to understand the extent to which Web services affect User Perceived Performance, three different Library Information Systems (LIS) were implemented tested against the same data samples.

The three different LISs are Parameterized Query based LIS, Stored Procedures based LIS, and Services based LIS. While Parameterized Query based LIS SQL statements exist within the web pages and accesses DB directly, Stored Procedures LIS highlights the separation between data layer and application layer by the presence of Stored Procedures as a middle layer in-between the portal and the databases.

The Portal consumes stored procedures to access the databases. The services based LIS presents the services layer in between the portal and DB layer to present a standard based interface layer that consumes stored procedures and available for portals.

Arithmetic Mean (Average) is the value obtained by dividing the set of quantities by the number of quantities in the set, so the Arithmetic Mean for each presented architecture regarding each presented process = Total time consumed to perform operation / Total no. of records. Another statistical measure that is used to evaluate the performance of presented architecture is Mode, which is the Most Repeated Value for each operation.

Figure 10.4 presented in page 219 illustrates statistics of the Insert process for the three implemented LISs. Table 10.3.1 presented in page 219 is a summary of the arithmetic mean and mode for each system. Services based LIS presented the

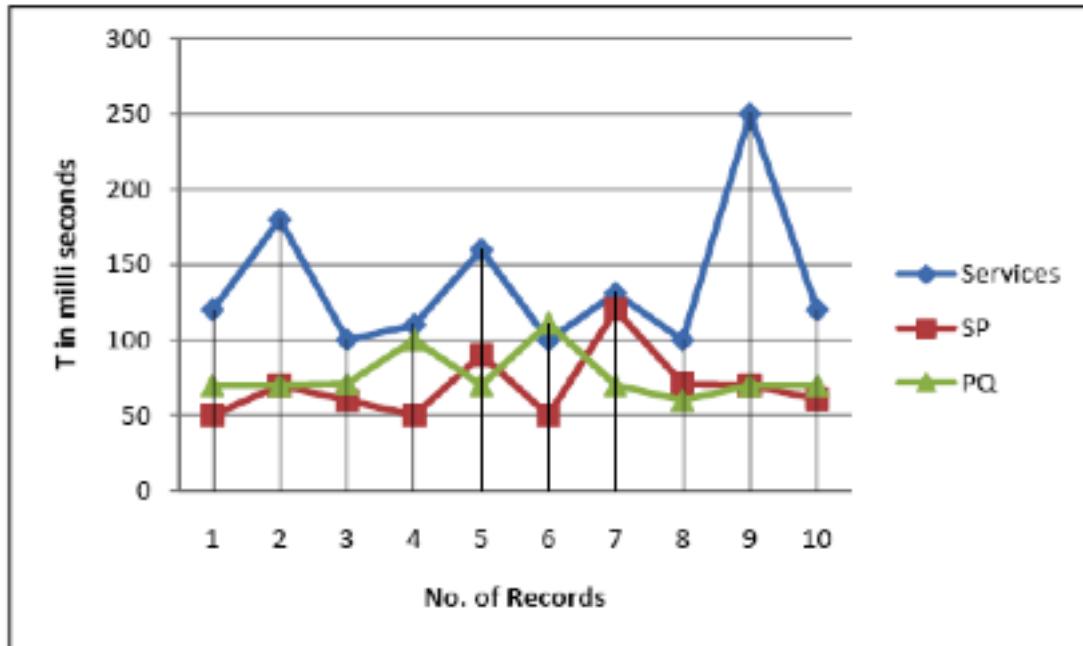


Figure 10.4: Insert Performance Measures of the three LIS Architectures

Table 10.3: Insert Operation Measurements Summary

Architecture	Arithmetic Mean	Mode
Service Based	137.1	100
Stored Procedure Based	69.2	50
Parameterized Query Based	76.2	70

highest arithmetic mean and mode values while Stored Procedure based LIS were the best for the insert operation.

Figure 10.5 presented in page 220 presents the statistics of the Update operation followed by table 10.3.1 presented in page 219 that displays summary of arithmetic mean and average of the same operation. Services based LIS is the lowest in performance compared to stored procedures and parameterized query based LISs. In the mean while, stored procedures based LIS is the best performance regarding the Update operation.

Figure 10.6 presented in page 221 presents statistics of the Select By ID operation of the three implemented LISs, followed by table 10.3.1 presented in page 220 that

Table 10.4: Update Operation Measurements Summary

Architecture	Arithmetic Mean	Mode
Service Based	141.2	120
Stored Procedure Based	103	80
Parameterized Query Based	109.3	70

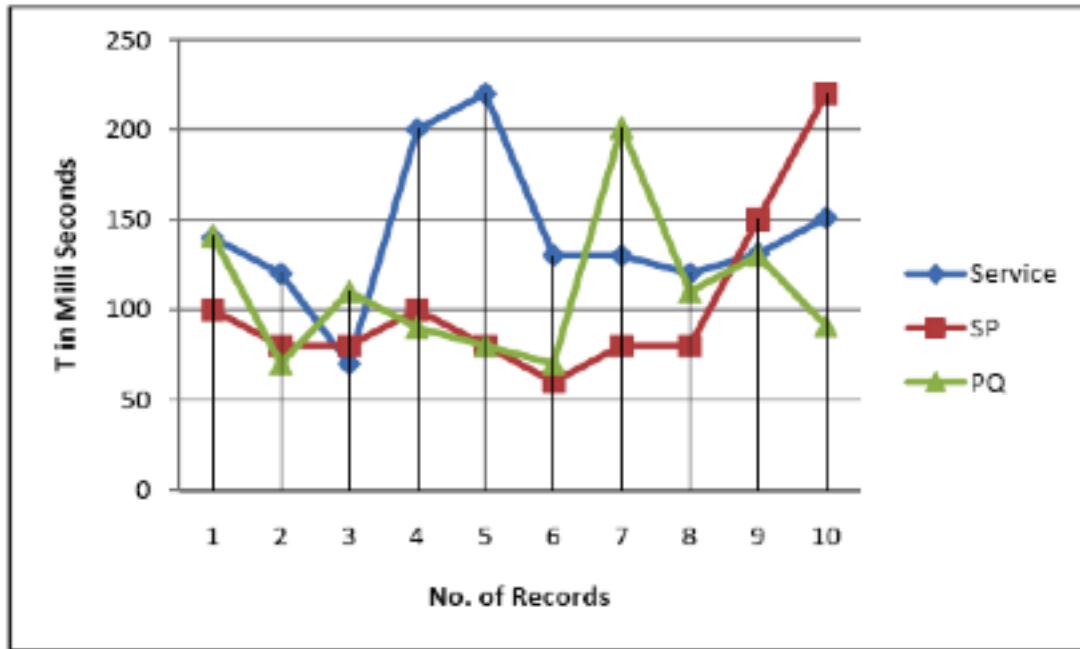


Figure 10.5: Update Performance Measures of the three LIS Architectures

Table 10.5: Select By ID Operation Measurements Summary

Architecture	Arithmetic Mean	Mode
Service Based	143	110
Stored Procedure Based	113	100
Parameterized Query Based	112.1	90

summarizes the arithmetic mean and mode. Services based LIS is the highest in ranges. While arithmetic mean and mode depicts that parameterized query based LIS performance is better than the stored procedure based LIS, it is noticed that parameterized query based architecture was highly affected by the amount of data retrieved, and its performance was not within small ranges; not like stored procedure one.

Figure 10.7 presented in page 221 presents the total amount of time required by each of the three LISs to retrieve all data stored in the DB, with no filter applied. Stored Procedures based LIS achieved the best time, services based architecture required time to retrieve the stored records exceeded the double time consumed by stored procedure based system, and parameterized query based system performance lies in between.

From the presented performance analysis and after evaluation of the three LISs, it is clear that the time consumed to perform the same operations using the services based LIS exceeds the time consumed to perform the same operation using either the stored procedure LIS, or the parameterized query one.

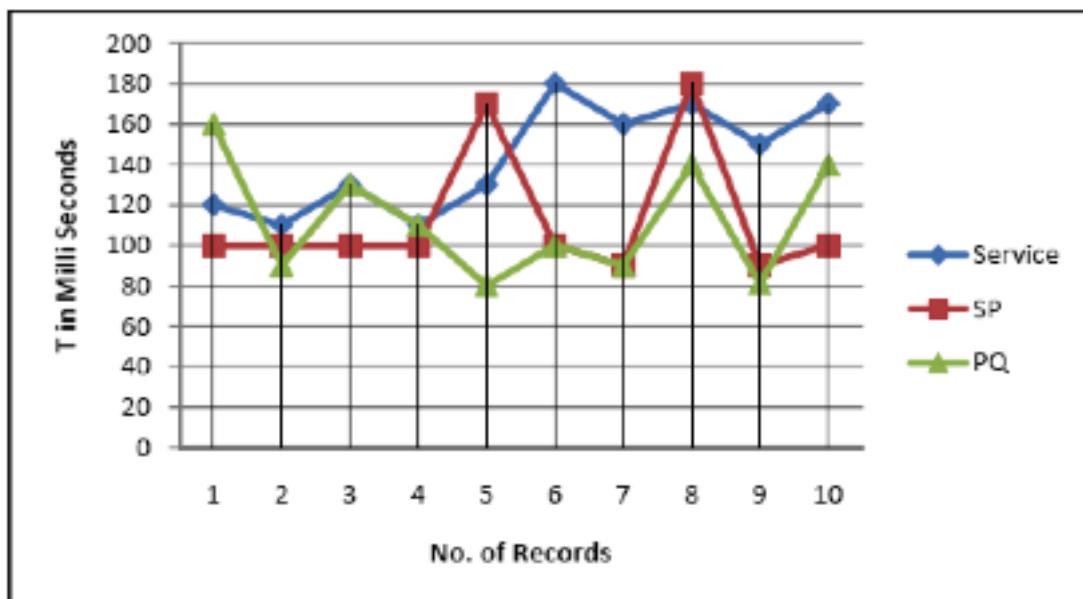


Figure 10.6: Select By ID Performance Measures

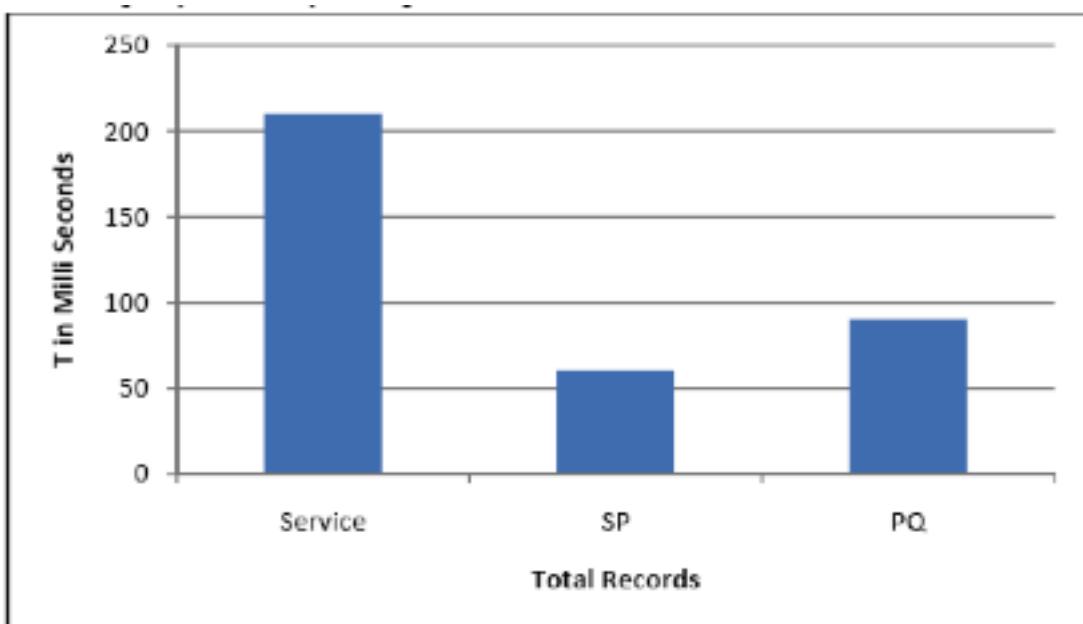


Figure 10.7: Display All Performance Analysis

## Functionality

Functionality evaluation focuses on the existence of a set of functions and their specified properties. Functions are those that satisfy stated or implied needs.

**Integration and Interoperability** A slight difference between integration and interoperability exists, where integration means the capability of an application to perform its tasks without the need from other application, but still the fact that results of performing those tasks needs to be shared with other applications. On the other hand, interoperability refers to the application need to communicate with other applications to achieve the task. It is really worth the effort to see that systems can share their effects within a single operation by integrating them on service level, not just on data or application level. Systems can share their effects within a single operation via service level integration. Assessment Management System (AMS) did not have to access Student Affairs IS DB tables to retrieve and update student table data, instead, it just invoked the UpdateStudent service exposed. AMS includes a Take Assessment Process that needs interoperability between AMS and external systems as presented in chapter [114]. Without this interoperability, Mobile assessment would not have taken place at all. SOA utilization in the system gave the system capability to expose standard interfaces that act like sockets to be plugged in to connect systems.

**Compliance** Compliance refers to the software adherence to application related standards of conventions or regulations in laws and similar prescriptions. Proposed SOA based system implemented the abstraction and separation of layer concerns. Orchestration and application services layer were separated to achieve the compliance goal. It was clear in the Assessment Management System when the learner had to take an assessment within 24 hours; this simple data was stored explicitly in the database and managed by the Manage Assessment service. When the system needs to change the period of which the student can take the assessment, changes are not made to the application, neither at application services layer, nor at orchestras.

**Security** Security refers to software ability to prevent unauthorized access, whether accidental or deliberate, to programs or data. Classical SOA; by default; utilizes certain security hints that enhance the overall system security, they include:

- The first step for security is to hide the existence of the service from the attackers, and this can happen easily via SOA. Middleware can be placed in the middle between consumer and the service, this middleware guides the request to the service while protecting it from attacks.
- Webservices make use of the enhancements performed by Web services manufacturers (Microsoft and IBM for example). Webservices 2.0 and Web Services Enhancements (WSE) are examples of by default acquired advantages of implementing Web services overtime.

- Web services act as an isolating layer of database; there is no direct access to the database available, and operations are stated within the service. The service requestor does not even need to have an account on the database.

**Maintainability** Maintainability refers to the ease and speed with which systems can be understood and modified.

**Analyzability, Decomposability, and Modularity** Analyzability is the effort needed for diagnosis of deficiencies or causes of failures or for identification of parts to be modified. Decomposability is the process of breaking down a system into its smaller components. These components may themselves be systems (sub-systems) and can be broken down into their components as well. This decomposition results in smaller and less complex pieces that are easier to understand than larger, complicated pieces. Decomposing a system allows analyst to focus on one part of the system, making it easier to think of how to modify that one part independent of the entire system. Modularity is a direct result of decomposability. It refers to dividing the system into chunks or modules of a relatively uniform size. Modules can represent a system simply, making it easier to understand and easier to redesign and rebuild. Composing systems into stand alone services makes it more analyzable, because it gets decomposable and modular. SOA based applications are easier to determine areas of failure.

**Testability** Software testing is the process used to measure the quality of developed software. Usually, quality is constrained to such topics as correctness, completeness, security, but can also include more technical requirements as described under the ISO standard ISO 9126, such as capability, reliability, efficiency, portability, maintainability, compatibility, and usability. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Levels of Testing:

- Unit Testing: tests the minimal software component, or module. Each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented. In an Object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. The minimal software component or module is: Service. Each service can be easily tested by itself to assure it performs the required functionality. WSDL provides a standard interface that enables service test even without the need to write a testing application. Applying SOA saves time of testing so, there can be completely new systems build without the need to apply unit testing, because they utilize previously tested services.
- Functional Testing: tests at any level (class, module, interface, or system) for proper functionality as defined in the specification. Functional test needs to be applied to units and system level. By applying functionality test to services via

easy by default designed services based unit test, it is clear that the available unit test facilitated the functional testing.

- **Integration Testing:** exposes defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system. Because proposed SOA based system utilizes Web services as the main SOA enabler, and because Web services utilizes standard protocols set, integration shortages and limitations were overcome. Integration requirements are not hard to design, implement, and tested incase it is based on Web services based SOA. Integration has turned to be the art of connecting the proper communication points to each other within systems.

**Portability** Portability is the ability of software to be transferred from one environment to another.

**Replaceability** It is the opportunity and effort using the software in the place of specified other software in the environment of that software. One of the main advantages of SOA is that it enables replaceability of services utilized from outside incase that the new utilized services maintain the same service interface. It will take just changing the address URI to the new address to utilize the new service.

**Scalability** Scalability is the ease with which a system or component can be modified to fit the problem area. Anarchic Scalability refers to the need for architectural elements to continue operating when they are subjected to an anticipated load, or when given malformed or maliciously constructed data, since they may be communicating with elements outside their organizational control. The architecture must be amenable to mechanisms that enhance scalability. Scalability can be achieved via hardware, or software, or both.

### Hardware Scalability

- **Modify Hardware Requirements:** Highly rated requested services can be separated on stand alone servers specialized for those services in order to scale the overall system. If this option is not applicable, the servers that hold the highly rated requested services can get hardware upgrades faster than the rest of the system, so the overall system performance can be enhanced.
- **Load Balancing:** Scalability can be achieved by Load Balancing Servers. Load Balancing Servers are the servers responsible for balancing loads between multiple servers hold the same required services. Figure 10.8 presented at page 225 depicts the classical Web service consumption scenario that is the one without a load balancing server as depicted in figure 10.9 presented at page 225. In figure 10.9, the load balancing server is responsible for balancing the requests between the two application servers that hold the same service.

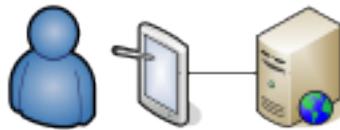


Figure 10.8: Classical Web service Consumption Scenario

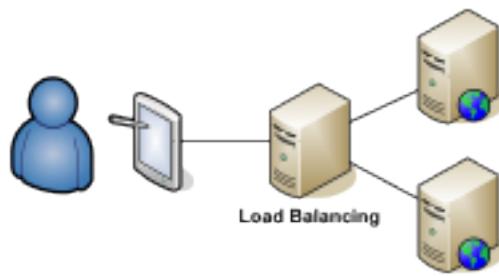


Figure 10.9: Load Balanced Web service Consumption Scenario

### Software Scalability

- System modularity: enables it to be enhanced service by service so the system lifecycle is enhanced on the software scalability level. Services that highly rated requested can be enhanced, and optimized, so the overall system will be enhanced.
- Web services based SOA: is immune to malicious software attacks because exchanged messages between requestors and services providers do not hold code to be executed, it only holds XML based data. Messages are not instructive, they are directive.

**Simplicity** The primary means by which architectural styles induce simplicity is by applying the principle of separation of concerns to the allocation of functionality within components. Separation of concerns into layers is what enables SOA to be simple. Layers enable the system to be understandable and less complex because by checking in which layer the service exists gives a general overview of the type of this service functionality.

**Modifiability** Modifiability is about the ease with which a change can be made to application architecture. A particular concern of network-based systems is dynamic modifiability, where the modification is made to a deployed application without stopping and restarting the entire system. Even if it were possible to build software system that perfectly matches the requirements of its users, those requirements will change over time just as society changes over time.

**Extensibility** Extensibility was tested in the proposed architecture by adding functionalities to the system that required adding new services that did not exist in the first place. Because services are loosely coupled, they can be added and consumed without affecting any other services. New added services can utilize existing services, consume databases and stored procedures, wrap legacy systems, and any functionality required without affecting the existing system.

**Reusability** Reusability is a property of application architecture if its components, connectors, or data elements can be reused, without modification, in other applications. It is the practice that a segment of source code can be used again to add new functionalities with slight or no modification. IEEE 90 defines it as the degree to which a software module or other work product can be used in more than one computing program or software system. Reusable module and classes reduce implementation time, increase the likelihood that prior testing and use has eliminated bugs and localizes code modification when a change in implementation is required. Reusability is achieved in the proposed architecture on two levels: Internal and External. Internal reusability refers to the application capability to use the same implemented functionality more than one time without modification. This happened for example with the Update functionality, where it consumed Delete and Insert functions. Besides, functions were not written every time from the start, on the other hand, they were recalled from the implemented service. External Reusability refers to the external systems that consumed the exposed internal services to achieve functionalities.

### 10.3.2 Pedagogical Evaluation

While integrating UMIS and LMS needed the SOA based system, pedagogical aspects are affected indirectly. No matter how mature LMSs become, there are more functionalities to be added.

- LMS was not affected; from the pedagogical point of view, by adopting SOA. SOA adoption facilitated integration of software agents within new proposed systems. Software agents have played, and still, important roles in e-Learning. Integrating software agents with Web services was presented successfully in proposed CMS [114] and Assessment Management System [114].
- Mobile assessment refers to the capability of conducting assessments via mobile devices. Mobile assessment relies on external services that are not part of the LMS. Integrating different external systems and services to be virtually part of the educational institution LMS is one of integration challenges. Mobile Learning (M-Learning) is an approach to e-Learning that simply utilizes mobile devices, yet it can also be viewed as a quite different learning experience. It is possible to force series of interactive SMS exchanges between learner and LMS to achieve completion of a task or goal. Learner will take part, and complete the task. M-learning has been used as a pre and/or post activity to other types of learning. Assessment for learning can be thought as one of the post

learning activity that can be achieved via mobile phones. Mobile Learning was successfully implemented and the main enabler was SOA adoption.

- Unlocking Course Repositories via Automating the Discovery, Downloading, and Paying of Shared Courses was one of the pedagogical advantages gained by adopting SOA [113]. One of the critical limitations of a newly established educational institution is the lack of available well prepared courses. It is more applicable to use widely available courses that might be higher in quality than preparing new courses. Current CMSs do not exploit courses shareability. A SOA based CMS is proposed to highlight automated discovering and importing of courses maintained and managed by external CMSs. Educational institutions can increase Return-On-Investment (ROI) by selling courses.
- Digital Library contents are available to all LMS components to utilize, search within, and enrich the learning activity with valuable contents without the need to adopt new systems. SOA facilitated the integration between LMS components and Digital Library solution.

## 10.4 SOA Selected Topics

### 10.4.1 SOA Governance

SOA involves people, process and technology, is cross-functional involving lines of business and IT. SOA governance extends all aspects of governance present in organizations necessary for creating specific outcomes (e.g., faster time to market for new products) using SOA. Governance activities focus on the outcomes an organization desires to effect via SOA adoption. For example, EA might establish IT principles, standards, and a common infrastructure, any or all of which might need changes (optimize) after SOA adoption. Such changes might include standardizing on an enterprise service bus and a registry. Investment processes for provisioning new applications and the prioritization process for those investments are examples of process changes that might occur upon the adoption of SOA because existing processes would need to accommodate the provisioning and sharing of services, not just applications or systems. The approach currently used by business and IT stakeholders for prioritizing spending for the next calendar year would change as a result of SOA adoption as organizations begin to adopt a shared-services approach in addition to their current practices related to applications. Organizations that have adopted COBIT (Control Objectives for Information and related Technology), which is an IT governance framework and toolset, or ITIL (Information Technology Infrastructure Library), which is a set of practices for IT service management, might make adjustments to accommodate specific SOA adoption goals in their organization. For example, ITIL provides an excellent list of practices that are largely IT focused. With SOA adoption, it would be useful to have business metrics captured reflecting whether the business process has met key performance indicators, or whether business and IT alignment is progressing according to defined metrics.

### 10.4.2 Need of SOA Governance

Organizations with effective EA perform four governance processes and they use these same processes with the adoption of SOA:

- **A vitality process** maintains the applicability and currency of the architecture reflecting the business and IT direction and strategy. Architectural principles are often used to guide the vitality process. SOA adoption adds new architectural principles such as, “Service models will be used to capture the enterprise portfolio of shared services.”
- **A compliance process** reviews and approves or rejects the design of a solution. This process can be performed at various points throughout the business and project life cycle. EA teams will review SOA artifacts for compliance to SOA reference architecture or standards as an example.
- **A communication process** educates and communicates the architecture across the organization. This includes ensuring easy access to and consumption of architectural information and assets. Implementing SOA requires communication decks, white papers and training materials be updated to reflect SOA adoption at the enterprise level. This includes standards, architectural guidance, reference architecture and refinements to any architectural building blocks necessary for SOA. item **An exception and appeals process** allows projects to appeal the noncompliance of a solution or design decision or investment with the board and perhaps be granted an exception. Project teams will need to make architectural decisions that in some cases may conflict with an architectural standard. The EA team will listen and grant exceptions as necessary for projects employing SOA.

SOA governance uses the exact same governance processes found with effective EA. Where effective IT governance is in place, SOA governance operates solely to enhance.

### 10.4.3 Proper Service Granularity

The proper granularity of a service or a right service or the optimal service is not about deciding on granularity but on identifying services.

## 10.5 Summary

SOA adoption within e-Learning in the form of UMIS and LMS presented IS advantages as well as pedagogical ones. It is clear that there is still more to be discovered and more advantages will become available upon adopting SOA in e-Learning. Evaluating IS quality parameters highlighted the fact that SOA based architectures most fit between ISs as an integration facilitator, but when used to build the entire applications, it affects the system performance greatly.

Pedagogically, SOA has helped e-Learning achieve more than one goal. One of the critical limitations of a newly established educational institution is the lack of

available well prepared courses. It is more applicable to use widely available courses that might be higher in quality than preparing new courses. Current LMS do not exploit courses shareability. Proposed SOA based LMS addressed this shortage, automated discovering and importing of courses maintained and managed by external LMSs. Proposed LMS facilitates integration between different LMSs in order to share resources of educational institutions. SOA facilitated integration between software agents that play an important role in educational institutions and Web services; that is the core of proposed SOA LMS. Also, integrating legacy systems and newly added systems is facilitated by SOA. M-Learning is enabled by proposed system. Mobile assessment is one of the M-Learning activities facilitated by proposed SOA based system. Mobile assessment relies on external services that are not part of the LMS. Integrating different external systems and services to be virtually part of the educational institution LMS is one of integration challenges. The capability to integrate the different digital library contents and make it available to different LMS components is a clear example of the SOA capabilities to integrate different and standalone system components and make them available to each other.

## 10.6 Review Questions

1. How can we evaluate SOA implementation?
2. Describe one of the scenarios that best suite SOA in enterprise integration.
3. How can we define proper SOA granularity?

## 10.7 Exercises and Labs

Right now, you shall be fluent in using JAVA and Webservices. Web 2.0 APIs you have been using are just Webservices. Now, it is time to try to build your own Webservices. Web services are distributed application components that are externally available. You can use them to integrate computer applications that are written in different languages and run on different platforms. Web services are language and platform independent because vendors have agreed on common web service standards. Several programming models are available to web service developers. These models fall into two categories, both supported by the Netbeans IDE. Follow the tutorials available online:

1. Netbeans tutorial titled Webservices Using Java and Netbeans presented at <http://netbeans.org/kb/docs/websvc/intro-ws.html>
2. Netbeans tutorial titled Getting Started with JAX-WS Web Services presented in <http://netbeans.org/kb/docs/websvc/jax-ws.html>

to build your own Webservices and to start consuming them. Isn't it interesting that you can build, deploy, and provide your own services to the world?



# Chapter 11

## Mobile Information Agents

### 11.1 Introduction

Agent is a software system that is situated in some environment and is capable of autonomous actions in this environment in order to meet its design objectives. Mobile agent is an executing program that can migrate during execution from machine to machine in a heterogeneous network. Major characteristics of agents are: autonomy, interactivity, reactivity, proactivity, and mobility [178, 255, 229]. Autonomy means that agent can make its own decisions. Reactivity means that agent can react to changes in environment. Proactivity means that agent should not wait for changes in the environment to take actions; instead it initiates its actions to start affecting the environment. Interactivity takes place when more than one agent cooperates to achieve certain task that is the summation of all sub tasks performed by all agents involved in the system. Mobility happens when agent can transport itself across different systems architectures and platforms. Information agent is an autonomous computational software entity that has access to one or more heterogeneous and geographically distributed information sources, and which is able to proactively acquire, mediate, and maintain relevant information on behalf of its users or other agents preferably just-in-time [178]. Mobile information agent is one of the evolving technologies that proved lot of success though lot of limitations. The employment of mobile agents has been particularly attractive in many applications like query processing within widely distributed heterogeneous environments (for retrieving information among different sites), network management, electronic commerce, mobile learning, and mobile computing [53, 228, 63, 168, 275]. Retrieving information from distributed DB systems is an essential requisite in nowadays distributed systems. Generally, it can be said that information retrieval and data collection are the most important requirements in the information technology.

Mobile agent systems provide new perspectives for distributed e-commerce applications. Sea-of-Data (SoD) applications are those that need to process huge quantities of distributed data. They present specific restrictions, which make mobile agent systems one of the most feasible technologies to implement them. A Mobile Agent-based Reservation System, MARS, was presented in [222]. MARS aimed to implement an online reservation system that uses mobile agents to determine what best option for the user is and reserve a ticket based on user preferences. An ar-

chitecture for implementing a multi-agent system within the context of a learning environment is introduced in [228]. The system implements website, DB, virtual classroom, and agents. IAs roles in e-Learning systems were presented. Two approaches for information retrieval based on mobile agents are presented in [229]. The first approach utilizes the mobility of agent for moving the query to the desired site where the data resided while the second one is based on reduction of the number of migrating agents. Mobile agents have been chosen for their intrinsic properties of autonomy, synchronicity, dynamicity of distribution, and adaptability to available system resources in VM (Virtual museum) by using the internet as the network infrastructure to make available heterogeneous museum information, from pictures, figures and maps to texts, audio and animated images [63]. Within any information network, users have many short term goals of information retrieval. Short term goals are constituted as finding information in as little time as possible based on a specific query. In recent years, peer-to-peer based information retrieval systems have raised significant interests among computer scientists. A peer-to-peer based information retrieval system consists of a set of nodes connected in a peer-to-peer fashion [168, 275]. Mobile agents can be used as a component in clustering architecture for routing in mobile ad hoc networks [100]. This chapter tends to build a MNAS, presenting different approaches for MNAS, and clarifies differences in performance among them. While doing so, advantages and disadvantages of mobile news agents will be discussed.

### 11.1.1 Importance of Mobile Information Agents

There are many arguments with mobile information agents that advance them to other technologies. Mobile information agents are able to travel autonomously through the internet. Such agents enable dynamic load balancing in large scale networks, reduction of data transfer among information servers, and migration of small business logic within medium-range corporate intranets on demand [178]. Mobile information agent advantages include that [255]:

1. It can invoke operations, retrieving and processing data, locally, eliminating the transfer of intermediate results, so, conserve bandwidth, reduce latencies, and responds to user actions rapidly.
2. It can continue execution even if the link between client and server goes down. This, because it executes on different machine(s) than the one initiated it, so, if the agent is already initiated, and is currently executing at another machine, and the link goes down, user, theoretically, should be able to retrieve results once the link is back. Besides, user gained time of offline processing.
3. It can choose different migration strategies depending on its task and current network conditions. This can be considered as an intelligent aspect of mobile information agents. Though distributed processing environments depend on certain pre-determined route, mobile information agents can be intelligent and overcome this constraint by adapting its execution plan according to task and/or network condition.

4. It allows traditional clients and servers to offload work to each other, and to change who offloads to whom according to the capabilities and current load of the client, server, and network.
5. It can serve in distributed environments. More than one mobile information agent can be initiated at the same time, and task can be subdivided on them. In this case, task becomes the summation of all subtasks that constitutes it. In this case, certain aspects of management should take place.

There has been a renewed interest in mobile agent technology due to the continued exponential growth of Internet applications, the establishment of open standards for these applications, as well as the semantic web developments. Building mobile agent systems is a challenge because few guidelines exist on the topic and also because exhaustive testing is virtually impossible. The many (rather recent) Internet services, applications and technologies (e.g. Web Services and XML) are playing a major role in the further exploration of mobile agents as a viable Internet technology [237]. Mobile agents are programs that can migrate from host to host in a network of computers, at times and to places of their own choice. Unlike applets, both the code and the execution state (heap and stack) move with the agent; unlike processes in process-migration systems, mobile agents move when and where they choose. A mobile-agent programmer, thus, has an option not available to the programmer of a traditional distributed application: to move the code to the data, rather than moving the data to the code. In many situations, moving the code may be faster, if the agent's state is smaller than the data that would be moved. Or, it may be more reliable, since the application is only vulnerable to network disconnection during the agent transfer, not during the interaction with the resource [180].

This chapter presents several approaches for implementing MNAS. It is a mobile information agent based system that aims to retrieve news to users. Here comes the name "mobile news agent", because mobile information agent retrieves news to users as an example of sort of information. The organization of this chapter is as follows: In section 2, problem definition is presented. Section 3 includes development framework, components, analysis, design, and implementation of MNAS. Section 4 focuses on different approaches that can be used in MNAS. Section 5 illustrates measuring performance of each approach in a comparative analysis. Section 6, illustrates the results of comparative analysis. Summary is presented in section 7.

### 11.1.2 Mobile Information Agents in Distributed Systems

Distributed Environments must provide minimal level of objectives to be flexible and effective. Though mobile information agents are not distributed DB systems, the fact that both distributed mobile information agents and distributed DB systems share the same concepts can not be ignored. The fundamental principal of a distributed DB system is to the user, system should look exactly like a non-distributed system and this implied twelve objectives [96]. Seven objectives of them will be applied to mobile information agents as distributed system and one more will be adapted to fit the system. Objectives like Fragmentation Independence, Distributed Query

Processing, Distributed Transaction Management, DBMS independence will not be discussed because they are pure DBMS topics. The eight objectives are:

1. Local autonomy: autonomy should be considered for both mobile information agent, and hosting site. Autonomy is a main characteristic of mobile information agents, and it should be built in by default. Though mobile information agents still can not start themselves, they still take decisions in performing actions autonomously. Mobile information agent decides when, where to move, and how to perform tasks at hosting sites. Mobile information agent is aware of different destinations, and it determines the suitable next destination. Mobile information agent recognizes different hosts' types, and detects suitable actions to be performed at each host. Autonomy of hosting site lies within host capability to authenticate agents in order to present a system security feature.
2. No reliance on a central site: There should be no central controlling site. Disadvantages of central sites include single point of failure, and bottle neck. Unfortunately this objective has to be broken because mobile information agent needs central sites to serve as distributed processing systems. mobile information agent should be able to clone, that is replicate itself to all destined sites, or spawn, that is generate children to perform subtasks at destined sites, and the sum of all mobile information agents subtasks satisfies the task. Generating replicas or children requires tracking, and summing subtasks. Tracking and summing can be performed either at client machine, or at a central dedicated server, or at any host.
3. Continuous operation: Mobile information agent's dead end destinations should be carefully managed. Dead ends might result due to communication disorder or site's failure. If a mobile information agent is destined to a dead end, this mobile information agent is lost. Dead ends can be managed either centrally or by all participating servers. Local autonomy property of hosts enable them to decide whether mobile news agent was serialized successfully to the desired host, and keeps trying if not. Parallel processing requires a central manager server to enable management to enforce continuous operation.
4. Location independence: user should not be aware of site's locations and open channels, so the vulnerability to outside attacks might be decreased as much as possible. This objective can be achieved using Server Aliases. Server aliases can be translated centrally or at all participating sites. Server aliases reflects news servers URIs stored in directory server DB.
5. Hardware independence: mobile information agent should be able to operate on different hardware platforms. This objective gives the mobile information agent ability to retrieve and process data with no limitations or restrictions. This can be achieved through hardware isolating layer, that is, operating system.
6. Operating system independence: mobile information agent should be able to run on different operating systems. The more operating systems mobile in-

formation agent can execute on, the more efficient and flexible the system is. Host independence can provide this objective.

7. Network independence: mobile information agent has to merge smoothly among sites at different networks. Operating Systems, at different levels, provide this functionality. The effect of different networks on mobile information agents is the responsibility of transport, network, and link layer components of network service model. It is not the responsibility of mobile information agent at all, though it is mentioned for the sake of completeness.
8. Host independence: this is the adapted objective to fit mobile ISs. Host is the always running application at each site that enables mobile information agent to perform its tasks. Host layer should act as an isolating layer from operating system, moreover, is able to run any received mobile information agent.

### 11.1.3 Problem Definition

Assume that we have a homogenous distributed DB system that includes horizontally fragmented DB relations. These relations contain data for news system. User can issue a request for news based on one or more preferred categories. These relations are fragmented among different ( $n$ ) sites. Relation schema is the following: TB = Project(ID, Category, DateTime, Content). There is a query submitted by a user at client machine requesting "Find the News for Sports and Business categories". A mobile news agent will be initialized at client machine, heads toward directory server to obtain information about news servers and collects news from them and return news to user. MNAS can perform this task using more than one approach. To decide which approach to implement, a comparative analysis between different approaches is needed.

## 11.2 Mobile News Agent System

### 11.2.1 Development Framework

Any programming language that supports connection and serialization among machines is suitable for building mobile information agents. Mobile agent systems were built using C/C++, Java, and many other programming languages. .Net framework is an emerging Microsoft technology that advanced programming in this century. Microsoft thinks of integration as the solution to lots of problems. Microsoft .net framework was presented in year 2000 to form a framework that enables executing any .net compiled code on any machine that has .net framework installed on it. Examples of those machines are: PDA; personal digital assistants, tablet PCs, smart phones, and computers. Besides, .net framework is not a software developing tool, neither a programming language. .Net framework supports more than 30 programming languages, including Delphi, Visual Java, and Visual C++ [189]. By using .net framework, mobile information agents make a step toward host independence objective. .Net framework is the isolating layer between different hardware, operating systems, and mobile information agents. .Net framework provides the most efficient

programming capabilities. Communication and Serialization of objects among machines can never be made simpler in programming and implementation than they are in .net. Choosing .net framework as the development environment was affected by two other factors: efficiency of Microsoft SQL Server 2005, and seamlessly efficiency of integration between it and .net framework. Microsoft SQL Server 2005 presents new vision for DB management systems that worth respect. Finally, .net framework deals with standards efficiently. News should be retrieved to users in a standard format allowing them to display results in a customizable environment. As news will be in the XML (eXtensible Markup Language: standard language for information interchange) format, XSL (eXtensible Stylesheet Language: standard language for formatting XML data) should be used to display news in the user's browser. The main goal was not to build a framework for mobile information agents. The goal was to present a step by step method to build a mobile information agent based system that utilizes mobile agents in several approaches, while doing so, advantages, disadvantages, restrictions, and limitations of mobile information agents and the several presented approaches will become clear.

### 11.2.2 MNAS Components

Figure 11.2.2 presented in page 237 depicts MNAS architecture. Managing this system efficiently indicates the need to present a central managing server. Though relying on central servers presents problems like lack of efficiency due to single point of failures and bottleneck access, distributed processing management addressed the need of central manager server. This dedicated central management server will be referred to as Directory Server. MNAS contains the following components:

1. Client machine, which is responsible for:
  - Help users to determine their most interest topics.
  - Initiate mobile news agent.
  - Serialize the initiated mobile news agent to the Directory Server.
  - Waits for the return of the mobile news agent to display its news content.
  - Serialize mobile news agent to its next destination.
2. Directory Server, is the MNAS central manager that is responsible for:
  - Authenticate mobile news agent. Authentication here is for machine, not users. Client machines add a username and a password that is authenticated at directory server to check if mobile news agent is initiated by authenticated machine. Another security level can be added at client machine, that allow only authenticated users to initiate mobile news agent. Because MNAS's data is not sensitive, and public, this security level is neglected.
  - Keep records for all system's News Server and MNAS categories of news. Mobile news agents can check those records and compare them to user's preferences to determine the agent route (list of news servers that will be visited).

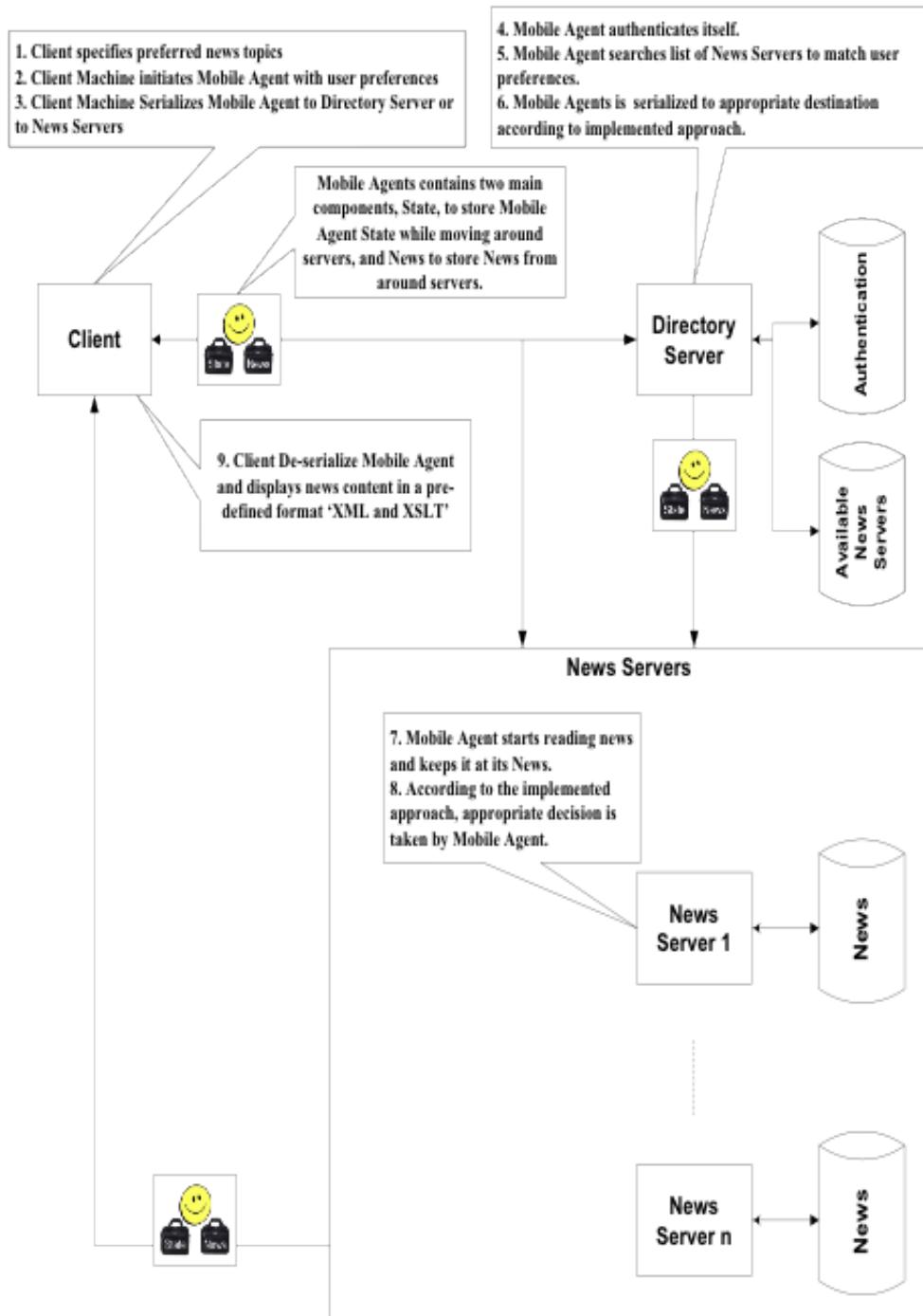


Figure 11.1: Mobile News Agent Architecture  
 Online: <http://zoom.it/ojubN>

- Prevent dead ends. Directory server keeps records for all news servers and their states. Mobile information agent checks whether news server is available or not. If news server is not available, it is discarded from route. Preventing dead ends enhances MNAS reliability.
  - Serialize mobile news agent to its next destination.
3. News Server, is the server that holds different news which mobile news agent can search. It does the following:
- Contain records of different news categories. Different news servers can contain different news. Records are stored in a standard XML format, so, they can be handled using XSL and displayed when mobile news agent returns to the client machine.
  - Serialize mobile news agent to its next destination.
4. DB, consists of the following tables:
- Authentication: used to enable directory server to authenticate originating client machine. Each client machine has an id and password, so, directory server can ensure that mobile news agent is initialized from an authorized machine. Authentication = Project(ID, Pwd).
  - Available News Server: contains available online news servers list and news categories on them that can be accessed. AvailableNewsServer = Project(Server URI, Category, Available).
  - News: where news content are available for query by mobile news agent. News = Project(News ID, Category, Content).

### 11.2.3 Advantages and Disadvantages of MNAS

Advantages of MNAS are:

- Offloading information processing from clients to servers.
- Ability of client to perform in disconnected environment; user initiates mobile news agent and disconnects. It waits till mobile news agent is ready to come back. This point helps user to overcome poor network conditions.

Disadvantages of MNAS are:

- Efficient and reliable offloading of information processing enforces high server(s) hardware and software requirements.
- Enormous amount of transferred data over network in certain approaches affects the overall performance of the system.

### 11.2.4 Technologies Used in MNAs

More than one paradigm can be used to implement mobile news agents. The Two main paradigms are mobile code, and mobile objects.

- **Mobile Code:** It is a technology that interprets data transported from one machine to another as code. Mobile code can be implemented in one of two ways: remote evaluation, and code on demand. The difference lies in which party initiates the action. With remote evaluation, machine sends code to another machine to be evaluated there, DB query is an example. In code on demand, machine asks for code so it can evaluate it, JAVA Applet is an example. The difficulty to move the code depends on two factors: the portability of the code that is exchanged and the difference between the two platforms that needs to exchange code. Portability means on what other platform the code can run without modification. Many mobile agents solutions tried to take advantage of platform independent code, so, this code can move freely by increasing portability.
- **Mobile Objects:** Objects are defined as computational entities that encapsulate some state, are able to perform actions or methods on this state, and communicate by message passing. Agent can be thought of as an extension of an object, where agent is an object with extended state and behavior. The major difference between agents and objects is that agents have clear intention and goal. Agents embody a stronger notion of autonomy than objects and, in particular, they decide for themselves whether or not to perform an action on request from another agent. While Agent is autonomous, interactive, reactive, and proactive, Object is only completely reactive and passive elements. Thus, usage of mobile agents is more suitable to systems requirements than objects. Mobile Objects are objects that can be programmatically moved among machines, and so, mobile agents will.

Figure 11.2 presented in page 240 uses a modified UML sequence diagram to illustrate serialization technique that is used to move mobile news agent between two hosts: host 1 and host 2. It will be moved between two running hosts that have an open channel between them. There are five steps that take place in serializing mobile news agent between the two hosts:

1. Host1 started the process by initiating the mobile news agent.
2. Mobile news agent is running, and it finally request to be moved to host2.
3. Host1 starts serializing mobile news agent through the open channel.
4. Host2 de-serializes mobile news agent.
5. After host2 de-serializes mobile news agent, host2 initiates it to continue processing.

In this chapter, MNAS depends heavily on serialization. MNAS considers mobile information agent as an extended mobile object that will be serialized from one machine to another, and will be executed there; as a mobile code.

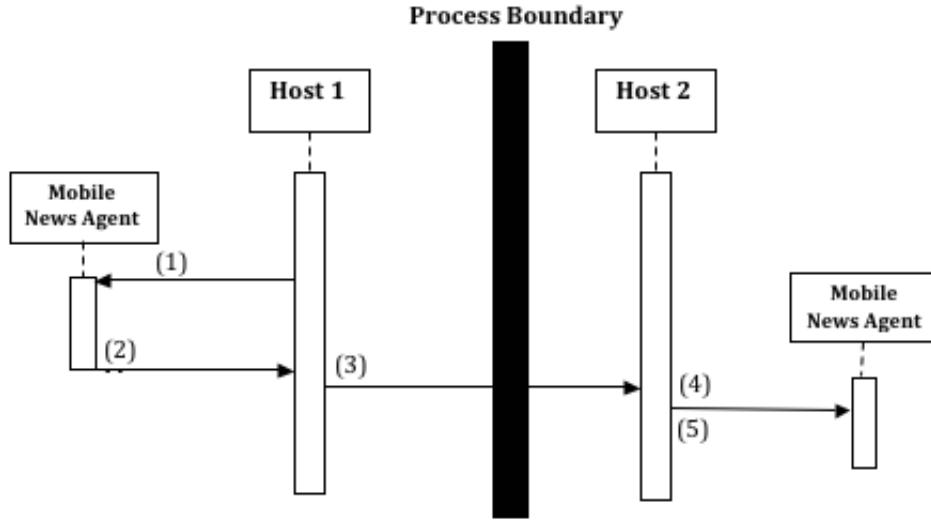


Figure 11.2: UML Sequence Diagram for Serializing Mobile Objects

### 11.3 Analysis and Design of MNAS

Logically, a mobile agent executes in a sequence of stage actions. Each stage action consists of potentially multiple operations [9]. Mobile news agent is the main concern in any mobile information based system. Mobile news agent like any object consists of methods and data. Methods that mobile news agent can perform are: Move, Initialize, Directory Server Search, Add News Server, News Server Search, and Add News Content. Figure 11.3 presented at page 241 shows the steps host follows to serialize mobile news agent to another host when Move method is invoked. Figure 11.4 presented in page 242 shows what mobile news agent performs after it is de-serialized and Initialize method is invoked by host. In Directory Server Search method, mobile news agent checks directory server DB for news servers that match user preferences. In Add News Server method, mobile news agent adds the matching news server URI to mobile news agent News Servers URI list. Mobile news agent invokes this method every time it finds a news server URI holds news contents that matches user preferences. News Server Search method is where mobile news agent searches news servers for news content that matches user preferences. Add News Content is the method that mobile news agent invokes every time it finds news content that matches user preferences.

Client methods are: Determine Categories, Manage Channel; either Open Channel or Close Channel, Serialize, De-Serialize, Apply XSL Format, Display Results, Host Agent, and Initialize Agent. Client machine needs to Determine Categories that user is most interested in to initiate a mobile news agent to retrieve news matches user categories of interest. Open channel is the method responsible for opening a communication channel with other hosts to be able to serialize/de-serialize a mobile news agent. Close channel takes place after client machine finish serialization/de-

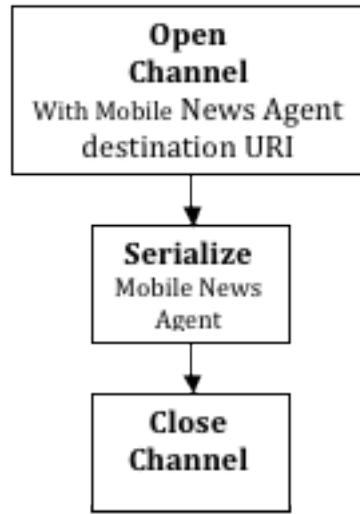


Figure 11.3: Hosts' Actions upon MNAS Move method invocation

serialization process. Mobile news agent content is in XML format to enforce standardization over contents in all news servers and mobile news agent. Client machine allows users to determine a preferred format to display results. This format is determined in XSL. XSL format is applied to retrieved news and display retrieved news when Display Results method is invoked. Client machines and any host within the MNAS have two common methods: Host Agent and Initialize Agent. Host Agent is the always running process that enables host to receive mobile news agent and move it to required hosts. Initialize Agent is the invoked method when an agent arrives. Initialize Agent invokes mobile news agent to continue processing. In this chapter, most of previously mentioned methods will be explained in detail. Figure 11.5 presented in page 243 shows MNAS class diagram of MNAS components.

Designing MNAS requires designing DB beside designing MNAS components. DB design will include directory server and news servers' databases. MNAS components include Client, Directory Server, News Server, and Mobile News Agent. MNAS design will start by DB design. Directory Server's DB contains one table that has three columns: Server URI, Category, and Available. Server URI represents the Unified Resource Identifier of the news server. URI consists of news server name concatenated with port number that represents TCP channel that news server is listening to. Category contains different news categories available at that news server. Both ServerURI and Category composes the primary key. Available reflects the state of the news server at certain time. So, if the news server is not available at that time, mobile news agent does not include it. News server's DB contains one table that has 4 columns: NewsID, Category, DateTime, and Content. News server manages its news using unique ID which is the primary key. NewsID is used to prevent duplication of retrieved news to the user. Category, DateTime, and Content



Figure 11.4: Initialization Actions of Mobile News Agent

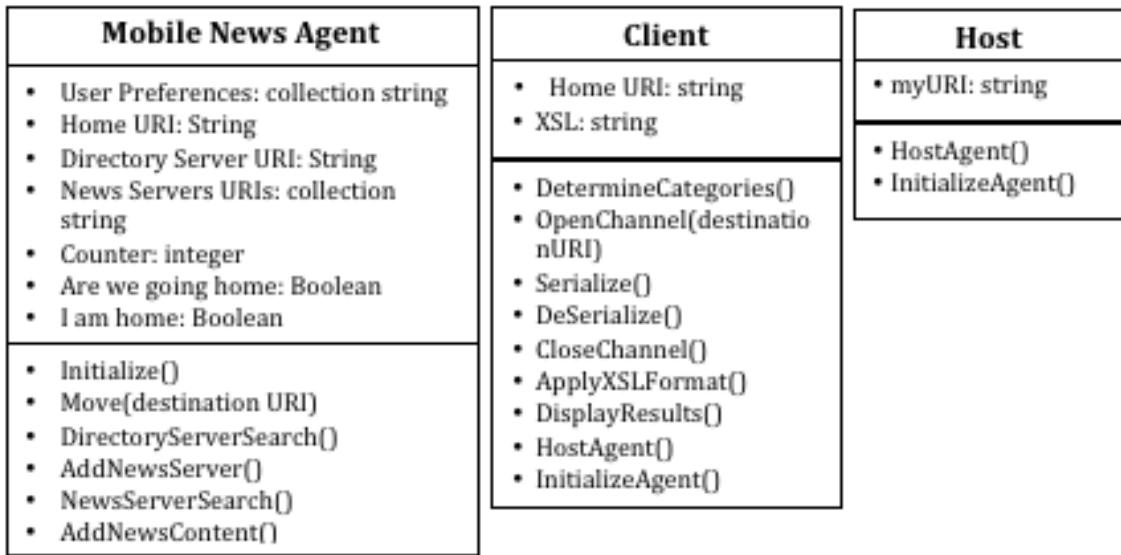


Figure 11.5: Mobile News Agent’s Class Diagrams

represents components of news that is retrieved to user.

## 11.4 MNAS Methods

This part describes methods that are available at the system and steps that each method implements.

### 11.4.1 Methods performed by Mobile News Agent

Mobile agent can perform six methods, they are:

1. Initialize: After Host initializes Mobile News Agent, it does the following:
  - Check where mobile news is at right now (Directory Server, News Server, Client)
  - Case: Directory Server
    - Read user preferences.
    - Invoke Directory Server Search method.
    - Invoke Add News Server method.
    - Invoke Move method with destination = first news server in queue.
  - Case: News Server
    - Read user preferences.
    - Invoke News Server Search method.
    - Invoke Add News Content method.
    - Invoke Move method with destination = next news server or Home if there is no more news servers in queue.

- Case: Home
    - Pass results to Host to display.
2. Directory Server Search It is the method that searches Directory Server for information about News Servers. It consists of the following steps
- Connect to Directory Server DB.
  - Search for news servers URIs records where news servers contain categories that match user preferences and those news servers are available.
  - Invoke Add News Server method for each matching news server.
3. Add News Server: It is the method that mobile news agent performs every time it finds a news server with content matching one of the user preferences. It goes like following:
- Check if the News Server to be added exists in queue or not, if yes, read next news server.
  - Add News Server URI to News Servers list of mobile news agent queue.
4. News Server Search: It is the method that searches News Server for news that match user preferences. It consists of the following steps:
- Connect to News Server DB.
  - Search for news records where news servers that match user preferences.
  - Invoke Add News method for each matching news record.
5. Add News Content: It is the method that mobile news agent performs every time it finds news with that match one of the user preferences. It goes like following
- Check if the News ID to be added exists in queue or not, if yes, read next news server.
  - Add News ID and content to News of mobile news agent.
6. Move This method was explained previously in details.

### 11.4.2 Methods performed by Hosts

There are two main methods that can be performed by any host within MNAS.

1. Host Agent: Every host has the ability to move mobile news agent to any destination mobile news agent asks for. When mobile news agent invokes host's move method, host performs the following steps
  - Read Destination URI.
  - Initializes a communication channel with destination.
  - Serialize Mobile News Agent.

- Closes communication channel.
2. Initialize Agent: Every host has the ability to initialize the mobile news agent after deserialization process completes and host receives mobile news agent. Initialization process consists of the following steps:
- Deserialize mobile news agent.
  - Load mobile news agent in memory. Additionally, client machine performs additional methods, like Display Results.
  - Display Results: It is the method that performed by client machine as a final step to display results of querying MNAS.
    - Read news contents from news bag.
    - Display news in web browser according to user preset XSL format.

## 11.5 Implementation Approaches of MNAS

MNAS can be implemented using several approaches. Three approaches are presented in this chapter:

1. Single Processing – Holding Data.
2. Single Processing – Delivering Data.
3. Distributed Processing.

### 11.5.1 Approach 1 'Single Processing – Holding Data'

Single processing refers to the system, not the mobile information agent itself. Mobile information agent can be multithreaded, but it can only be one mobile information agent doing some task at certain time, no cloning or spawning. Figure 11.6 presented in page 246 depicts the first approach. Client machine initiates mobile news agent, and then serializes it to the directory server where it accesses information regarding news servers, then directory server serializes mobile information agent to the next server, and so on. In this approach, routing algorithm depends on a list of sequential predefined servers identified by Directory Server. Information is carried within the mobile news agent along the route and is sent back to the client after journey ends. Simplicity is the advantage of this approach. Disadvantage is Lack of flexibility. Mobile news agent has a predefined list of news servers, so there are no reactivity to environmental changes, like varying network conditions.

### 11.5.2 Approach 2 'Single Processing – Delivering Data'

Figure 11.7 presented in page 247 depicts the second single processing based approach. It is similar to the first one in routing algorithm, but differs in handling retrieved information. This approach forces mobile news agent to send data, once they are obtained to client. This approach has main advantage of decreasing amount of

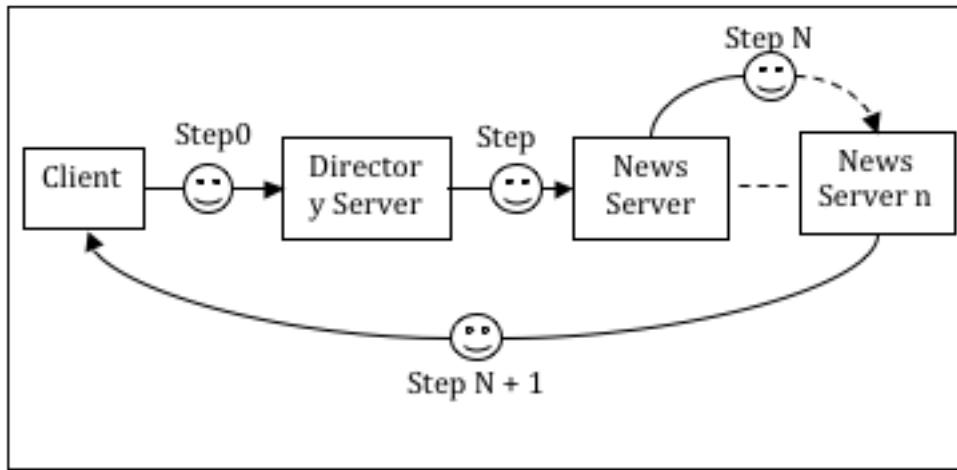


Figure 11.6: Mobile News Agent – Approach 1

transferred data over network because mobile news agent size does not increase proportionally per increment number of visited servers, thus, increasing performance. However, it does not overcome poor network performance condition, which is one important aspect of using mobile agent technology.

### 11.5.3 Approach 3 'Distributed Processing'

MNAS can perform as distributed processing systems by allowing more than one mobile news agent to execute at one or more hosts at the same time. This objective can be achieved by giving the mobile information agent the ability to clone itself; generate replicas, and/or spawn; generate children, of the mobile information agent to more than one place at the same time, each replica/child executes one or more specific task/subtask, and summation of all tasks/subtasks forms the required task. Mobile information agents can perform as distributed processing systems through more than one scenario. Figure 11.8 presented in page 248 depicts one of these scenarios where client machine initiates clones and/or children of the mobile information agent at the same time and starts distributing them with specified task(s)/subtask(s) to destinations. Managing this process can be performed either by client or directory server. In this approach, client machine will organize this process. Managerial tasks can take place, like:

- Subdivide task(s) into subtasks.
- Assign each child/clone one or more subtask(s).
- Follow children/clones state.
- Track children/clones.
- Track hosts of children/clones.

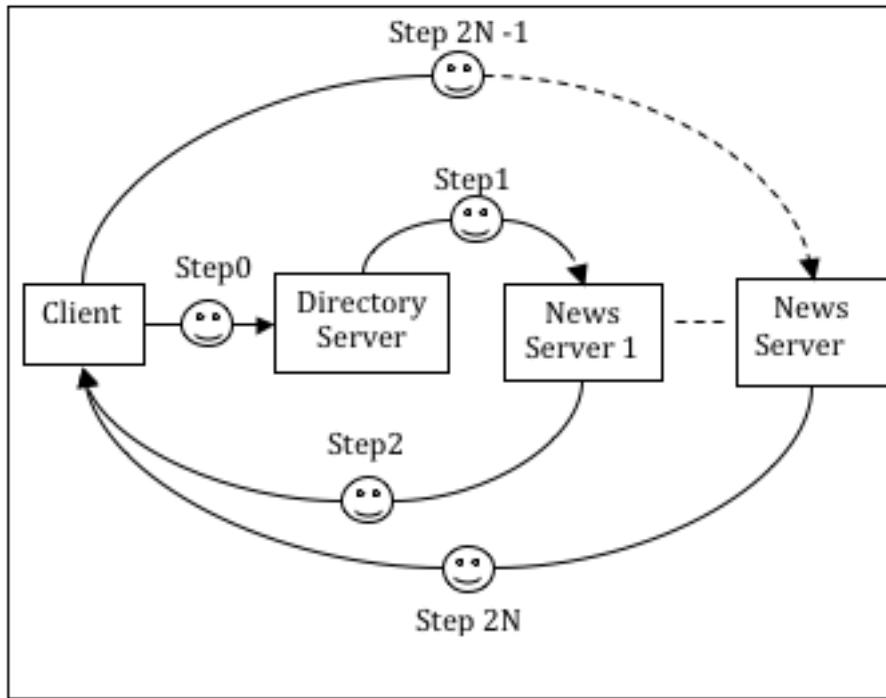


Figure 11.7: Mobile News Agent – Approach 2

Another approach that is similar to the one depicted in figure 11.8 presented in page 248 with a slight modification can be introduced. Client machine initiates mobile information agent and moves it to a centralized dedicated server that is responsible for performing the mentioned managerial tasks. This design can be edited to overcome problems of centralized server and make all hosts participating in the system act as servers. Generally, distributed processing increases performance by nearly  $n$  times, where  $n$  is the number of processing mobile information agents that are performing at one or more servers at the same time.

Every approach has its advantages and disadvantages. There is no standard approach that is applicable at all scenarios. It is the system's needs that force using one approach and not others. The depicted approaches are not the only available approaches. Enhancements can be added to the approaches to fit the system. Generally, there can be points of strength and weakness that advances one approach to another, among them:

- Amount of transferred intermediate results: the less the amount of transferred results, the better the approach.
- Dependency on central sites: the less the number of central sites, the better the approach, due to problems of central sites.
- Where information processing takes place: at central site, at any site, at all sites, or at client. Every trend has its advantages and disadvantages. This point is affected by hardware and network conditions.

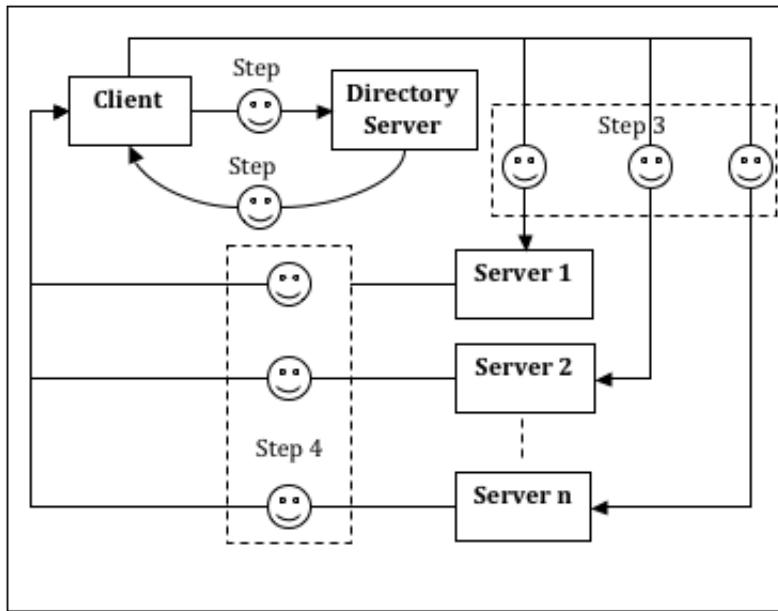


Figure 11.8: Parallel Processing Mobile News Agent

- Whether distributed processing is really needed or not: distributed processing is really complex to implement and manage efficiently. If system requirements can avoid this complexity, it should really do.

Comparative analysis between the three approaches is performed in this chapter to illustrate points of strength and weakness of each approach.

## 11.6 Performance Measurements

Performance needs to be estimated so approaches can be differentiated. Client sends a mobile news agent carrying a query "Find the news for sport and business categories" to the directory server. In approach 1, mobile news agent heads to directory server and query it for news servers URIs that contain news of sports and business categories. Next, mobile news agent moves toward first news server in the list, reads news, and moves to the next news server. Mobile news agent keeps moving while carrying news obtained from all news servers till it reaches final news server so mobile news agent returns home to the client machine. In approach 2, mobile news agent heads to directory server and query it for news servers URIs that contain news of sports and business categories. Next, mobile news agent moves toward first news server. After retrieving news from first news server, mobile news agent heads again to the client to deliver retrieved news. Then, mobile news agent moves to the next news server. Then, returns back to the client machine, and so on. In approach 3, mobile news agent heads to directory server and query it for news servers URIs that contain news of sports and business categories. Client retrieves list of news servers URIs and clones n mobile news agents, where n is the number of matching

news servers, and sends those n mobile news agents to their destinations. Results are submitted to the client machine where all processing of intermediate results is performed. In order to decide which approach is the best to execute the query, a cost model is used to estimate the total cost (response time) of each approach. The total time cost is the summation of input/output, CPU (processing) time, and communication time.

$$T_{Total} = T_{I/O} + T_{Processing} + T_C \quad (11.1)$$

$T_{I/O}$  and  $T_{Processing}$  will be neglected in this study compared to communication time. Hence,  $T_{Total}$  will be equal to  $T_C$  where

$$T_{Total} = T_C = Y + \frac{X}{DBR} \quad (11.2)$$

Where:

- $T_C$  : Communication time cost (sec.)
- Y: Communication Initialization Time and Client Processing Time (sec.)
- DBR: Data Bit Rate (Network Data Transmission speed: bytes/sec.)
- X: Amount of transmitted data (bytes).
- $X = (\text{Mobile News Agents Transfer}) + (\text{Messages Transfer}) + (\text{Tuples Transfer})$

The Communication time cost is written as

$T_C = \text{time (migrating mobile agents and messages)} + \text{time (Total Access Delay)} + \text{time (Transferred Tuples)}$

$$TC = C + \frac{S * N}{DBR} + D * N + \frac{T_n * T_S}{DBR} \quad (11.3)$$

Where

- N: number of servers will be visited
- S: size of mobile news agent
- $T_n$ : No. of final results tuples
- $T_S$ : tuple size
- D: access delay at each server, and
- C: processing time at client machine.

Directory Server will be excluded from calculation, because it will be visited by all mobile news agents approaches with same size and same number of times. Tuple size, tuple numbers, and number of visited sites are same at three approaches. For the first approach, size of mobile news agent varies as mobile news agent moves

between news servers. Mobile news agent is loaded with the transferred tuples so communication time cost of first approach becomes

$$TC = \frac{S + (Tn * Ts)}{DBR} + \sum_{i=1}^N Di + c \quad (11.4)$$

Approach 1 requires  $N+1$  steps to visit  $N$  sites and obtain data for using, noticing that  $S$  increases everytime mobile news agent leaves a news server, so, equation 11.4 presented in page 250 will be:

$$TC = \frac{(S1 + T_{s1} * T_{n1}) + (S2 + T_{s2} * T_{n2}) + \dots + (Si + T_{si} * T_{ni})}{DBR} + \sum Di + C \quad (11.5)$$

$$T_C = \frac{\sum Si + (T_{si} * T_{ni})}{DBR} + \sum Di + C \quad (11.6)$$

Where

$S1 = Ts1 * Tn1$ ,  $S2 = S1 + Ts2 * Tn2$ ,  $S3 = S2 + Ts3 * Tn3$  and so on.

And this arises because mobile news agent size increases every time mobile news agent leaves a news server holding the new data. While in approach 2, mobile news agent delivers news every time to the client, so, mobile news agent size is constant allover the network, but it takes  $2N$  to retrieve news to the client.  $N$  mobile news agents holding data and  $N$  with no data. So, equation 11.3 presented in page 249 can be modified to become:

$$TC = \frac{N * S + (S + T_{s1} * T_{n1}) + (S + T_{s2} * T_{n2}) + \dots + (S + T_{si} * T_{ni})}{DBR} + \sum Di + C * N \quad (11.7)$$

$$Tc = \frac{2N * S + \sum(T_{si} * T_{ni})}{DBR} + \sum Di + C * N \quad (11.8)$$

Where;  $S$  is constant and equals original mobile agent size.

For the third approach, the Communication time cost of the equation 11.3 presented in page 249 becomes:

$$T_C = \frac{(N + 1)S + \sum(T_{si} * T_{ni})}{DBR} + MAX(D_i + C) \quad (11.9)$$

Where processing is counted one time only, which is the MAX of all processing occurring at the same time.

## 11.7 Comments on Results

In this study, the effect of mobile agent size, access delay, and increment of DBR on communication time are presented. Comparing three approaches performance with each other helps highlighting advantages and disadvantages of each approach, but it is not the only factor to determine the suitability of an approach. The other factors to determine which approach is the best fit include availability of advanced programmers to build distributed processing systems, different network conditions, amount of data to be retrieved every time, amount of intermediate results, reliability,

and security. After testing three approaches in different scenarios, three figures are presented: Figure 11.9 presented in page 252 shows the effect of increasing mobile news agent size on communication time. It is noticeable that mobile news agent size increases tremendously every time a new news server is visited by mobile news agent in approach 1. Not like approaches 2 and 3, every time mobile news agent leaves news server, mobile news agent delivers acquired news to the client. The client delivery criteria makes mobile news agent almost constant in approaches 2 and 3, and increases proportionally with time in approach 1. Delivering acquired news every time to the client has a main disadvantage as shown in figure 11.10 presented in page 252. In approach 2, mobile news agent has to deliver its content to the client after visiting every news server, by calculating total access delay of the system implementing this approach, it is clear that by allowing mobile news agent to visit client every time it acquires new news content, total access delay increase proportionally. Total access delay is the access delay time of client machine multiplied by number of returned mobile news agent and every visited news server machine access delay. Figure 11.10 presented in page 252 shows that the more times mobile news agent returns to the client, the more total access delay will be. Total access delay of approach 3 is constant because MNAS initiates mobile news agents' number equivalent to number of news servers in the system, so, access delay takes place once at all news servers, and access delay is the maximum access delay of any of the access delays taking place at the servers. Figure 11.11 presented in page 252 shows the effect of increasing data bit rate 'DBR' of the network while retrieving same amount of news from same number of servers. The figure shows that the three approaches performance is getting better by increasing DBR which enables larger amount of data to be transferred over network. Approach1 is the most beneficial approach from increasing DBR due to enormous amount of transferred data over network. From the results we find that identifying the best agent approach depends on the network condition, number of servers, and agent size. The distributed processing approach is the most suitable one in this study. Presence of Directory Server as a central server wrecked MNAS reliability because Directory Server failure is a global MNAS failure. Influence of News Server being down is controlled by Directory Server. Directory Server determines failed News Servers and mobile news agents are autonomous enough not to move to failed news servers. MNAS reliability and availability can be increased by adding Directory Server clustering function.

## 11.8 SOA Selected Topics

### 11.8.1 Information Services

There is a category of services for which the creation uses information architecture and where a separation of information from applications and processes occurs. This type of service is referred to as an information service. Information services are a specific type of service that encapsulate the underlying information entities and their data sources. Information services can provide processing, consolidation, partitioning, cleansing, validation, and transformations necessary to fulfill a request to access, update, create, search, or validate information or data. Information services

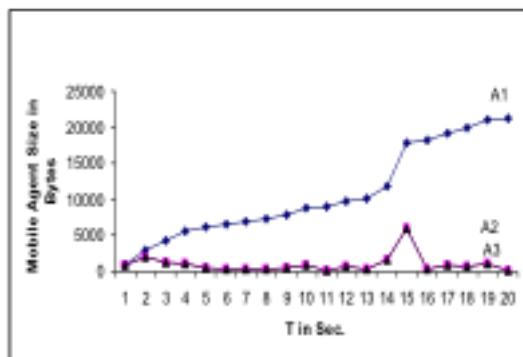


Figure 11.9: Effect of Mobile Agent Size on Tc

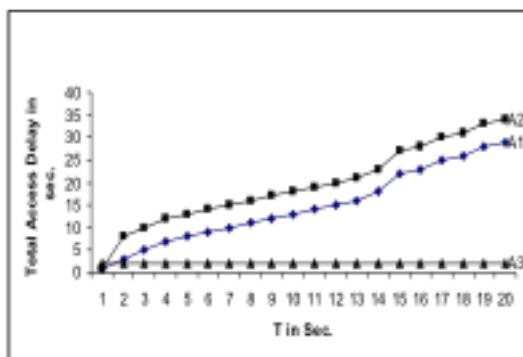


Figure 11.10: Effect of Total Access Delay on Tc

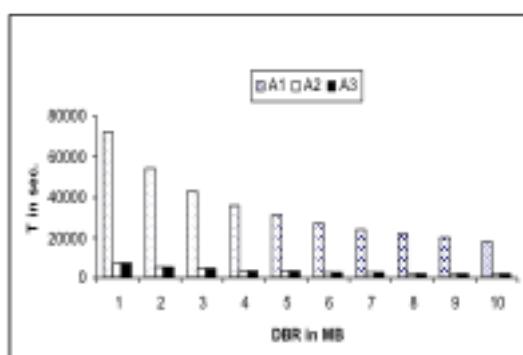


Figure 11.11: Effect of DBR incremental on TC

consolidate underlying data entities, accessing multiple and disparate information sources, transforming and consolidating the results into a format acceptable to the requesting party, the consumer. Information services are often used to address the heterogeneous nature of data sources and the fact that data sources often are replicated across several vertical systems. When presented with these two challenges, information services can be used to eliminate inconsistencies in business processes. Reusable, enterprise information can be viewed as sets of business entities standardized for reuse across the enterprise and used to create standard structures, semantics, and service contracts. The goal is to create a set of services that become the authoritative, unique, and consistent way to access the information.

### 11.8.2 Information Services Classification

Information services allow the consumer to retrieve information in a variety of formats using generic interfaces that increase the reusability of the service across heterogeneous platforms and vertical systems. Accomplishing this goal generally involves classifying information services as follows:

- **Integration services** that are responsible for data cleansing, data transformations, data consolidation, or federating data across multiple data sources to provide a consistent and authoritative data source. Integration services provide a service consumer access to consistent and integrated data that resides in heterogeneous sources. Integration services are most often read-only.
- **Data services** handle queries and the typical create, read, update, and delete (CRUD) functions. Data services access structured data as a service.
- **Content services** expose federated content, imaging data, archival records, or record management. Content services manage distributed and heterogeneous unstructured information so that a service consumer can access the content seamlessly.
- **Master data** services manage and expose trusted master data as services. Master data services provide accurate, consistent, and contextual access to master data from data residing in heterogeneous and inconsistent sources.
- **Analytical services** provide insights as data is sourced from demographic data stores, merchandise, contacts, transactions from data warehouses, to create analytical information. Analytical services provide access to analytic data out of raw heterogeneous structured and unstructured data. Analytical services are mostly read-only.

## 11.9 Summary

Mobile information agent technology provides advantages that are hard to find in one technology, but it should be used carefully to avoid complexities and overcome limitations. Mobile agent technology's disadvantages increases a lot if not implemented cautiously. Mobile information agent technology itself is not the advantage,

but it is how it is used that gives advantages. Mobile information agent technology fits greatly within situations where the following conditions are true:

1. Poor network conditions.
2. Enormous amount of intermediate results.
3. Security is a less concern.
4. Unique platform can be forced.
5. Distributed processing is an option.

Incase, one of these conditions does not apply, using mobile information agents might turn to be a disadvantage, because it takes away advantages, and leaves complexity. Mobile Information Agents can be implemented using more than one approach. Approaches are not equivalent, and needs to be considered thoroughly before choosing one to be implemented. The less dataflow in network, the higher performance system gets. Distributed processing is one solution that should be considered due to its high output ( $n$ ) times like single processing systems, where  $n$  is the number of executing mobile information agents.

Future work will include routing optimization of mobile news agent in MNAS. Mobile news agent does not have to visit news servers by their order list, instead, mobile news agent can determine its route using an optimization technique. Routing optimization might enhance overall system performance. Also, MNAS management should be enhanced to make MNAS more manageable, and reliable. What news servers are available now and what content they have is not enough information to manage a mobile agent based system. Future work will include also implementation of the same news agent system using Web services integrated with mobile agents technologies. Web services are an emerging technology that is also usable and platform independant. Web services are the main enabler of SOA (SOA). Future work will test whether SOA can be founded over integrated mobile agent's technology and Web services.

## 11.10 Review Questions

1. What are Information Agents?
2. What are Mobile Information Agents?
3. What are the advantages of utilizing Mobile Information Agents?
4. Design a MNAS highlighting different approaches, and how we can compare between them.

## 11.11 Exercises and Labs

Right now, you shall be fluent in using JAVA and Webservices. Web 2.0 APIs you have been using are just Webservices. Now, it is time to try to build your own Webservices. Web services are distributed application components that are externally available. You can use them to integrate computer applications that are written in different languages and run on different platforms. Web services are language and platform independent because vendors have agreed on common web service standards. Several programming models are available to web service developers. These models fall into two categories, both supported by the Netbeans IDE. Follow the tutorials available online:

1. Netbeans tutorial titled Webservices Using Java and Netbeans presented at <http://netbeans.org/kb/docs/websvc/intro-ws.html>
2. Netbeans tutorial titled Getting Started with JAX-WS Web Services presented in <http://netbeans.org/kb/docs/websvc/jax-ws.html>:

Build your own Webservices and to start consuming them. Isn't it interesting that you can build, deploy, and provide your own services to the world?



# **Chapter 12**

## **Mobiles in Community Services**

### **12.1 Introduction**

Mobile phones are efficient communication devices that make life easier. Whether locating a friend or following up with a new contact, mobile phones allow you to connect to people in any part of the world. New mobile-phone models are constantly engineered to meet the needs of consumers and now have multifunctional tools that may be useful in everyday life. Mobile phones' basic function, of course, is to allow talking to another person almost anywhere. Calls can be placed and messages can be received by simply pressing a person's name in contact list, eliminating the need for memorization or a separate address book. The ability of mobile phones to connect to people in other countries helps families and friends who are far from each other to stay in touch. Mobile phone applications extend the functionality of mobile phones. Word and spreadsheet processors are available, and those who may need on-the-go computing may find this a great benefit of mobile phones. Standard mobile phone applications include alarm clocks, calculators and even more, applications that provide web services, all of which surely help our productivity. This chapter presents one the Mobiles utilization in Community Service, that is Sanay3y "Craft Man".

#### **12.1.1 Need for The Project**

Since cellular phone networks have become key tool used by researchers as they try to locate people all over the world for several purposes. This chapter presents an overview on what have been accomplished in the field of creating an iPhone application whose aim is to provides a service that will help many people to have fast and easy access to required data that belongs to other people, in a mutual benefit manner. This project consisted mainly of three phases, the first of which was to design logos and application icons and use them to build an effective user interface in the second phase. The third phase involved writing a web service, which is to be called in the background, while the user is using their mobile phones.

### 12.1.2 Scope

In the midst of all the buzz surrounding Apple, iPhone application development trends are beginning to shift towards business use cases, according to a study by iPhoneAppQuotes.com. This project focuses on building a business iPhone App that tends to help many people in Egypt finding work, seeking better jobs, especially those who don't have daily jobs. The project has been divided into four main phases as follows:

1. Design logo and Application icons: involves designing logos that converge with the site ones and give the same impression, and designing icons that give an appropriate form for the application.
2. Application building: it entails building the application, which allow the user to add new “Sanay3y” or search about one. The interface was built using the designed logos and icon photos from the first phase build an Interactive interface.
3. Writing the web services: involves writing services that will return back with the desired output.

### 12.1.3 Aim

The overall aim of the project was to build an iPhone application which offers the same service provided by the sanay3y.com website, which is allowing the craftsmen to add their data in an easy access place so that every one can reach it when needed.

### 12.1.4 Objectives

The main objectives behind this project can be summarized as follows:

1. Showing all craftsmen currently available data
2. Provide advanced search about the craftsmen depends on their region or their profession.
3. Allowing craftsmen or any other one to add craftsmen data.

### 12.1.5 Integrating Web and Mobile

Web services are widely used in current web-based business applications, and form the basis of emerging frameworks in Grid computing. Current mobile agent systems implement similar (often proprietary) mechanisms for service description, invocation and discovery. Figure12.1 presented in page 259 presents the context diagram explaining this operations in our Sanay3y application. Table 12.1.5 presented in page 259 illustrates symbols used in the figure.

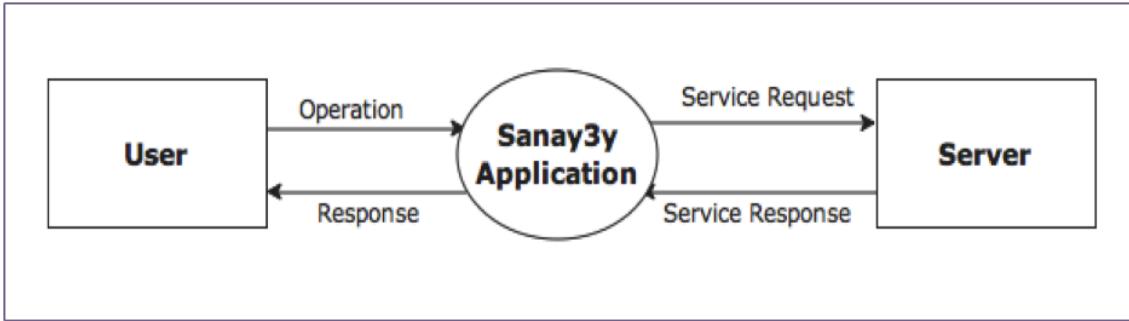


Figure 12.1: Sanay3y Context Diagram

Table 12.1: Tabular description of the Context Diagram

Symbol	Description
Square	Represents internal data source or data destination
Vowel	Indicates an internal intity or transfer mation process
Line	A line with an arrow indicates the direction of the flow of data

## 12.2 Mobile Components

This section will specify the requirements that customer expects from software to be constructed in sanay3y project, the user interaction with our application, and describes how the application is developed to achieve the desired output. Finally it goes on to review the coding of the basic application functions, leaving details of how the output looked like to the Application Outcome section.

### 12.2.1 User Requirements

An important and difficult step of designing a software product is determining what the customer actually wants it to do. This is because often the customer is not able to communicate the entirety of their needs and wants, and the information they provide may also be incomplete, inaccurate and self-conflicting. The responsibility of completely understanding what the customer wants then falls on the providers of the product. Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do .

In our project the customer requirements were to build an iPhone application that allows the user to add new member to the system with no registration policies, make an advanced search about system members depends on their region and profession, and view all the members of the system. Figure 12.2 presented in page 260 presents the use case diagram of the application which contains the basic user requirements. Table 12.2.1 presented in page ?? presents a tabular description of one the use cases presented in the figure.

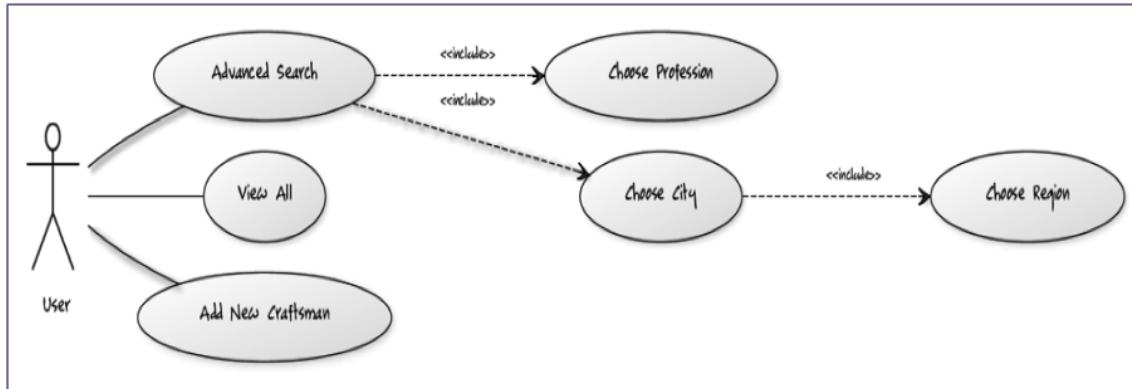


Figure 12.2: Sanay3y Use Case

Table 12.2: Tabular description of the ‘Data Transfer’ use case

<b>Actor</b>	User
<b>Description</b>	User may transfer data from the database that the application connected to by the web service, or add data of new craftsman to the database
<b>Response</b>	Confirmation that the craftsman data was successfully added
<b>Comments</b>	There is no security permission needed for any operation (add or search), it is a public service.

### 12.2.2 User Interaction with Sanay3y

The easiest way for users to interact with any program is through buttons which has predefined events that can be used to initiate actions. When the user clicks a button at run time, the button raises the click event. When an event occurs, controls run code in response to those events. the code that should run when the user clicks the button can be wrote by creating an event handler. An event handler is a method that executes when an event occurs. When a user clicks a button, the button’s click event has an event handler.

Figure 12.3 presented in page 261 presents activity diagram for the Add Craftman Process in Sanay3y application. In the diagram three activities are identified which are associated with conditions:

- Send add request by the user.
- Confirm Adding.
- View confirmation message that the user was successfully added.

After receiving the add request, condition checks are performed to check if the information is complete or not. After the information is confirmed the view confirmation message is performed and that is marked as the termination of the process.

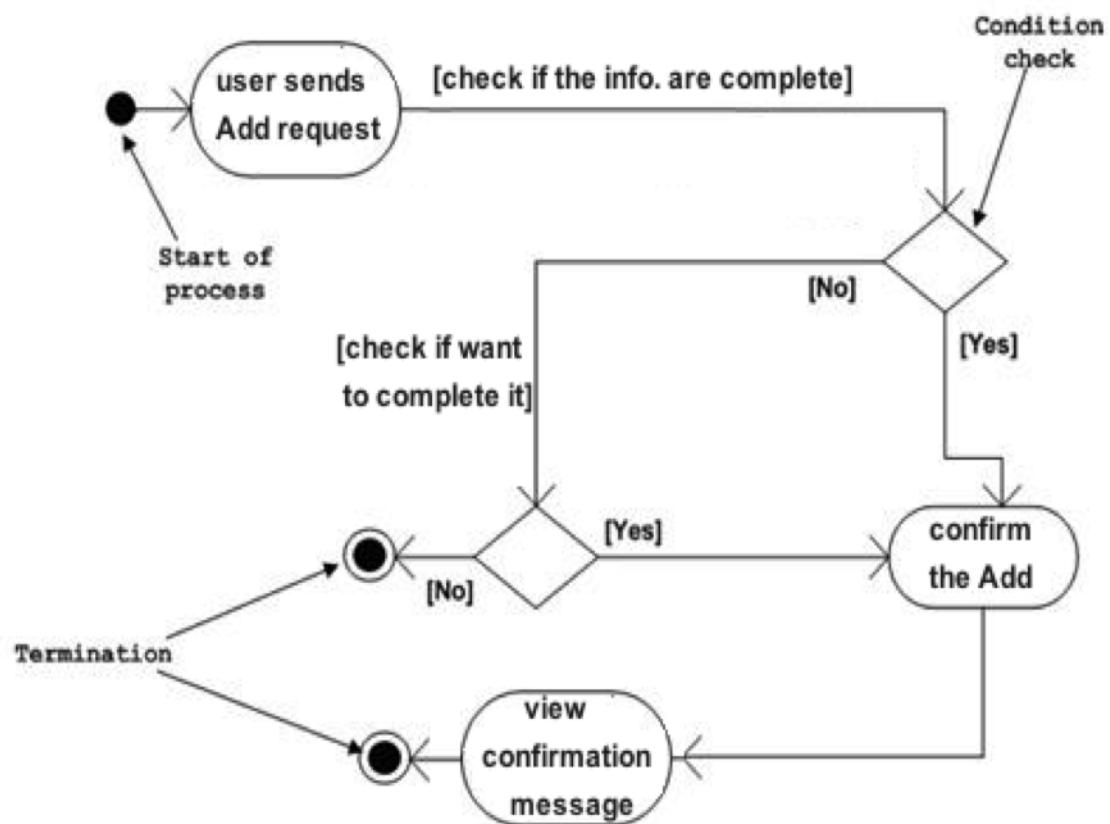


Figure 12.3: Add Craftman Process Activity diagram



Figure 12.4: Main Character in Sanay3y Application

### 12.3 Interface Design

A good mobile application starts life in the design stage. There are several aspects of the application that are formed at this stage, including among other things, layout, color, sound, content, functionality and maintainability. No reasonable person would start to build a house without designing it first, no reasonable application builders should begin construction without a design either.

Design makes things easier to use or to understand, and also serves as visual representation of your information. With the increasing information overload, our increasingly busy schedules, and plethora of apps and websites coming up everyday, good design can be an important differentiator. In an interview with Fortune Magazine, Steve Jobs said: “Design is the fundamental soul of a human-made creation that ends up expressing itself in successive outer layers of the product.”. When you start out with a beautiful and awe inspiring wireframe or prototype, your expectations about that product and everything associated with it is expected to also be beautiful and awe inspiring. If the initial expectations for a mobile application are low by bad or no design, then bad practices seem to find their way into the code. Figure 12.4 presented in page 262 presents the main character used in Sanay3y Application. Figure 12.5 presented in page 263 presents the Sanay3y Application general designs. Figure 12.6 presented in page 263 presents the Sanay3y Application layout. Figure 12.7 presented in page 264 presents the Real Application design that maps to previous figures. Figure 12.8 presented in page 264 presents the User Interaction and the final outcome of the Sanay3y Application.

### 12.4 Integrating Web and Mobile

This section describes the importance of integration between web and mobile and presents the integration method used in Sanay3y application. It starts by describing



Figure 12.5: Sanay3y Application General Designs

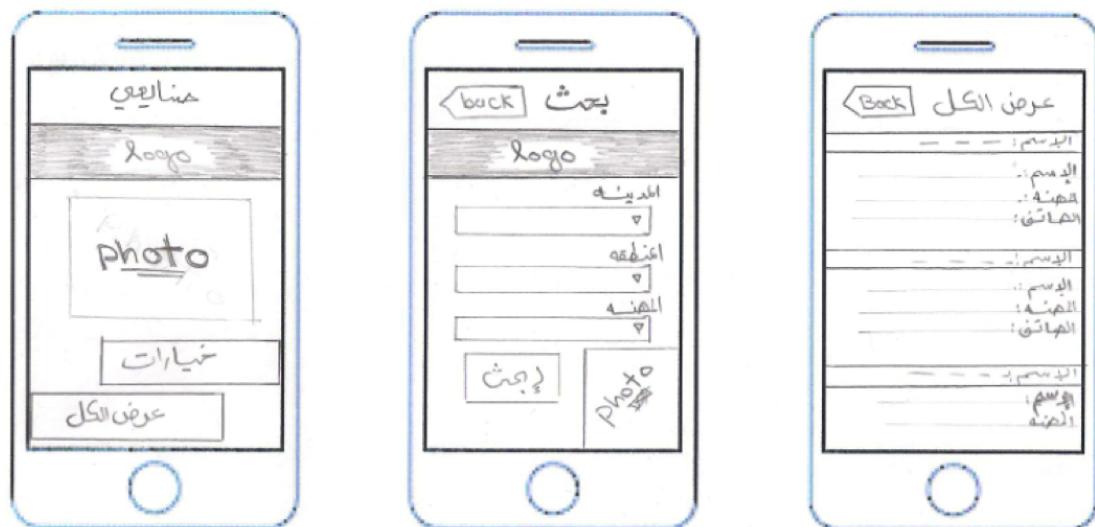


Figure 12.6: Sanay3y Application Layout

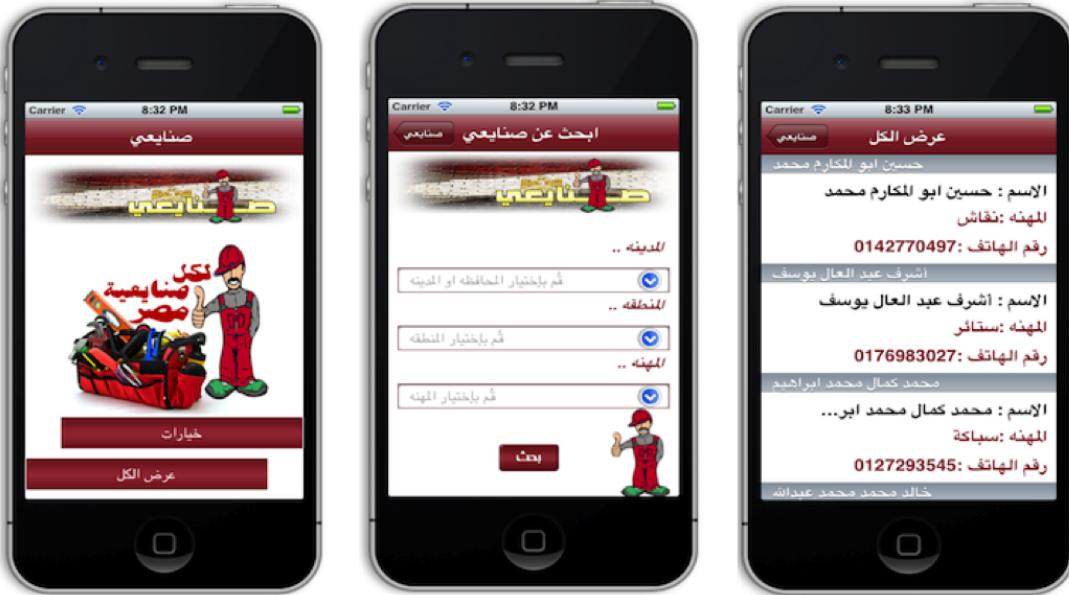


Figure 12.7: Real Application Design



Figure 12.8: Sanay3y Application in Action

the integration method followed, then goes on to describe the results of this method and finally moves to conclusion.

For the systems analyst, the most difficult job in specifying a system is often that of determining what is part of that system and what is not. Whatever system you develop, no matter how ambitious, no matter how grandiose, it will be part of an even larger system: even if our job were to “design the world,” we would have to recognize that the world is only a part of the solar system, which is part of a small, obscure galaxy, which is (ultimately) part of the universe.

Thus, the first major model that must be developed as a systems analyst is one that does nothing more than define the interfaces between the system and the rest of the universe, that is, the environment. For obvious reasons, this model is known as the environmental model. It models the outside of the system; the model of the inside of the system, known as the behavioral model.

In addition to determining what is inside the system and what is outside the system (which we accomplish by defining the boundary between the system and the environment), it is also critically important to define the interfaces between the system and the environment. We need to know what information comes into the system from the external environment, and we must know what information the system produces as an output to be delivered to the external environment.

Of course, inputs and outputs are not produced at random: no information system gobbles up all available data from the universe, nor does any realistic system spew output at random for consumption by the external environment. The systems we build are rational, purposeful systems; specifically, they produce outputs as a response to an event, or a stimulus, in the environment. Thus, another critical aspect of the environmental model is that of identifying the events occurring in the environment to which the system must respond. Not all events — after all, the environment, in its totality, generates an infinite number of events! We are only concerned with those events which (1) occur in the external environment, and (2) require a response from the system. Note that the boundary between a system and its environment, as illustrated in Figure 12.9 presented in page 266 is arbitrary. It may be made by management decree, or as the result of political negotiation, or simply by accident. And it is something that the systems analyst usually has some opportunity to influence.

### 12.4.1 Integration

Application integration is a strategic approach to binding many information systems together. The need to integrate applications has been a requirement since business process automation was presented. Application integration can be one of the following forms: Internal Application Integration, External Application Integration, or Enterprise Application Integration (EAI). Internal application integration techniques integrate organizational applications with each other, while external application integration techniques consider integrating organizational applications with applications outside organizational boundaries. EAI platforms centrally integrate heterogeneous system landscapes on process, method and data level. Internal and External Application Integration present traditional integration levels, where EAI

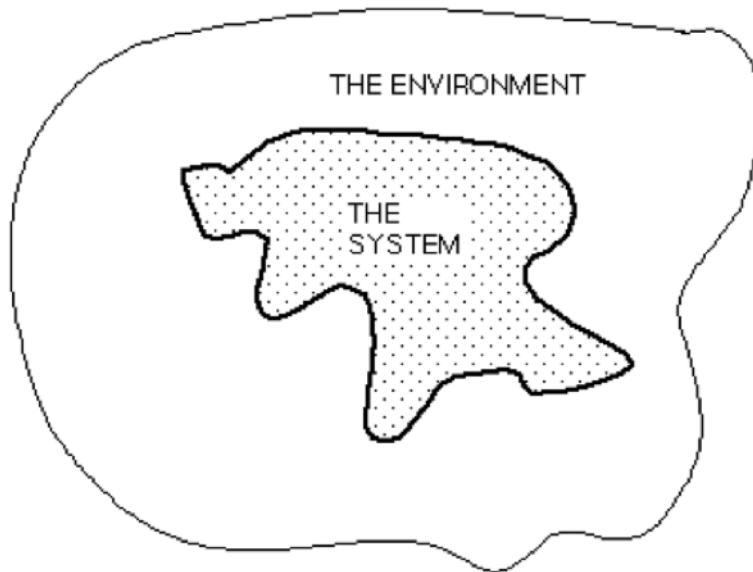


Figure 12.9: System and Environment Boundary

presents recent integration levels. Figure 12.10 presented in page 267 highlights the proposed Web and Mobile Integration Method for Sanay3y Application.

#### 12.4.2 Sanay3y Application Architecture

From the different many application architectures available, Sanay3y Application utilized the Three-Tier Architecture that is widely used for similar applications. Figure 12.11 presented in page 267 depicts Sanay3y Application Architecture. Three-Tier Architecture consists of three layers:

- Presentation Layer: deals with user interactions. Thin clients interface do not contain business logic, just code required to process user input, send requests to the server, and show the results of these requests.
- Application Layer: processes client requests. It is the actual web application that performs all functionality specific to the web application. However, it does not store the persistent data itself. Whenever it needs data of any importance, it contacts the database server.
- Database Layer: contains Database and Database Management System (DBMS). In the second and third tier there can be multiple instances, because of scalability, load-balancing and redundancy. N-tier architecture (with N more than Three) is really Three-Tier architectures in which the middle or bottom tier is split up into new tiers.

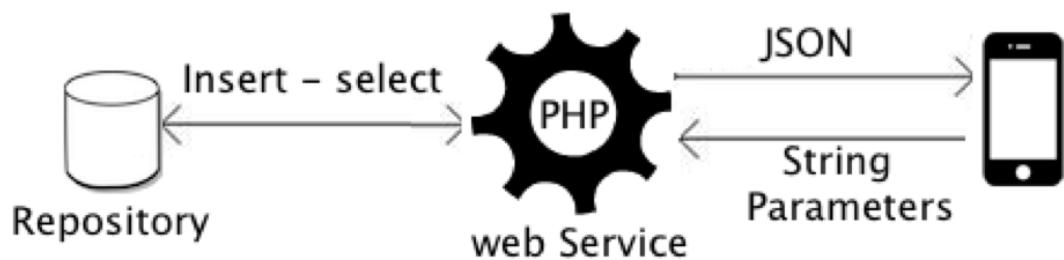


Figure 12.10: Proposed Web and Mobile Integration

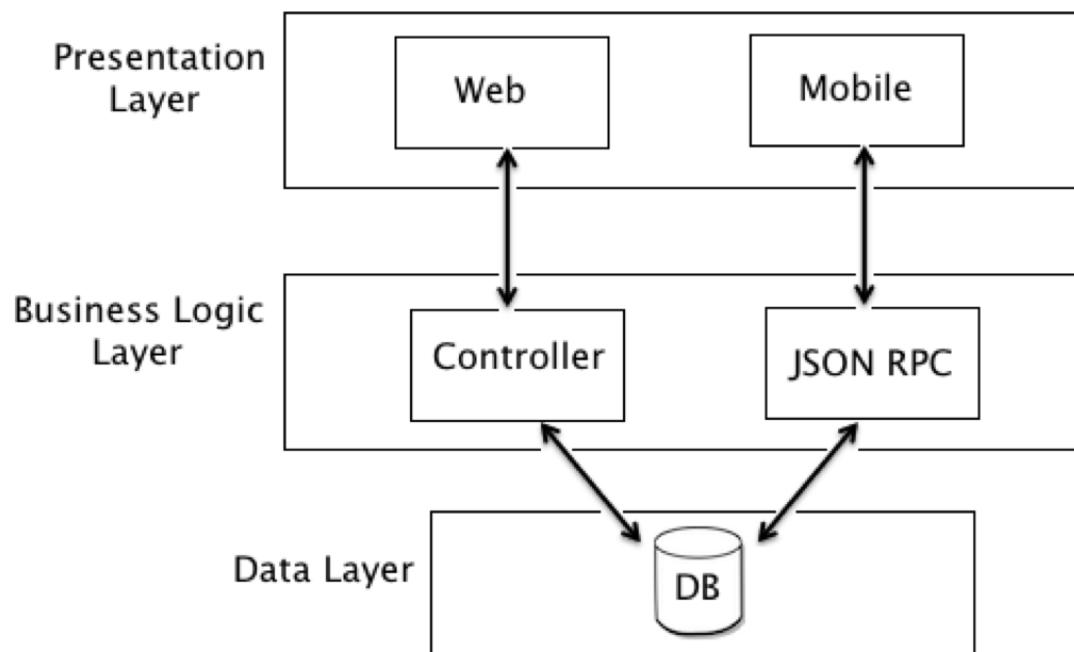


Figure 12.11: Sanay3y Application Architecture

## 12.5 Summary

The use of Smartphones are increasing day by day. Due to such increase in demand ultimately there is increase in supply. While there is increase in supply we need Mobile Application developers to develop new applications for different mobile platforms. Mobile application development is an emerging as well as relatively lucrative field with many organizations and independent developers plunging in great numbers to make hay while it lasts. It is expected to last as long as man is in love with the hand-held devices that make life so easy and accessible. Today mobile is obviously not seen just as an instrument to make calls. Mobile app development has led to the creation of innumerable innovative and unique applications such as web browsing, email, Internet faxing, games, wireless information services etc.

There are certain elements that are decisive in deciding what kind of a mobile application one should get made:

1. Platform: The cellphone application are different for different platforms and one application cannot be made to run on all the platforms. Each of the mobile platforms is quite different and therefore can run certain applications that are developed for a specific technology. We have chosen iPhone for our application because it is widely used, and available. iPhone SDK is a rich SDK that helped us through the development process.
2. Customer base: Whenever deciding to outsource the requirements of mobile software to a company it is important to determine the kind of audience you are targeting at. The target audience determines to an extent whether it will be a business mobile application or an application that will be used by individual customers. This would help in figuring out the design and technology solutions for the mobile software. Distribution capabilities of Sanay3y App using Apple App Store will help us a lot in reaching diverse customers. Because we have been working with an NGO (Non Governmental Organization), we had the accessibility of reaching thousands of online Sanay3y records that we have integrated and utilized in the project. Targeting iPhone users helped us identifying the requirements of the users, committing to Apple guidelines in developing applications helped us a lot.
3. Pricing and timeframe: It is important to take into account the costing of the mobile app development and the exact time frame for launching the product. Researching your competitors' products will help in determining the kind of price strategy you should adopt while to get the maximum return on investment it is also pertinent to launch at the right time. Investing our graduation project time and effort in presenting a real life application that is useful for Community Services turned out to be an interesting experience. In conjunction with Sanay3y NGO, we are presenting this application for free to be utilized by all iPhone users.

Presenting same application for different platforms is the next step to take. We need to present this application for BlackBerry, Android Smart Phones. We might present a Tablet edition in case Sanay3y NGO thought about adding new features.

Embedding a business model in this application is an important step. Sanay3y NGO were thinking about utilizing Sanay3y Mobile App in raising funds, or directing users for future phone calls.



# Appendix A

## Resources

This chapter includes resources that are complimentary to the information presented in this book, and that are useful for further reading. Resources are divided by sections, based on their categories.

### A.1 Book Resources

At the end of the book, I really hope you enjoyed the journey and found the book useful. Kindly accept my apologies if you have found some difficulties in interpreting some of the presented figures. This happens mainly as a result of decreasing images resolution to fit for printing. This has nothing to do with printing quality by the way. An enhanced online version of figures is available online at course page.

**Course Page** <http://www.helghareeb.me/publications/books/eaibook>

**Errata** Though we have tried heavily to make this book free of errors; and by we I mean the review team that have contributed greatly in debugging this book, I am sure there is no book edited by human that is free of errors. In case you have found any error within the book, kindly email me at:

helghareeb@mans.edu.eg

and use EAI Book Errata in the email subject. Though it is not for granted, but mostly you will be rewarded with a free edition of this book, or may be another book from my library.

### A.2 Learning Management Systems

- Angel Learning - <http://www.angellearning.com>
- ATutor - <http://www.atutor.ca>
- Avilar LMS - <http://www.avilar.com/products/lms.html>
- Basic Support for Cooperative Work - <http://bscw.fit.fraunhofer.de>
- Blackboard - <http://www.blackboard.com/us/index.aspx>

- Claroline - <http://www.claroline.net/index.ph>
- Centra-Saba - <http://www.saba.com/centra-saba>
- Colloquia - <http://www.colloquia.net>
- COSE: Creation of Study Environments - <http://www.staffs.ac.uk/COSE>
- Desire2Learn - <http://www.desire2learn.com>
- eCollege - <http://www.ecollege.com>
- e-Learning at the University of Western Cape - <http://kewl.uwc.ac.za>
- Eledge: open LMS, The University of Utah - <http://eledge.sourceforge.net>
- ILIAS, Internet Campus Solutions - <http://www.ilias.de/ios/index-e.html>
- Jenzabar - <http://www.jenzabar.com>
- IntraLearn: Learning Server - <http://www.intralearn.com>
- Janison - <http://www.janison.com.au/janison/default.asp>
- KnowEdge: eLearning Suite - <http://www.knowedge.net>
- The Learning Manager - <http://www.thelearningmanager.com>
- LON-CAPA - <http://www.lon-cap.org>
- moodle - <http://www.moodle.org>
- Unicon: open source commercial support - <http://www.unicon.net>

# Appendix B

## Acronyms and Glossary

### Acronyms

ADL	Advanced Distributed Learning
AEHS	Adaptive Educational Hypermedia Systems
AIEHS	Adaptive Intelligent Educational Hypermedia Systems
ANN	Artificial Neural Network
ANSI/IEEE	American National Standards Institute / Institute of Electrical and Electronics Engineers
ARPA	Advanced Research Projects Agency
BA	Business Analytics
BI	Business Intelligence
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Webservice
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BPMS	Business Process Management System
BPR	Business Process Reengineering
CASE	Computer assisted Systems Engineering
CBA	Component Based Architecture
CMS	Course Management System
CORBA	Common Object Request Broker
CRM	Customer Relation Management
DB	Data Base
DBMS	Database Management System
DCE	Distributed Computing Environment
DM	Data Mining
DSS	Decision Support System
DW	Data Warehouse

EA	Enterprise Architecture
EAI	Enterprise Application Integration
EIP	Enterprise Information Portal
EIS	Enterprise Information System
ELF	E-Learning Framework
ERM	Enterprise Resource Management
ERP	Enterprise Resource Planning
ES	Expert System
ESB	Enterprise Service Bus
FIPA	Foundation for Intelligent Physical Agents
GSS	Group Support System
HTTP	Hyper Text Transfer Protocol
IA	Intelligent Agent
IS	Information System
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
ITS	Intelligent Tutoring System
KAoS	Knowledgeable Agent-oriented System
KIF	Knowledge Interchange Format
KMP	Knowledge Management Portal
KMS	Knowledge Management System
KQML	Knowledge Query and Manipulation Language
LMS	Learning Management System
MAS	Multi Agent System
MIS	Management Information System
MNAS	Mobile News Agent System
moodle	Modular Object Oriented Dynamic Learning Environment
MSS	Management Support System
NAEP	National Assessment of Education Process
OKI	Open Knowledge Initiative
OLAP	Online Analytical Processing
ROI	Return On Investment

RPC	Remote Procedure Call
SCM	Supply Chain Management
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOI	Service Oriented Integration
SW	Software
TQM	Total Quality Management
UDDI	Universal Description Discovery and Integration
UMIS	University Management Information System
VLE	Virtual Learning Environment
W3C	World Wide Web Consortium
WfMC	Workflow Management Coalition
WfMS	Workflow Management System
WOA	Web Oriented Architecture
WSDL	Webservice Description Language
XML	Extensible Markup Language
XPDL	XML Process Definition Language

## Glossary

### **ADL**

ADL: Advanced Distributed Learning, established in 1997 to standardize and modernize training and education management and delivery. The Department of Defense (DoD) Office oversees the initiative. <http://www.adlnet.gov>

### **AEHS**

Adaptive Hypermedia Systems as systems that refer to all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system; it should have a user model; it should be able to adapt the hypermedia using this model. Educational Hypermedia Systems are the most popular area for adaptive hypermedia research.

### **AIEHS**

Adaptive and Intelligent Educational Hypermedia System: Enhancing AEHS with methods and techniques from ITS creates the AIEHS.

### **ANN**

Artificial Neural Network: a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data.

### **BA**

Business Analytics: refers to the skills, technologies, applications and practices for continuous iterative exploration and investigation of past business performance to gain insight and drive business planning. BA focuses on developing new insights and understanding of business performance based on data and statistical methods.

### **BI**

Business Intelligence traditionally focuses on using a consistent set of metrics to both measure past performance and guide business planning of an enterprise, which is also based on data and statistical methods.

### **BPM**

Business Process Management is a systematic, structured approach to analyze, improve, control, and manage processes. Business Process is a series of interrelated activities that cross functional enterprise boundaries with individual inputs and outputs.

### **CASE**

Computer assisted Systems Engineering (CASE Tools): is the scientific application of a set of tools and methods to a software system which results in

high-quality, defect-free, and maintainable software products. It also refers to methods for the development of IS together with automated tools that can be used in the software development process.

**CORBA**

CORBA is a well-known specification for a middleware system that was developed in the 1990s by the Object Management Group. It was intended as an open standard that would allow the development of middleware to support distributed component communications and execution, plus provide a set of standard services that could be used by these components.

**CRM**

Customer Relation Management: is a widely implemented model for managing a company's interactions with customers, clients, and sales prospects. It involves using technology to organize, automate, and synchronize business processes-principally sales activities, but also those for marketing, customer service, and technical support.

**DB**

Database is a collection of persistent data.

**DM**

Data Mining is the process that attempts to discover patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

**DSS**

Decision Support System: a computer-based information system that supports business or organizational decision-making activities. DSSs serve the management, operations, and planning levels of an organization and help to make decisions, which may be rapidly changing and not easily specified in advance. DSSs can be either fully computerized, human or a combination of both.

**DW**

Data Warehouse: data warehouse or enterprise data warehouse (DW, DWH, or EDW) is a database used for reporting and data analysis. It is a central repository of data which is created by integrating data from multiple disparate sources. Data warehouses store current as well as historical data and are used for creating trending reports for senior management reporting such as annual and quarterly comparisons.

**EA**

Enterprise Architecture is the process of translating business vision and strategy into effective enterprise change by creating, communicating and improving the key requirements, principles and models that describe the enterprise's future state and enable its evolution. Practitioners of EA call themselves enterprise architects. An enterprise architect is a person responsible for performing this complex analysis of business structure and processes and is often called

upon to draw conclusions from the information collected. By producing this understanding, architects are attempting to address the goals of EA: Effectiveness, Efficiency, Agility, and Durability.

**EAI**

Enterprise Application Integration integrates applications and systems using middleware. EAI solutions are technology based but often not based on standards such as Web services. EAI has two main integration patterns: hub and spoke and publish-subscribe. EAI solutions using messaging middleware could be programmed differently to have such awareness.

**EIP**

Enterprise Information Portal: is a framework for integrating information, people and processes across organizational boundaries. It provides a secure unified access point, often in the form of a web-based user interface, and is designed to aggregate and personalize information through application-specific portlets. One hallmark of enterprise portals is the de-centralized content contribution and content management, which keeps the information always updated.

**EIS**

Enterprise Information System: is generally any kind of computing system that is of "enterprise class". This means typically offering high quality of service, dealing with large volumes of data and capable of supporting some large organization ("an enterprise"). EISs provide a technology platform that enables organizations to integrate and coordinate their business processes.

**ELF**

E-Learning Framework, an international effort to develop a service-oriented approach to the development and integration of computer systems in the sphere of learning, research and education administration. Official Website: <http://www.elframework.org>

**Enterprise**

Enterprise is simply a convenient generic term for any reasonably self-contained commercial, scientific, technical or other organization.

**ERM**

Enterprise Resource Management: describes software that lets an enterprise manage user access to its network resources efficiently. ERM software generally lets a user sign on to different enterprise systems and applications using the same password. ERM software makes it easy for the enterprise to control and keep track of which systems and resources each user has access to, and provides consistent standards for creating and changing passwords.

**ERP**

Enterprise Resource Planning: systems integrates internal and external management information across an entire organization, embracing finance/accounting, manufacturing, sales and service, CRM, etc. ERP systems automate this activity with an integrated software application.

**ES**

Expert System: is a computer system that emulates the decision-making ability of a human expert. ESs are designed to solve complex problems by reasoning about knowledge, like an expert, and not by following the procedure of a developer as is the case in conventional programming.

**ESB**

Enterprise Service Bus: is an architectural pattern that has multiple motivations: Supports large numbers of service interactions in a manageable way, Provides support for advanced service interaction capability, such as transactions, store and forward, infrastructure services, security, quality of service, and so on, Supports a variety of interaction styles such as synchronous request/response, messaging, publish/subscribe, and events, Provides a robust, manageable, distributed integration infrastructure consistent with the principles of SOA, Supports service routing and substitution, protocol transformations, and other message processing, Supports both Web Services and traditional EAI communication standards and technologies.

**GSS**

Group Support Systems: Facilitates decisions made by groups within the organization. Traditional meeting can last a long time and money. Collaborative computing systems, groupware, electronic meeting can run over the Web and provide video conferencing, audio conferencing in addition to meeting tools like electronic brainstorming, voting and document sharing.

**IA**

Intelligent Agent: is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals (i.e. it is rational). IAs may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an IA, as is a human being, as is a community of human beings working together towards a goal.

**IS**

Information System can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to supporting decision making, coordination, and control, information systems may also help managers and workers analyze problems, visualize complex subjects, and create new products

**KMP**

Knowledge Management Portal: provides a platform to store information and knowledge into multiple place, exploit it and share it like never before. Helps organization to organically grow their knowledge base. In a typical organization, some 75% of internal knowledge is trapped in one's email and desktop computers. Good information should be everywhere and nowhere near.

**KMS**

Knowledge Management Systems: A KMS is a computerized system designed to support the creation, storage, and dissemination of information. Such a system contains a central repository of information that is well structured and employs a variety of effective and easy to use search tools that users can use to find answers to questions quickly.

**LMS**

Learning Management System is the software that automates the administration of training. LMS registers users, tracks courses in a catalog, records data from learners, and provides reports to management. An LMS is typically designed to handle courses by multiple publishers and providers. It usually doesn't include its own authoring capabilities. Instead, it focuses on managing courses created by a variety of other sources.

**moodle**

Moodle: an Open Source LMS. Stands for: Modular Object Oriented Dynamic Learning Environment. Official Moodle website: <http://moodle.org/>

**MSS**

Management Support System is the support of management tasks by the application of technologies. Sometimes called DSSs or BI.

**OKI**

OKI: Open Knowledge Initiative, an organization responsible for the specification of software interfaces comprising a SOA based on high level service definitions. Official Website: [www.okiproject.org](http://www.okiproject.org)

**OLAP**

Online Analytical Processing: is an approach to answering multidimensional analytical (MDA) queries swiftly. OLAP is part of the broader category of BI, which also encompasses relational database report writing and DM. Typical applications of OLAP include business reporting for sales, marketing, management reporting, BPM (BPM), budgeting and forecasting, financial reporting and similar areas, with new applications coming up, such as agriculture.

**SCM**

Supply Chain Management:is the management of a network of interconnected businesses involved in the provision of product and service packages required by the end customers in a supply chain. SCM spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption.

**SOA**

Service Oriented Architecture: There is no agreed upon definition for SOA. Refer to section 1.5.1 presented at page 13 for more information.

**SOAP**

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably HTTP and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

**UDDI**

Universal Description, Discovery and Integration (UDDI) is a platform independent, Extensible Markup Language (XML)-based registry by which businesses worldwide can list themselves on the Internet, and a mechanism to register and locate web service applications. UDDI is an open industry initiative, sponsored by the Organization for the Advancement of Structured Information Standards (OASIS), for enabling businesses to publish service listings and discover each other, and to define how the services or software applications interact over the Internet.

**WSDL**

The Web Services Description Language is an XML-based interface description language that is used for describing the functionality offered by a web service. A WSDL description of a web service (also referred to as a WSDL file) provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. It thus serves a roughly similar purpose as a method signature in a programming language.

**XML**

Extensible Markup Language is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.



# Bibliography

- [1] Architectural patterns, enterprise architecture standards, certification, and services, the open group.
- [2] Artifical neural networks, wikipedia.
- [3] Biztalk home - your business connected.
- [4] Business analytics, wikipedia.
- [5] Business intelligence, wikipedia.
- [6] Cbdi service oriented architecture practice portal.
- [7] Client/server and the n-tier model of distributed computing, n-tier.com, micromax information services.
- [8] Computer assisted systems engineering, wikipedia.
- [9] Corba basics, omg: Open management group.
- [10] Cps 720, artificial intelligence programming, ryerson university, toronto, canada.
- [11] Customer relationship management, wikipedia.
- [12] Data mining, wikipedia.
- [13] Data warehouse, wikipedia.
- [14] Decision support systems, wikipedia.
- [15] Developing agile organizations: Skills for managers and team leaders, gower.
- [16] E-commerce, wikipedia.
- [17] e-learning engineering.
- [18] Enterprise information portals, wikipedia.
- [19] Enterprise information system, wikipedia.
- [20] Enterprise resource management.
- [21] Enterprise resource planning, wikipedia.

- [22] Expert systems, wikipedia.
- [23] Fayetteville state university: Blackboard virtual classroom features.
- [24] Fipa.org.
- [25] The foundation for intelligent physical agents (fipa).
- [26] Global virtual classroom.
- [27] Houston pbs: 50 years of houston pbs history, channel 8, houstonpbs.
- [28] Intelligent agents, wikipedia.
- [29] An introduction to enterprise architecture, arch2arch, bea.
- [30] Joint information systems committee (jisc): Supporting education and research—a higher education committee, great britain higher education.
- [31] Knowledge management portals, wikipedia.
- [32] Knowledge management systems, wikipedia.
- [33] Message broker, msdn: Microsoft digital subscriber network.
- [34] Message bus, msdn: Microsoft developers network.
- [35] Michigan state university's virtual university.
- [36] Online analytical processing, wikipedia.
- [37] Oracle e-business suite.
- [38] Publish/subscribe, msdn: Microsoft digital developers network.
- [39] Sap: Business software solutions applications and services.
- [40] Sri consulting business intelligence.
- [41] Supply chain management, wikipedia.
- [42] Technologies for online interoperable assessment (toia).
- [43] Wbt systems e-learning from wbt systems.
- [44] Enterprise architecture seminar, software architecture training and consulting, bredemeyer consulting, 2004.
- [45] *Management Information System: Managing the Digital Firm*. Prentice Hall, 9th edition, 2006.
- [46] Haitham A. El-Ghareeb A. M. Riad. New architecture for mobile news agent system. *Egyptian Informatics Journal*, 8, 2007.

- [47] I. Aaen, A. Börjesson, and L. Mathiassen. *SPI agility: How to navigate improvement projects*. John Wiley and Sons, 2006.
- [48] A. S. Agee and C. Yang. Top ten it issues. *educause review*, 43(4), EduCAUSE, 2009.
- [49] H. Ajjan and R. Hartshorne. Investigating faculty decisions to adopt web 2.0 technologies: Theory and empirical tests. *The Internet and Higher Education*, 11(2), 2008.
- [50] Ajlan Al-Ajlan and Hussein Zedan. E-learning (moodle) based on service oriented architecture. Technical report, De Montfort University, ZDnet White Papers, 2008.
- [51] S. T. Albin. *The Art of Software Architecture: Design Methods and Techniques*. John Wiley and Sons, 2003.
- [52] B. Alexander and A. Levine. Web 2.0 storytelling. emergence of a new genre. In *In EduCAUSE Review*, volume 43, pages 40–56. EduCAUSE, 2008.
- [53] H. A. Ali and A. M. Riad. Studying the performance of distributed database application based on a heuristic algorithm for query execution. In *2nd International Conference on Information (ICI-2), 7-9 Oct., Mansoura, Egypt*, pages 636–655, 2003.
- [54] G. Anderson and D. Larocca. *Sams Teach Yourself SAP<sup>TM</sup> in 24 Hours, Second Edition*. SAMA, 2005.
- [55] T. Anderson and R. Mason. The bangkok project: New tool for professional development. *American Journal of Distance Education*, 7:5–18, 1993.
- [56] G. Andrews. Paradigms for process interaction in distributed programs. *ACM Computing Surveys*, 23:49–90, 1991.
- [57] Theodre K. Apostolopoulos and Anna Kefala. An xml-based e-learning service management framework. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 2004.
- [58] Christopher Bailey, Wendy Hall, David E. Millard, and Mark J. Weal. Towards open adaptive hypermedia. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, Spain, 2002.
- [59] Christopher Paul Bailey. An agent-based framework to support adaptive hypermedia. *Departments of Electronics and Computer Science, Faculty of Engineering, University of Southampton, United Kingdom*, 2002.
- [60] M. Bangemann and E. Cresson. Educational software and multimedia. *Task Force, European Commission*, 1996.

- [61] D. K. Barry. *Web Services and Service-Oriented Architecture: The Savvy Manager's Guide*. Morgan Kaufmann Publishers, 2003.
- [62] R. Baskerville. *Business Agility and Information Technology Diffusion*. Springer, 2005.
- [63] P. Bellavista, A. Corradi, and A. Tomasi. The mobile agent technology to support and to access museum information. In *Proceedings of the 2000 ACM symposium on Applied computing - Volume 2, ACM Press*, pages 1006–1013, 2000.
- [64] P. Bergen. Garfixia software architectures.
- [65] P. Bernus and M. Fox. Knowledge sharing in the integrated enterprise: Interoperability strategies for the enterprise architect. *Springer*, 2005.
- [66] D. Bitzer, E. Lyman, and J. Easley. The uses of plato: A computer-controlled teaching system. *Coordinated Science Laboratory, University of Illinois*, 1965.
- [67] D. Booth. Web services architecture, w3c.
- [68] A. Bork. The future of learning: An interview with alfred bork. *Review, Educom*, 34, 1999.
- [69] Alfred Bork, Carole A. Bagley, and Xiwen Zhang. Technology and learning: A proposal for adaptive learning tutorial. In *Michael Allen's 2008 e-Learning Annual*, volume 11. J-B Pfeiffer Annual Looseleaf, 2008.
- [70] J. M. Bradshaw. Kaos: An open agent architecture supporting reuse, interoperability, and extensibility.
- [71] M. Brake. Explaining corba. Technical report, Colorado State University, 1998.
- [72] S. Britain and O. Liber. *A Framework for the Pedagogical Evaluation of Virtual Learning Environments*. JISC Technology Applications Programme., 2008.
- [73] W. Broderick, J. Brahan, and R. Shevel. An instructional management system for natal-74. In *National Research Council Canada, Third Canadian Symposium on Instructional Technology*, 1980.
- [74] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction: 6*, 1996.
- [75] Peter Brusilovsky. A distributed architecture for adaptive and intelligent learning management systems. In *Proceedings of the AIED 2003 Workshop Towards Intelligent Learning Management Systems*, Sydney, 2003.
- [76] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction: 11*, 2001.

- [77] Daniel Burgos, Colin Tattersall, and Rob Koper. Representing adaptive and adaptable units of learning. how to model personalized elearning in ims learning design. In *CELSTEC Research on Learning Networks, Open Universities*, Netherlands, 2006.
- [78] L. Butler. Curriculum development system for navy technical training. *Eric*, 1990.
- [79] Schooley C., Charron C., Driver E. Li C., and Fossner L. Social computing re-shapes elearning: Informal learning becomes a component in blending learning and work. Technical report, Forrester Research, Inc., 2006.
- [80] M. Cambronero, , G. Diaz, J. Pardo, and V. Valero. Using uml diagrams to model real-time web services. In *Second International Conference on Internet and Web Applications and Services (ICIW'07)*, 2007.
- [81] California Distance Learning Project (CDLP). What is distance learning? for adult educators.
- [82] CDW-G. The 2009 21-st century campus report.
- [83] V. W. Chairman. A glossary of e-learning terms and acronyms, elearning network.
- [84] J. F. Chang. *Business Process Management Systems: Strategy and Implementation*. AUERBACH, 2005.
- [85] Wan-Jen Chang, Hsuan-Hung Chen, Hui-Kai Su, Yuan-Sun Chu, and Kim-Joan Chen. Design of knowledge management learning system based on service oriented architecture. In *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06)*, 2006.
- [86] Brant A. Cheikes. Gia: An agent-based architecture for intelligent tutoring systems.
- [87] W. Chen and W. Tseng. A novel multi-agent framework for the design of home automation. In *International Conference on Information Technology (ITNG'07)*, pages 277–281, 2007.
- [88] Y. Chen. Use feedback and fuzzy self-organizing map to improve adaptive e-learning model. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA)*, 2007.
- [89] D. N. Chorafas. *Integrating ERP, CRM, Supply Chain Management, and Smart Materials*. CRC Press, 2001.
- [90] Chih-Ping Chu, Ching-Pao Chang, Chung-Wei Yeh, and Yu-Fang Yeh. A web-service oriented framework for building scorm compataible learning management systems. In *International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2004.

- [91] D. Clark and D. Tennenhouse. Architectural considerations for a new generation of protocols. In *ACM SIGCOMM'90 Symposium*, pages 200–208, 1990.
- [92] J. Coelho. *BPM and Continuous Improvement*. Meghan-Kiffer Press, 2005.
- [93] T. Connolly and C. Begg. *Database Systems: A Practical Approach to Design, Implementation and Management*. Addison-Wesley, 2004.
- [94] Albert T. Corbett, Kenneth R. Koedinger, and John R. Anderson. Intelligent tutoring systems. *Handbook of Human-Computer Interaction*, 1997.
- [95] Declan Dagger, Alexander O'Connor, Séamus Lawless, Eddie Walsh, and Vincent Wade. Service oriented elearning platforms: From monolithic systems to flexible services. *IEEE INTERNET COMPUTING*, 2007.
- [96] C. J. Date. *An Introduction to Database Systems*. Addison Wesley, 2004.
- [97] Lawrence Wilkes David Sprott. Understanding service oriented architecture. *Microsoft Architect Journal*, 1:10–17, 2004.
- [98] S. de Cesare and A. Serrano. Collaborative modeling using uml and business process simulation. In *39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, 2006.
- [99] G. V. de Putte, T. Benedetti, D. Gagic, P. Gersak, K. Krutzler, and M. Perry. *Intra-Enterprise Business Process Management*. IBM Redbooks, 2001.
- [100] M. K. Denko. The use of mobile agents for clustering in mobile ad hoc networks. In *Annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology SAICSIT '03*, pages 241–247, 2003.
- [101] Michael Derntl and Juergen Mangler. Web services for blended learning patterns. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 2004.
- [102] M. d'Inverno and M. Luck. *Understanding Agent Systems*. Springer, 2003.
- [103] K. Dirk, B. Karl, and S. Dirk. *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall, 2004.
- [104] B. P. Douglass. *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*. Addison Wesley Professional, 2002.
- [105] R. Dove. *Response Ability-The Language, Structure, and Culture of the Agile Enterprise*. Wiley, 2001.
- [106] S. Downes. Ten web 2.0 things you can do in ten minutes to be a more successful e-learning professional. *eLearn Magazine*, 2008.

- [107] E. Driver, C. Moore, P. Jackson, T. Keitt, C. Schooley, and J. Barnett. Web3d: The next major internet wave for information and knowledge management professionals. Technical report, Forrester Research, Inc., 2008.
- [108] T. M. Duffy and R. J. Kirkley. *Learner-Centered Theory and Practice in Distance Education*. Lawrence Erlbaum Associates, 2004.
- [109] Charles D. Dziuban, Joel L. Hartman, and Patsy D. Moskal. Blended learning. *Educase Connect*, 2004.
- [110] Franya Franek Edward R. Sykes. Web-based architecture of an intelligent tutoring system for remote students learning to program java.
- [111] John Eklund, Peter Brusilovsky, and Elmar Schwarz. Adaptive textbooks on the world wide web. In *Third Australian World Wide Web Conference*. Southern Cross University, 1997.
- [112] Haitham A. El-Ghareeb. E-learning and management information systems: Universities need both. *eLearn Magazine, ACM*, 2009.
- [113] Haitham A. El-Ghareeb and A. M. Riad. A service oriented architecture to integrate web services and software agents in course management systems. ", *Egyptian Informatic Journal, Cairo Univiersity*, 8(1), 2007.
- [114] Haitham A. El-Ghareeb and A. M. Riad. A service oriented architecture to integrate mobile assessment in learning management systems. *Turkish Online Journal of Distance Education – TOJDE*, 2008.
- [115] H. El-Hadidi, A. M. Riad, and M. Matsumoto. A multi-agent system to enhance information retrieval results. In *IEIC*, 2007.
- [116] PCmag.com Encyclopedia. Pcmag.com: Audio conferencing.
- [117] PCmag.com Encyclopedia. Pcmag.com: Data conferencing.
- [118] PCmag.com Encyclopedia. Pcmag.com: Video conferencing.
- [119] M. Endrei. *Patterns: Service Oriented Architecture and Web Services*. IBM Redbooks, 2004.
- [120] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, T. Newling, and Tony. *Patterns: Service Oriented Architecture and Web Services*. IBM Redbooks, 2004.
- [121] T. Erl. *Service Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.
- [122] T. Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.
- [123] T. Erl. *Service Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.

- [124] E. Espinosa, V. Robledo-Rella, L. Neri, and J. Noguez. Towards an adaptive delivery of evaluation tools. In *Frontiers in education conference, FIE apos; 07*, 2007.
- [125] N. Evans. *Business Agility: Strategies for Gaining Competitive Advantage Through Mobile Business Solutions*. Prentice Hall, 2001.
- [126] A. Feenberg. Distance learning: Promise or threat?
- [127] L. Ferrara, F. Hayden, D. Neal, and C. Poirier. *The Networked Supply Chain: Applying Breakthrough BPM Technology to Meet Relentless Customer Demands*. J. Ross Publishing, 2003.
- [128] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, Irvine: University of California., 2001.
- [129] Peter Fingar. *Business Process Management: Integration in a Web-enabled environment*. Prentice Hall, 2003.
- [130] T. Finin, R. Fritzson, D. McKay, and R. McEntire. Kqml as an agent communication language. In *3rd International Conference on Information and Knowledge Management (CIKM'94)*, 1994.
- [131] J. Fischman. Continuing education and social networking combine to attract students. Technical report, College of Liberal and Professional Studies, Penn LPS., 2009.
- [132] R. Flores-Mendez. Towards a standardization of multi-agent system framework. *Crossroads, ACM Press*, 9:18–24, 1999.
- [133] I. Foster, Y. Zhau, R. Ioan, and S. Lu. Cloud computing and grid computing: 360-degree compared. In *Grid Computing Environments Workshop*. Cornell University Library, 2008.
- [134] R. Foster and S. Kaplan. *Creative Destruction: Why Companies That Are Built to Last Underperform the Market – and How to Successfully Transform Them*. McKinsey, 2001.
- [135] S. Fox, K. Zickuhr, and A. Smith. Twitter and status updating. Technical report, Pew Internet and American Life Project, 2009.
- [136] Stan Franklin and Art Graesser. Is it an agent, or just a program?:a taxonomy for autonomous agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag, 1996.
- [137] D. Randy Garrison and Heather Kanuka. Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, 7(2):95–105, 2004.

- [138] Alfredo Garro and Luigi Palopoli. An xml multi-agent system for e-learning and skill management. *Agent Technologies, Infrastructures, Tools, and Applications for E-Services , Lecture Notes in Computer Science*, pages 283–294, 2003.
- [139] Gartner. Enterprise architecture.
- [140] M. R. Genesereth. draft proposed american national standard (dpans). Technical report, Knowledge Interchange Format, NCITS.T2/98-004, 2004.
- [141] G. Grosseck. To use or not to use web 2.0 in higher education? In *World Conference on Educational Sciences, New Trends and Issues in Educational Sciences.*, volume 1. El-Sevier, Nicosia, North Cyprus., 2009.
- [142] The Open Group. Enterprise architecture.
- [143] The Open Group. Togaf: Introduction, 2011.
- [144] M. Gudgin. Soap version 1.2 part 1: Messaging framework, w3c.
- [145] J. M. Gundry. Web 2.0 social media, knowledge ability ltd.
- [146] Christian Gutl, Victor Manuel Barrios, and Felix Modritscher. Adaptation in e-learning environments through the service-based framework and its application for adele. *IDC: US Corporate and Government eLearning Forecast*, 2003.
- [147] B. Ha, J. Bae, and S. Kang. Workload balancing on agents for business process efficency based on stochastic model. In *Second International Conference on Business Process Management (BPM)*, 2004.
- [148] K. A. Hall. *Computer-Assisted Instruction: Status in Pennsylvania*. Pennsylvania State University, 1970.
- [149] B. L. Hardy and K. Bower. From correspondence to cyberspace: Changes and challenges in distance education. In *New Directions for Community Colleges*, pages 5–12, 2004.
- [150] Andreas Harrer, Niels Pinkwart, Bruce M. McLaren, and Oliver Scheuer. The scalable adapter design pattern: Enabling interoperability between educational software tools. *IEEE Transactions on Learning Technologies*, 2008.
- [151] S. Harwood. *ERP: The Implementation Cycle*. NetLibrary Inc., 2003.
- [152] Ioannis Hatzilygeroudis, Constantinos Koutsojannis, Constantinos Papavlasopoulos, and Jim Prentzas. Knowledge-based adaptive assessment in a web-based intelligent educational system. In *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies (ICALT)*, 2006.

- [153] J. V. Hebenstreit. Computer-assisted instruction in france: present situation and prospects for the future. In *Ottawa: National Research Council Canada, Proceedings of the Third Canadian Symposium on Instructional Technology*, pages 77–91, 1980.
- [154] D. Hill, B. Webster, E. Jezierski, S. Vasireddy, M. Al-Sabt, B. Wastell, J. Rasmussen, P. Gale, and P. Slater. Smart client architecture and design guide. In *Microsoft Patterns and Practices*, Microsoft Corporation, 2004.
- [155] S. Hoare. Schools of enterprise application architecture: Semantic arts - enterprise architecture development and application integration. *Semantic Arts*, 2003.
- [156] L. Hohmann. *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Addison Wesley, 2003.
- [157] G. Hohpe and B. Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison Wesley Professional, 2003.
- [158] D. Holden. Restructuring schools on a service-industry model. *T.H.E. Journal*, 21:70–71, 1994.
- [159] KERRIE HOLLEY and ALI ARSANJANI. *100 SOA Questions*. Prentice Hall, 2011.
- [160] Willam Horton and Catherine Horton. *E-Learning Tools and Technologies*. Wiley, 2003.
- [161] Anna Hovakimyan, Siranush Sargsyan, and Sergey Barkhoudaryan. Genetic algorithm and the problem of getting knowledge in e-learning systems. In *ICALT: Proceedings of the IEEE International Conference on Advanced Learning Technologies*, 2004.
- [162] Mu-Jung Huang, Hwa-Shan Huang, and Mu-Yen Chen. Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems with Applications: An International Journal*, 2007.
- [163] A. K. Hulme and John T. *Mobile Learning: A Handbook for Educators and Trainers (Open and Flexible Learning)*. Routledge Falmer, 2005.
- [164] S. Hunka and G. Buck. The rise and fall of cai at the university of alberta's faculty of education. *Canadian Journal of Educational Communication*, 21:153–170, 1996.
- [165] J. Hurwitz. *Service Oriented Architecture For Dummies*. Wiley, 2007.
- [166] IASTED, Artificial Intelligence and Soft Computing. *A Framework of Incorporating Software Agents into SOA*, 2005.
- [167] W. H. Inmon. *Building the Data Warehouse*. Wiley, 2005.

- [168] A. Ghorbani J. Carter and S. Marsh. mobile software agents: Just-in-time information sharing architectures in multiagent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, ACM Press, pages 647–654, 2002.
- [169] R. Tobin J. Cowan. Xml information set (second edition), w3c.
- [170] A. Jafari. Putting everyone and every course online: The oncourse environment. *Webnet Journal*, 33, 1999.
- [171] P. Jiang, Q. Mair, and J. Newman. Using uml to design distributed collaborative workflows: from uml to xpdl. In *Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003.
- [172] L. Bass B. John and J. Kates. Achieving usability through software architecture. Technical report, Software Engineering Institute (SEI), 2001.
- [173] Andrew Josey. Togaf version 9.1 - enterprise edition. Technical report, The Open Group, 2011.
- [174] E. Kaplan-Leiserson. Astd's source for e-learning. learning circuits.
- [175] G. Kappel, E. Michlmayr, B. Pröll, S. Reich, and W. Retschitzegger. *Web Engineering - Old Wine in New Bottles?* Springer, 2004.
- [176] C. Klims. Portal solutions for university information systems with the lms integration.
- [177] Pedro J. Muñoz-Merino Carlos Delgado Kloos and Jesús Fernández Naranjo. Enabling interoperability for lms educational services. *Computer Standards and Interfaces, El-Sevier*, 2009.
- [178] M. Klusch. *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*. Springer, 1999.
- [179] Alfred Kobsa, Jurgen Konemann, and Wolfgang Pohl. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16:111–155, 2001.
- [180] D. Kotz, G. Cybenko, R. S. Gray, G. Jiang, R. A. Peterson, M. O. Hofmann, D. A. Chacón, K. R. Whitebread, and J. Hendler. Performance analysis of mobile agents for filtering data streams on wireless networks. In *Mobile Networks and Applications, Volume 7 Issue 2*, Kluwer Academic Publishers, pages 163–174, 2002.
- [181] E. Küfi and B. Özgür. Web 2.0 in learning english: the student perspective. In *World Conference on Educational Sciences - New Trends and Issues in Educational Sciences, Nicosia, North Cyprus*, volume 1, 2009.
- [182] M. Kunkel. From stand-alone lms to integrated systems.

- [183] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*, 3/E. Addison Wisely, 2005.
- [184] H. LAU, L. ZHANG, and C. LIU. Solving generalized open constraint optimization problem using two-level multi-agent framework. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 558–564, 2005.
- [185] L. Lau. *Managing Business with SAP: Planning, Implementation, and Evaluation*. IGI Publishing, 2005.
- [186] D. Leidner, R. Beatty, and J. Mackay. How cios manage it during economic decline: Surviving and thriving amid uncertainty. *MIS Quarterly Executive*, 2, 2003.
- [187] A. Lenhart and M. Maden. Social networking web sites and teens. Technical report, Pew Internet and American Life Project, 2007.
- [188] G. A. Lewis. Cloud computing. sei [webinar]. Technical report, Carnegie Mellon University, 2009.
- [189] J. Liberty. *Programming CSharp*. Wiley, 2005.
- [190] I. Liccardi, A. Ounnas, R. Pau, E. Massey, P. Kinnunen, S. Lewthwaite, M. Midy, and C. Sarkar. The role of social networks in students' learning experiences. In *Annual Joint Conference Integrating Technology into Computer Science Education, Working group reports on ITiCSE on Innovation and technology in computer science education*. ACM, NY, USA., 2007.
- [191] P. Liew, K. Kontogiannis, and T. Tong. A framework for business model driven development. In *12 International Workshop on Software Technology and Engineering Practice (STEP'04)*, pages 47–56, 2004.
- [192] D. S. Linthicum. *Next Generation Application Integration: From Simple Information to Web Services*. Addison-Wisely, 2004.
- [193] N. Llewellyn and C. Armistead. Business process management: Exploring social capital with processes. *International Journal of Service Industry Management*, MCB University Press, 11:225–234, 2000.
- [194] L. E. Long and N. Long. *Introduction to Computers and Information Systems: The Internet Edition*. Prentice Hall, 1996.
- [195] K. K. Loo and B. Cheng. Fuzzy logic and data mining for e-learning personalization. In *Proceedings of Artificial Intelligence and Applications*. ACTA Press, 2003.
- [196] H. Lowood. Supplement: Current bibliography in the history of technology. *JSTOR, Technology and Culture*, 39(1-138), 1990.

- [197] Zongmin Ma. Web-based intelligent e-learning systems: Technologies and applications. *Information Science Publishing*, 2006.
- [198] A. Macaulay. Enterprise architecture design and the integrated architecture framework. *Microsoft Architects Journal*, pages 4–9, 2004.
- [199] Jim E. Greer Mary A. Mark. Evaluation methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 1993.
- [200] N. Matar, S. Khwaldeh, and Z. Hunaiti. Adaptive unified e-learning system for supporting better e-learning approac. In *The 8th Annual Postgraduate Symposium: The Convergence of Telecommunications, Networking and Broadcasting*. Liverpool John Moores University, 2007.
- [201] M. McClellan. *In Search of BPM Excellence - Straight from the Thought Leaders*. Meghan-Kiffer Press, 2005.
- [202] B. McCrea. Purdu u brings social networking to the classroom: Hotseat allows students to text in class. Technical report, Campus Technology, 2009.
- [203] Miikka Miettinen, Jaakko Kurhila, and Henry Tirri. On the prospects of intelligent collaborative e-learning systems. In *The 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, 2005.
- [204] D. P. Millar and R. L. Heath. *Responding to Crisis: A Rhetorical Approach to Crisis Communication*. Lawrence Erlbaum Associates, 2003.
- [205] Felix Modritscher, Victor Manuel Garcia-Barrios, and Christian Gütl. The past, the present, and the future of adaptive e-learning. In *Proceedings of the International Conference on Interactive Computer Aided Learning (ICL2004)*, 2004.
- [206] D. Moldt and J. Ortmann. A conceptual and practical framework for web-based processes in multi-agent systems. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 1464–1465, 2004.
- [207] Michael Graham Moore. *Handbook of Distance Education*. Lawrence Erlbaum Associates, 2003.
- [208] J. Mostafa, W. Ke, and Y. Fu. Automated text classification using a multi-agent framework. In *Fifth ACM/IEEE-CS Joint conference on Digital libraries (JCDL '05)*, pages 157–158, 2005.
- [209] D. G. Moursund. *Introduction to Information and Communication Technology in Education*. David Moursund, 2005.
- [210] D. Nagel. 2know classroom response system gets short answer feature. Technical report, Campus Technology, 2009.

- [211] P. Noerr. 99 questions about digital libraries, vala - libraries, technology and the future inc.
- [212] P. Noerr. Digital library toolkit, sun education and research.
- [213] The Commonwealth of Learning. An introduction to open and distance learning.
- [214] The University of Wisconsin-Extension. Highlight history of extension in wisconsin 1862 to 1999," the university of wisconsin-extension.
- [215] James Ong and Sowmya Ramachandram. *Intelligent Tutoring Systems: Using AI to improve Training Performance and ROI*. Stottler Henke Associates Inc., 2003.
- [216] J. Owen and M. Raj. *BPMN and Business Process Management*. Popkin Software, 2003.
- [217] N. Pal and D. Pantaleo. *The Agile Enterprise: Reinventing Your Organization for Success in an on Demand World*. Springer, 2005.
- [218] Alexandros Paramythios and Susanne Loidl-Reisinger. Adaptive learning environments and e-learning standard. *Electronic Journal on e-Learning*, 2(1), 2004.
- [219] A. S. Patel and K. Bernard. *Intelligent Tutoring: from SAKI to Byzantium*. MCB UP Ltd, 2001.
- [220] Helmut Petritsch. Service oriented architecture vs. component based architecture.
- [221] L. M. Pittenger. The it organization of the future, people strategies. Technical report, CMIT, 2001.
- [222] S. Pleisch and A. Schiper. Approaches to fault-tolerant and transactional mobile agent execution-an algorithmic view. *ACM Computing Surveys (CSUR)*, Volume 36 Issue 3, ACM Press, pages 219–262, 2004.
- [223] J. T. Pollock. *Semantic Web for Dummies*. Wiley Publishing, 2009.
- [224] J. Poozhikunnel. Developing a simple service oriented architecture.
- [225] SUNGARD Higher Education Press. The new york times company and epsilen collaborate to support advances in e-learning. Technical report, SUNGARD Higher Education Press., 2009.
- [226] C. A. Ptak. *ERP: Tools, Techniques, and Applications for Integrating the Supply Chain*. CRC Press, 2nd edition, 2003.
- [227] R. Qui. *Enterprise Service Computing: From Concept to Deployment*. IGI Global, 2007.

- [228] A. M. Riad. euniversities: An intelligent system for e-learning. *Egyptian Informatics Journal, Faculty of Computers and Information, Cairo University*, pages 38–52, 2005.
- [229] A. M. Riad and H.A. Ali. New approach for information retrieval base-on mobile agent. *Egyptian Computer Journal, ISSR*, 10(1):110–123, 2002.
- [230] A. M. Riad and Haitham A. El-Ghareeb. A service oriented architecture to integrate web services and software agents in course management systems. *Egyptian informatics Journal, Cairo University*, 8(1), 2007.
- [231] A. M. Riad and Haitham A. El-Ghareeb. Aligning service oriented architecture and business process management systems to achieve business agility. *Egyptian informatics Journal, Cairo University*, 9(1), 2008.
- [232] Margaret Rouse. Enterprise architecture.
- [233] S. Roy, S. Dasgupta, and N. Mukherjee. A multi-agent framework for resource brokering of multiple concurrent jobs in grid environment. In *Proceedings of The Fifth International Symposium on Parallel and Distributed Computing (ISPDC'06)*, pages 329–336, 2006.
- [234] S. Russell and P. Norvig. *Artificial Intelligence: Modern Approach*. Prentice Hall, 2003.
- [235] B. H. Sreenivasa Sarma and B. Ravindran. Intelligent tutoring system using reinforcement learning to teach autistic students.
- [236] D. Schaffhauser. iparadigms updates peer review app. Technical report, Campus Technology., 2009.
- [237] M. Schoeman and E. Cloete. Architectural components for the efficient design of mobile agent systems. In *annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology SAICSIT '03*, pages 48–58, 2003.
- [238] H. Sharifi and Z. Zhang. Agile manufacturing in practice application of a methodology. *International Journal of Operations and Production Management*, 21:772–94, 2008.
- [239] P. Shodjai. Web services and the microsoft platform, 2006.
- [240] Charles A. Shoniregun and Sarah-Jane Gray. Is e-learning really the future or a risk? Technical Report 10, Ubiquity, 2003.
- [241] A. Sinha. Client-server computing. *Communications of the ACM*, 35:77–98, 1992.
- [242] Aaron Smith. Twitter update. Technical report, Pew Internet and American Life Project, 2011.

- [243] H. Smith and P. Fingar. *Business Process Management: The Third Wave*. Meghan Kiffer Press, 2006.
- [244] Ian Sommerville. Restful web services.
- [245] Ian Sommerville. Service-oriented and component-oriented approaches.
- [246] Ian Sommerville. *Software Engineering 9*. Pearson, 2011.
- [247] D. Sprott. A web service maturity model, cbdi forum.
- [248] Natalia Stash. Incorporating cognitive/learning styles in a general-purpose adaptive hypermedia systems. *SIKS*, 2007.
- [249] David J. Steinhart. Summary street: An intelligent tutoring system for improving student writing through the use of latent semantic analysis. *Faculty of The Graduate School of the University of Colorado*, 2001.
- [250] A. Stone, J. Briggs, and C. Smith. Sms and interactivity – some results from the field, and its implications on effective uses of mobile technologies in education. In *IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02)*, 2002.
- [251] Moon Ting Su, Chee Shyang Wong, Chuak Fen Soo, Choon Tsun Ooi, and Shun Ling Sow. Service-oriented e-learning system. *Information Technologies and Applications in Education (ISITAE)*, 2007.
- [252] K. Syeara. Multi-agent systems and applications. In *9th ECCAI Advanced Course, ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School (EASSS), Prague, Czech Republic*, pages 17–45, 2001.
- [253] G. Tennant. *Six Sigma : SPC and TQM in Manufacturing and Services*. Ashgate Publishing, 2001.
- [254] R. Thierauf and J. Hoctor. *Optimal Knowledge Management: Wisdom Management Systems Concepts And Applications*. Idea Group Publishing, 2006.
- [255] Effraim Turban. *Decision support systems and intelligent systems*. Pearson, 2005.
- [256] L. T. Victoria. Ict in education", asia-pacific development information programme.
- [257] J.E. Villaverde, D. Godoy, and A. Amaldi. Learning styles' recognition in e-learning environments with feed-forward neural networks. *Journal of Computer Assisted Learning*, 22(3):197–206, 2006.
- [258] S. Virdhagriswaran, D. Osisek, and O'Connor. *Standardizing agent technology*. ACM Press, 1995.
- [259] R. Vokurka and G. Fliedner. *The journey toward agility*. MCB UP Ltd, 1998.

- [260] J. Vrettaros, G. Vouros, and A. Drigas. Development of a diagnostic system of taxonomies using fuzzy logic – case solo (useful for e-learning system. *WSEA Transactions on Information Science and Applications*, 1(6), 2004.
- [261] J. Waldo, G. Wyant, A. Wollrath, and S. Kendall. A note on distributed computing. Technical report, Sun Microsystems Laboratories, 2004.
- [262] Webopedia. Webopedia: Mis.
- [263] Dunwei Wen and Fuhua Lin. Ways and means of employing ai technology in e-learning systems. In *icalt, Eighth IEEE International Conference on Advanced Learning Technologies*, pages 1005–1006, 2008.
- [264] S. White. *Introduction to BPMN*. IBM, 2004.
- [265] D. Whitelock and S. Wheeler. Alt-c 2006: The next generation. In *Edinburgh : Heriot-Watt University, The next generation: 13th International Conference*, 2006.
- [266] Wikipedia. Enterprise architecture.
- [267] Wikipedia. Zachman framework.
- [268] L. Wilkes and L. Wilkes. Service oriented architecture - considerations for agile systems. *Microsoft Architect Journal*, 2:11–23, 2004.
- [269] M. Wooldridge. *Introduction to MultiAgent Systems*. Wiley, 2002.
- [270] Rogers Y., Price S., Fitzpatrick G., Fleck R., Harris E., Smith H., Randell C., Muller H., O’Malley C., Stanton D., Thompson M., and Weal M. Ambient wood: Designing new forms of digital augmentation for learning outdoors. In *IDC ACM Maryland, USA, ACM*, 2004.
- [271] H. EL Yamany, M. Capretz, and L. Capretz. A multi-agent framework for testing distributed systems. In *30th Annual International Computer Software and Applications Conference (COMPSAC’06)*, pages 151–156, 2006.
- [272] Zhengfang Xu Zheng Yin and Abdulmotaleb El Saddik. A web services oriented framework for dynamic e-learning systems. In *Canadian Conference on Electrical and Computer Engineering; CCECE/CCGEI*, 2003.
- [273] M. Yost. *Extracting Data From Integrated Student Information Systems*. Eric, 1983.
- [274] O.R. Zaiane. Building a recommender agent for e-learning systems. In *Proceedings of International Conference on Computers in Education*, pages 55–59, 2002.
- [275] H. Zhang and V. Lesser. A queuing theory based analysis of an agent control mechanism in peer-to-peer information retrieval systems. In *Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*, pages 17–24, 2005.

- [276] X. Zhou, J. Li, H. Shen, M. Kitsuregawa, and Y. Zhang. *Frontiers of www research and development*, springer.
- [277] I. Zennikus, G. Benguria, B. Elvesæter, and J. Vayssi‘ere. *A Model Driven Approach to Agent-Based Service-Oriented-Architectures*. ATHENA, European Integrated Project, 2002.

# **Index**

Enterprise Architecture, xxi, 1, 7–9, 11, 13, 20, 23, 25, 52, 56, 204, 227, 228

Enterprise Information System, xxii, 1, 3, 6, 79–81, 84, 87, 94

Expert System, 5

Group Support Systems, 3, 6

Knowledge Management System, 5, 6, 21, 134

Management Support System, 1

Service Oriented Architecture, xxi–xxiv, 13–20, 29, 48, 50, 52, 55–59, 63, 64, 66, 68