

Dedication

To my Parents, Wife and Kids

Mirghani

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my supervisor, Dr Mohamed Awad , Head of Computer Systems and Networks, College Of Computer Science and Information Technology, His wide knowledge and his logical way of thinking have been of great value for me, His understanding , encouraging and personal guidance have provided a good basis for the present thesis.

Also I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I want to thanks Sudatel for giving me permission to utilize the IT infrastructure, to do the necessary research work and to use their data.

Especially, I would like to give my special thanks to my wife whose patient enabled me to complete this work.

ملخص

إن التطوير في عالم البرمجيات يتزايد طبقاً لمقتضيات التطور في مجالات الأعمال حيث أن بعض التطبيقات للباحثين في مجالات برمجيات الأعمال والإدارة يمكن أن يكون بعض جزئيات البرمجة قد تم تطويرها من قبل في بحوث وتطبيقات أخرى، ولقيمة الوقت وتوفير الجهد اللازم وتقادياً لإعادة تطوير تلك الجزئيات البرمجية والتي تسمى بالمكونات (Components) تم إيجاد الية برمجية لإعادة استخدام نفس تلك المكونات لتطبيقات أخرى تحتوي على جزئية من نفس الوظيفة للتطبيق.

تقوم الشركة السودانية للاتصالات (سوداتل) باستخدام نظام فوترة وخدمات المشتركين وهو النظام قيد بحثنا هذا ، حيث أن سوداتل قامت باستخدام الأنظمة التالية لمواقع خدمات المشتركين :-

بي سي سي 1000 : نظام تم تطويره داخل بيئة سوداتل وتم تطبيقه وإستخدامه في مواقع خدمات المشتركين لسوداتل لخدمة ما لا يقل عن 144000 مشترك.

بي سي سي 2000: تم تطويره داخلياً وتطبيقه في مواقع على سبيل المثال موقع خدمات مشتركين الخرطوم وبعض المواقع الإقليمية حيث أنه يخدم ما لا يقل عن 248000 مشترك.

كلا النظامين أعلاه تمت برمجتهم باستخدام تقنية ذات الهيكل الثنائي (2-Tier Architecture) حيث أنه تم استخدام لغة الأكسس لبرمجة واجهة المستخدم والشاشات وتم استخدام ال MS SQL كقاعدة بيانات، وبمتابعة الأداء للنظام لوحظ بعض المشاكل والتي تنجم عن استخدام تقنية ال 2-Tier وهي ضعف أداء البرنامج (Performance) مما تسبب في إختناقات لقاعدة البيانات عند تنفيذ الأوامر وتحديث بشكل محدد عند دخول عدد كبير من المستخدمين في نفس الوقت على النظام والذي بدوره ينتج عن تأخر إكمال المعالجة للعمليات المنطقية والحسابية.

ولحل المشكلة تناولنا في بحثنا هذا الية تطبيق تقنية ال 3-Tier على نظام خدمات المشتركين ، حيث أنها تستخدم في حالة وجود عدة مخدمات متوزعة في مواقع مختلفة وكان ذلك باستخدام برمجة المكونات (COM+) وبمقارنة الأداء لوحظ التحسن في الأداء وإمكانية إعادة استخدام نفس المكونات (Components) في برامج أخرى تحتوي على جزئية برمجية سبق إعدادها، هذا بالإضافة لسهولة الصيانة وإكتشاف وحل المشاكل أثناء عمل البرنامج ودون الحاجة لتوقفه، وبتلك التقنية المستخدمة تمكنت سوداتل بسهولة من إدخال عدد أكبر من مواقع خدمات المشتركين ضمن المنظومة (3-Tier).

كدراسة حالة في بحثنا هذا قمنا بإعادة تصميم نظام خدمات المشتركين ذو تقنية ال 2-Tier وتطبيق نظام خدمات المشتركين (3-Tier using COM+) والذي بدوره تمكن من إستيعاب عدد أكبر من المشتركين دون التأثير على أداء وسرعة تنفيذ الأوامر حيث وصل العدد الى 560000 مشترك.

Abstract

An increasing number of applications is developed. These applications which support the researcher in management and analysis of data often contain parts which can be found in other applications in a similar form. To save time and development effort, existing components of other applications could be reused when a new application is developed. A module which is used for the composition of a new application should consist of the business logic, the database and the component interface.

Sudan Telecom (Sudatel) billing environment was used as a case study in this thesis and it was running the following 2 billing systems:

BCC 1000: developed in-house, deployed in regional Sites and Supporting 144,000 subscribers.

BCC 2000: developed in-house, deployed in Khartoum and some regional sites supporting 248,000 subscribers.

Both above mentioned billing systems was built on 2 – Tier Architecture (MS-Access as Front-End and SQL Server as Back-End) On the other hand, the problem arises when the performance of the existing billing systems is largely affected by the number of concurrent users, resulting in a major impact on transaction processing time.

We used 3-tier architecture as a solution , usually the three tier architecture is used when an effective distributed Client/Server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures.

The goal of this thesis is to implement a 3-Tier Architecture billing system using COM+ technology as a middleware to resolve the performance bottleneck and benefit

from the 3-tier architecture features, the solution (case study) was implemented in Sudatel and the reengineered (3-Tier) billing system is supporting 560,000 subscribers.

List of Figures

Figure No.	List of figure	Page No.
Figure 2.1	Multitier Application	4
Figure 3.1	Visual Basic Form	19
Figure 3.2	Visual Basic Toolbox	19
Figure 3.3	Visual Basic Property Sheet	19
Figure 3.4	CORBA Wizard	24
Figure 3.5	Tree- View of COBRA Naming Services	24
Figure 3.6	COREBA Deploy Tool	25
Figure 4.1	Component Designer	29
Figure 5.1	Non Client – Server System	34
Figure 5.2	Traditional Host system with LAN	35
Figure 5.3	Distributed Data Centers	36
Figure 5.4	Client \ Server 3- Tier Model	37
Figure 5.5	Distributed Client \ Server Model	39
Figure 5.6	Inter-Process Communication	40
Figure 5.7	Client \ Server 2- Tier Architecture	42
Figure 5.8	Typical 3- Tier Architecture	44
Figure 5.9	Middleware	45
Figure 6.1	Different Views of Interface Pointers	47
Figure 6.2	Threading Models	47
Figure 6.3	COBRA Architecture	50

Figure 6.4	Java RMI architecture	51
Figure 6.5	Interception Approach	54
Figure 6.6	Com Architecture	57
Figure 6.7	possible case for tread proliferation	65
Figure 7.1	Win32 system Mechanism	74
Figure 7.2	DLL loading	75
Figure 8.1	Information Flow in Accounting Process	84
Figure 8.2	Telecom. Management Network	85
Figure 8.3	TMN Physical Architecture	87
Figure 8.4	Functions of Mediation	92

Table of Contents

Dedication		I
Acknowledgement		II
Arabic abstract		III
Abstract		IV
List of Figures		VI
Table of Contents		VIII
<u>Chapter One</u>	<u>Introduction</u>	1
1.1	Background, Motivations	1
1.1.1	Software Components	1
1.1.2	Hierarchical Composition	2
1.2	The objectives of this Thesis	3
<u>Chapter Two</u>	<u>Background & Theory</u>	4
2.1	Distributed Application Deign	4
2.1.1	Multilayer Design	4
2.1.2	Benefits of Multilayer Designs	5
2.2	Software Components	7
2.3	Component Development Process	8
2.3.1	Design Stage	8
2.3.2	Development Stage	9
2.3.3	Assembly Stage	9
2.3.4	Deployment Stage	10
2.4	Existing Component Models	10

2.4.1	CORBA Component Model	10
2.4.2	Enterprise Java Beans	11
2.4.3	COM / DCOM / COM+	12
2.4.4	Dot NET Framework	14
2.5	Project BCC Middleware (COM+)	15
<u>Chapter Three</u>	<u>Software Development Tools</u>	18
3.1	Visual Basic	18
3.2	Visual Studio .NET	20
3.3	C++ Builder / Delphi	20
3.4	NetBeans IDE	21
3.4.1	Editor Module and Java Module	21
3.4.2	Form Editor Module	22
3.4.3	CORBA Module	22
3.5	EJB Deploytool	23
<u>Chapter Four</u>	<u>Visual Development</u>	26
4.1	Requirements	26
4.2	Proposed Concepts	27
4.2.1	Component Designer	28
4.2.2	Assembly Tree View:	30
4.2.3	Property Sheets	30
4.2.4	Component Description File	31
4.2.5	Primitive Components	32
4.2.6	Packaging	32

<u>Chapter Five</u>	<u>Host Systems</u>	34
5.1	Distributed Systems	36
5.2	Client/Server Model	37
5.3	Distributed Client/Server Model	38
5.4	Inter-process Communication	40
5.5	Benefits of the Client/Server Model	41
5.6	Client/Server 2-Tier Architecture	41
5.7	2-Tier Considerations	42
5.8	Two -Tier Pros and Cons	43
5.9	Three-Tier Client/Server Architecture	44
5.10	Middleware	45
<u>Chapter Six</u>	<u>Threading Models and Apartments (COM)</u>	46
6.1	Apartments – Protection Domain	46
6.1.2	Threading Models – Synchronization	47
6.1.3	Marshaling Interface Pointers	49
6.1.4	Standard Marshaling and Custom Marshaling	49
6.2.1	CORBA	50
6.2.2	JAVA/RMI	51

6.2.3	Comparison	52
6.3	Fault Tolerant Middleware	53
6.3.1	Eternal–Interception CORBA	53
6.3.2	COM on Multicast Transport	54
6.3.3	COM Architecture	56
6.3.4	Challenges	58
6.3.5	Design of the system	59
6.4	Overall Design	59
6.4.1	Replicating Instantiation	62
6.4.2	Multicast Invocation	63
6.4.3	Return Value Handling – Result Iteration	66
6.4.4	Unresponsiveness Handling	68
<u>Chapter Seven</u>	<u>Middle Tier Implementation details</u>	71
7.1	Extending Win32 System	72
7.1.1	The Win32 Systems and DLL	72
7.1.2	Extending Win32 system	74
7.1.3	Coexistence with Standard system	75
7.2	Instrumentation of Win32 APIs	76
<u>Chapter Eight</u>	<u>Sudatel BCC System – Middleware(Case Study)</u>	79

8.1	Scope and Contents	80
8.2	Charging Principles	80
8.2.1	Tariffs	81
8.2.2	Regulation	82
8.2.2.1	Charging Principles in PTN	82
8.3	Accounting Process	83
8.3.1	TMN	84
8.3.2	Mediation Systems	90
8.3.2.1	Functionality	91
8.3.3	Billing	94
8.3.3.1	Billing Requirements	94
8.3.3.2	Billing Functionality	95
8.3.4	Interfaces	96
8.4	Interconnect Billing	97
8.5	Charging for SS7 Signaling	98
8.6	Applying Accounting Model to Internet	99
8.7	Enterprise Component Services	100
<u>Chapter Nine</u>	<u>Conclusion & Recommendation</u>	103
9.1	Conclusion	103
9.2	Recommendation and Future Works	104
<u>References</u>		106
<u>Appendix</u>		109

A	Glossary	109
B	BCC-Middleware Component Services	112
C	BCC Modules Structure	113
D	BCC Methods and Functions	114