Journal of The Korea Society of Computer and Information
Vol. 23 No. 4, pp. 81-88, April 2018
www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.04.081

Design & Implementation of Enhanced Groupware Messenger

HyungSoo Park*, HoonKi Kim**, WooJong Na***

Abstract

In this paper, we present some problems with the Groupware Messenger functionality based on dot net 2.0 and implement a new design structure to solve them. They include memory leakage, slow processing, and client window memory crash. These problems resulted in the inconvenience of using instant messaging and the inefficient handling of office tasks. Therefore, in this paper, instant messaging functionality is implemented according to a new design architecture. The new system upgrades dot net 4.5 for clients and deploys the new features based on MQTT for the messenger server. We verify that the memory leak problem and client window memory crash issues have been eliminated on the system with the new messenger functionality. We measure the amount of time it takes to bind data to a set of messages and evaluate the performance, compared to a given system. Through this comparative evaluation, we can see that the new system is more reliable and performing.

▸Keyword: Groupware, Messenger, Dot Net, MQTT, Memory Leakage & Crash

I. Introduction

IT의 통합 솔루션의 핵심 구성 요소인 그룹웨어는 조직의 정보화를 위한 중요한 도구이며 조직 내의 효율적인 업무 처리를 위해 대부분의 회사에서 도입되어 사용되고 있는 실정이다[1]. MS사의 dot net 기술을 이용한 서버-클라이언트 기반 그룹웨어 솔루션, 자바를 기반으로 하는 웹 기반 그룹웨어 솔루션, ASP 기반의 비즈메카 그룹웨어 솔루션 개발 제품들이 등장하기 시작하였고 구축 사례들도 보고되었다[2-4]. 또한, 그룹웨어 기술들도 초기의 이메일만 사용하던 간단한 협업 소프트웨어 기술이 통합 커뮤니케이션과 같이 유무선과 소프트웨어가 결합된 형태의 아주 복잡한 기술도 등장하게 되었다[5].

최근에는 클라우드 기술 중에 하나인 SaaS 방식을 활용한 효과적인 그룹웨어 구축 방식에 대한 연구가 이루어지고[6], 스마트 폰 앱 기반의 모바일 그룹웨어 시스템이 다양한 분야에 활용되고 있다[7,8]. 그룹웨어의 다양한 기능에 대한 로그를 통합하여 조직 내의 업무 효율성과 안전성을 증대하고자 하는 연구도 진행되었다[9].

그룹웨어 사용 실태 조사 결과에 따르면, 그룹웨어를 사용 중인 515개사의 70%의 기업이 사용 중인 그룹웨어의 개선이 필요한 것으로 응답했다[1]. 개선이 필요한 이유는 '제한적인 기능, 브라우저의 호환성 제약, 잦은 오류, 제한된 모바일 기능, 협업 기능의 제한' 등이다. 가장 많이 이용하는 기능은 메일, 전자결재, 메신저, 게시판, 보고서, 업무 기능 등이다.

그룹웨어의 기능들 중에 메신저 기능은 사내 또는 잦은 출장으로 회사 조직 내 그룹원들 간의 긴급한 업무 내용을 공유하며 업무가 원활하고 효율적으로 진행되기 위해 반드시 필요한 중요한 기능이라고 할 수 있다. 따라서 메신저 기능의 끊김없는 안정화된 서비스가 반드시 필요하다. 이에 본 논문에서는 먼저 본 논문에서 사용된 그룹웨어와 타사 그룹웨어 솔루션과의 비교 분석을 진행하고[10], 본 논문에서 사용된 그룹웨어를 소개한다. 기존 그룹웨어의 메신저 기능 문제점을 도출하여 이를 분석하고, 이를 해결하기 위한 새로운 설계를 제안하고, 이 설계 문서를 기반으로 개발 구현한다. 현재 상용 운용하고 있는 회사에 적용하여 일정 기간을 모니터링하여 기존 문제점이 개선되었는지를 확인하도록 한다. 또한, 성능 측면에

• First Author: HyungSoo Park, Corresponding Author: HoonKi Kim
*HyungSoo Park (hspark@dongyang.ac.kr), Dept. of Computer Software Engineering, Dongyang Mirae University
**HoonKi Kim (kimhk@dongyang.ac.kr), Dept. of Computer Software Engineering, Dongyang Mirae University
***WooJong Na (solo89@itcrew.co.kr), ITCrew ltd.
• Received: 2018. 01. 18, Revised: 2018. 02. 20, Accepted: 2018. 03. 08.
• This work was supported by Dongyang Mirae Univ. Research Grant.

있어서 어느 정도 향상되었는지를 신구 시스템들 간 데이터 바인딩 시간 측정을 토대로 비교 평가해 본다.

II. Preliminaries

1. Related Study of Groupware

본 절에서는 본 논문에서 사용된 그룹웨어와 국내의 대표적인 두 개의 그룹웨어들을 서로 비교 분석해 보고 비교 분석 시, 각 제품의 사양과 기능 및 장단점에 대해서 정리해 본다. 해당 내용은 I사에서 제공한 자료를 기반으로 하고 있다[10]. 표 1은 본 논문에서 개선된 그룹웨어와 국내의 두 개 회사의 그룹웨어들을 비교한 것이다.

그룹웨어들 간의 사양, 기능 등은 대동소이하다고 볼 수 있다. 해당 그룹웨어 도입 시, 비용 측면에서 본 논문의 그룹웨어가 유리하다고 볼 수 있다.

2. Introduction of Groupware

본 논문에서 사용되고 있는 그룹웨어는 여러 개의 서버와 클라이언트, 다양한 기능들로 구성되어져 있다. 그림 1은 시스템 구조를 나타낸 것으로, 그룹웨어 시스템은 4개의 서버, 웹서버, 메일서버, DB서버, 메신저 서버와 PC 및 스마트폰인 클라이언트로 구성된다.

Table 1. Comparison of Groupware Solutions

Company / Classification	I	B	T
Product Information	Enterprise / Lease Groupware	Enterprise Groupware	Enterprise / Non-Profit Groupware Enterprise SNS/Lease Groupware
OS	Windows Server 2008	MS/Linux	Windows Server 2008
DB	MS SQL 2008 or later	MS SQL 2005/Oracle	MS SQL 2008 or later Mail(Charge), TOMCAT(Free)
Mail	.NET	JAVA	.NET
Browser	Cross Browser	IE 8~10	Cross Browser
.NET	Framework 4.0		Framework 4.0 (2006)
Deployment Type	Deployment/Service	Deployment/Service	Deployment/Service
Basic Functions(Free)	- Electronic Signature - Internal Communications - Collaboration Management - Business Report - Data Room Board - Announcement - Web Hard - Address List/SMS - Absenteeism and tardiness Management - Messenger - Mobile - TODO/Memo	- Portlet-based Portal available - Bulletin Board - Community(CoP) - Schedule Management (w/ Google Map and Schedule) - Electronic Signature (Work-Flow) - E-Mail - Document Management - Survey Management - Data Room Board - Resource Management - Address List (w/ Outlook) - Organizational Chart Management - System Environment Settings	- Electronic Signature - Announcement - Document Management - Schedule Management - Messenger - Video Conference/Face to face Signature - Resource Management - E-Mail - ERP Inter-Operability
Option(Charge)	- E-Mail - ERP Inter-Operability - Premium Web Editor	-Knowledge Management -EDMS	-E-Mail -ERP Inter-Operability -Video Conference/Face to face Signature, Mobile APP
Advantage	1) Developing based on a basic Group ware 2) Separate Community to Support Collaboration 3) Support for Customized Electronic Signature based on Work-flows 4) Web Standard Compliance and Latest UI/UX Support 5) Scalable Mobile Solution (Hybrid App) 6) Highly Qualified with Experience 7) Experienced ERP Inter-working and Customizing Know-how 8) Multilingual Support 9) Support for Seamless Maintenance	1) Support Korean electronic Signature culture (Work-Flow) 2) Support for a variety of Organizational Collaboration Cultures 3) Inter-working with other systems 4) Open Standard System Architecture 5) User-friendly interface 6) Multilingual Support 7) Solution GS Certification 8) Fully Inter-working with Google Map and Schedule 9) Cross Browsing 10) Multi Client OS Support 11) Multiple OS,RDBMS,WAS Support	1) Dynamic Inter-working with The Zone ERP 2) Wide variety of customer 3) Support for a variety of Organizational Collaboration Cultures 4) Inter-working with other systems 5) User-friendly interface 6) Multilingual Support 7) Cross Browsing 8) Multi Client OS Support 9) Multiple OS,RDBMS,WAS Support
Weakness	Company Size	High Cost of Purchase	Slow Drive Expensive Customization
Mobile Type	Hybrid APP	Hybrid APP	Cross Browser+ Mobile APP

www.kci.go.kr

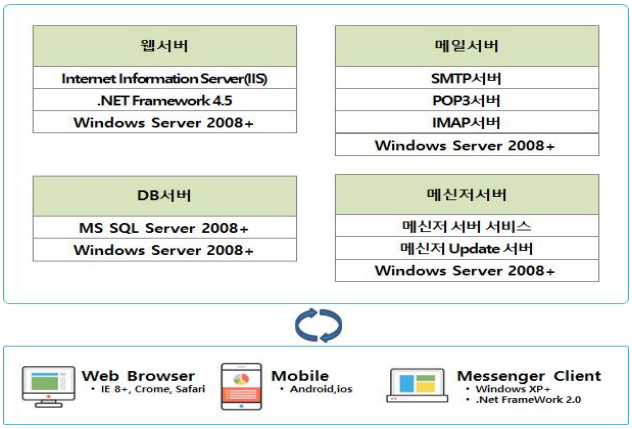

Fig. 1. Groupware System Architecture

Fig. 2. Groupware Functional Diagram

그림 2는 그룹웨어의 기능 구조를 나타낸 것으로 각 기능에 대한 설명은 아래와 같다.

- 전자결재 기능 : 조직 내에 결재할 내용을 작성하고 진행하며 공문 발송하고 문서를 이관 처리한다.
- 메신저 기능 : 새로운 정보를 실시간으로 알려 주고 대화방을 통해 구성원 간 의사소통을 진행하며 긴급한 의사결정에 효율적이며 쪽지 정보를 전달 공유한다.
- 전자메일 기능 : 메일을 주고받으며 외부 메일 서버와 연동하여 통합 관리한다.
- 자료실 기능 : 자료를 등록 관리하며 자료에 대한 통계를 통해 자료의 활용도를 확인한다.
- 공지게시 기능 : 조직 내에 공지 게시할 내용을 작성하고 공지게시하며 공지 게시 내용에 대한 통계를 통해 활용도를 확인한다.
- 근태관리 기능 : 구성원들의 근태 처리를 하며 연차 및 근태 현황을 관리한다.
- 업무 관리 : 조직원들의 각자 업무를 관리하며 일정 및 자원을 관리하여 업무를 효율적으로 처리한다. 매주/매월 주간/월간 보고를 처리한다.
- 기타 기능 : 개인/공용 웹하드 관리, 시스템 관리, 모바일 앱을 통한 연동 처리

3. The Existed Messenger Server

기존 메신저 서버는 자체적으로 구현된 소켓을 통해 메신저 기능을 수행하고 개발 환경은 dot net 2.0 기반이며 서버와 클라이언트 간에 주고받는 프로토콜은 비동기 소켓 기반에서 자체 개발한 것이다. 메신저 시스템에서 사용하는 OS와 인터페이스는 표 2와 같다.

Table 2. Messenger Development Environment

Classification	Usage Technologies
Client	.NET Framework 2.0
	Windows XP
Server	.NET Framework 2.0
	MS-SQL Server
	Windows 2008 R2 or later
Common	Asynchronous Socket (Self-Implemented)

그림 3은 메신저 시스템 구조로서 클라이언트는 외부망인 인터넷을 통해 메신저 서버에 접속하며 메신저 서버는 내부망을 통해 DBMS에 접속하여 필요한 정보를 검색 및 변경하게 된다.

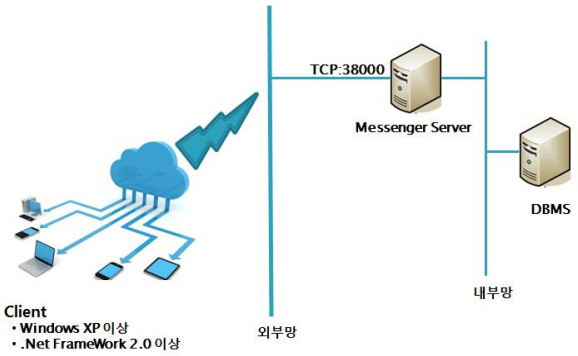

Fig. 3. Messenger Architecture

그림 4는 메신저 시스템 내의 프로세스 절차를 표현한 것으로, 클라이언트에서는 비동기 소켓을 통해 모든 요구 사항들과 메시지를 메신저 서버에게 전송하고 그에 대한 응답을 받는다. 메신저 서버는 각 기능을 수행하는 4개의 쓰레드가 존재한다.

LoginEventThread는 로그인/로그아웃, 세션 관리 기능을 수행한다. PrimaryEventThread는 쪽지 및 대화 송수신 기능을 수행하고 SecondaryEventThread는 파일 전송 처리와 시스템 메시지 수신 처리 기능을 담당한다. ExtraEventThread는 관리자용 이벤트를 처리한다.

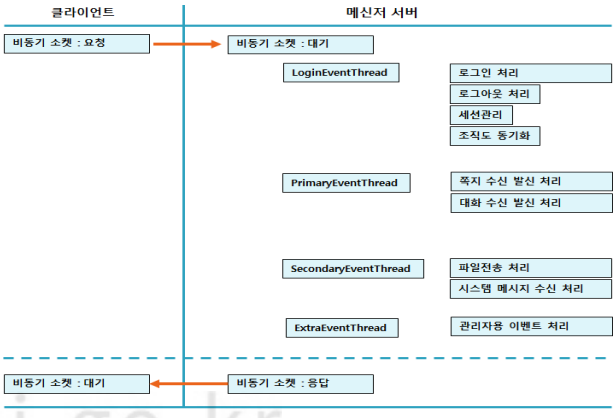

Fig. 4. Messenger Process Procedure

www.kci.go.kr

4. Messenger Problems

이 절에서는 기존 메신저 시스템에서 발생한 문제점들에 대해서 정리하고자 한다. 현재까지 확인된 문제점들로는 메신저 서버 메모리 누수, 처리속도 저하, 클라이언트 윈도우 메모리 충돌 발생 등이 있다. 이로 인해 메신저 기능은 시스템 부하를 가중시켜 자체 기능이 행업되는 결과를 초래하는 것은 물론, 간헐적으로 시스템 전체를 마비시킬 수도 있다.

4.1 Memory Leakage of Messenger Server

기존 그룹웨어 시스템에서 메신저 서버 실행 시간이 길어질수록 할당 메모리를 반환하지 못하고 메모리가 계속 누적 되는 문제점이 발생하는 것을 확인할 수 있다. 메신저 서버의 사양은 메모리 용량이 8G의 HP DL320이다.

Table 3. Memory Usage per Week in the Existed Messenger

Day	Memory Usage
first	20 MB
second	158 MB
third	354 MB
fourth	563 MB
fifth	676 MB
sixth	869 MB

표 3은 매일 메모리 사용량을 나타낸 것으로 메신저 최초 실행 시에는 20MB 내외 사용되는데 매주 100~200MB 정도의 메모리가 누수 되는 현상을 확인할 수 있다.

4.2 Slow Processing

기존 그룹웨어 시스템에서 메신저 서버 메모리 누수 문제와 더불어 쪽지 목록 등 대량 데이터 처리 시 속도가 저하되어 발송이 지연되거나 알림 통보가 늦어 새로운 쪽지가 수신되었어도 인지하지 못하거나, 실제 데이터베이스와 클라이언트 화면이 불일치되는 경우가 발생하여 구성원이 새로운 쪽지를 인지하지 못하는 경우도 발생할 수 있다. 이는 실시간으로 처리되어야 하는 메신저 기능과 쪽지 기능에 영향을 미쳐 긴급한 업무처리에 있어 상당한 애로사항을 발생한다.

4.3 Client Window Memory Crash

해당 문제는 텍스트 입력 및 표시 컨트롤 RichTextBox와 윈도우 메모리 충돌이 발생하는 문제이다. 특정 PC에서 대화 진행 시 지속적으로 발생한다. 그림 5는 0으로 나누려고 할 때의 로그를, 그림 6은 보호된 메모리를 읽거나 쓰려고 할 때의 로그를, 그림 7은 창 핸들을 만드는 동안 오류가 발생할 때의 로그를 수집한 것이다.

```
위치: System.Windows.Forms.UnsafeNativeMethods.CallWindowProc(IntPtr
      wndProc, IntPtr hWnd, Int32 msg, IntPtr wParam, IntPtr lParam)
위치: System.Windows.Forms.NativeWindow.DefWndProc(Message& m)
위치: System.Windows.Forms.Control.DefWndProc(Message& m)
위치: System.Windows.Forms.Control.WndProc(Message& m)
위치: System.Windows.Forms.TextBoxBase.WndProc(Message& m)
위치: System.Windows.Forms.RichTextBox.WndProc(Message& m)
위치: System.Windows.Forms.Control.ControlNativeWindow.OnMessage
      (Message& m)
위치: System.Windows.Forms.Control.ControlNativeWindow.WndProc
      (Message& m)
위치: System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg,
      IntPtr wparam, IntPtr lparam)
```

Fig. 5. Memory Crash Log: in case of deviding by 0

```
위치: System.Windows.Forms.UnsafeNativeMethods.CallWindowProc(IntPtr
      wndProc, IntPtr hWnd, Int32 msg, IntPtr wParam, IntPtr lParam)
위치: System.Windows.Forms.NativeWindow.DefWndProc(Message& m)
위치: System.Windows.Forms.Control.DefWndProc(Message& m)
위치: System.Windows.Forms.RichTextBox.WndProc(Message& m)
위치: System.Windows.Forms.Control.ControlNativeWindow.OnMessage
      (Message& m)
위치: System.Windows.Forms.Control.ControlNativeWindow.WndProc
      (Message& m)
위치: System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg,
      IntPtr wparam, IntPtr lparam)
```

Fig. 6. Memory Crash Log: in case of trying to read or write protected memory

```
위치: System.Windows.Forms.NativeWindow.CreateHandle(CreateParams cp)
위치: System.Windows.Forms.Control.CreateHandle()
위치: System.Windows.Forms.Control.get_Handle()
위치: System.Windows.Forms.Control.get_WindowText()
위치: System.Windows.Forms.Control.set_CacheTextInternal(Boolean value)
위치: System.Windows.Forms.Control.PaintWithErrorHandling(PaintEventArgs e,
      Int16 layer, Boolean disposeEventArgs)
위치: System.Windows.Forms.Control.WmPaint(Message& m)
위치: System.Windows.Forms.Control.WndProc(Message& m)
위치: System.Windows.Forms.ScrollableControl.WndProc(Message& m)
위치: System.Windows.Forms.Control.ControlNativeWindow.OnMessage
      (Message& m)
위치: System.Windows.Forms.Control.ControlNativeWindow.WndProc
      (Message& m)
위치: System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg,
      IntPtr wparam, IntPtr lparam)
```

Fig. 7. Memory Crash Log: If an error occurred while creating the window handle

세 가지의 메모리 충돌하는 경우는 프로그램 실행 시 사용되는 heap 및 stack 영역, OS 영역의 보호된 메모리를 잘못 읽거나 쓰려고 하는 경우에 발생한다. 이로 인해 메신저는 아무런 동작을 하지 않거나 메신저 기능을 수행하는 화면 창이 행업되거나 사라진다.

III. The Proposed Architecture

Ⅱ장에서 제시된 문제점들을 개선하기 위해서는 기존 메신저 시스템

www.kci.go.kr

의 구조를 변경하여야 하고, 향후 모바일 메신저 개발을 위해서 현재 소켓 연결 방식 대신에 경량화된 소켓과 프로토콜을 사용하여야 한다. 이를 위해서 메신저 서버에서는 MQTT(Mosquitto) 프로토콜을 사용해야 하고 클라이언트에서는 닷넷 4.5 업그레이드가 필요하다.

1. Design and Implementation

본 논문에서 제안하고자 하는 메신저 시스템 구조는 그림 8과 같고 메신저 서버가 MQTT 서버와 WAS로 분리되어 구현되어졌으며 그 외에는 기존 메신저 구조와 동일하다.

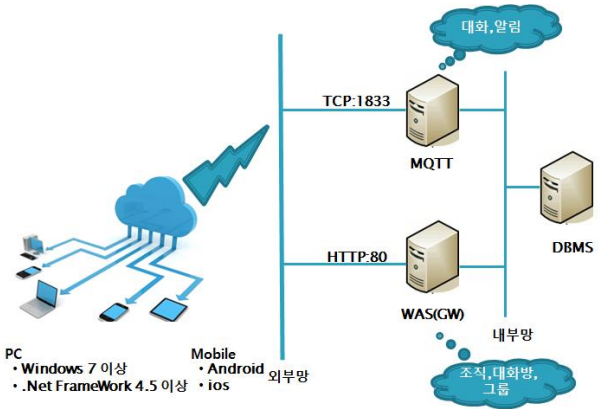

Fig. 8. The Proposed Messenger Architecture

Table 4. The Proposed Messenger Development Environment

Classification	Usage Technologies
Client	.NET Framework 4.5
	C# WPF MVVM(MVVM light)
	M2MQTT.Net
	SQLite
	Window 7
Server	MQTT v3.1/v3.1.1 Broker
	ASP.NET MVC4 Json
	Windows 2008 R2 or later

표 4는 메신저 시스템에서 사용하는 OS와 인터페이스를 나타낸 것으로 OS는 기존 메신저와 동일하며 인터페이스에 있어서 MQTT 프로토콜을 추가 사용하였으며 .NET을 4.5로 업그레이드하여 사용하였다.

메신저 시스템의 세부 기능에 대한 처리 절차는 세 가지 절차들이 있다. 먼저, 로그인 처리 절차는 그림 9와 같으며, 클라이언트 User1이 WAS(Web Application Server)에 로그인 신청을 하면 WAS는 로그인 절차를 수행하고 MQTT 서버는 User1이 정상적으로 로그인되었음을 User2에게 알린다.

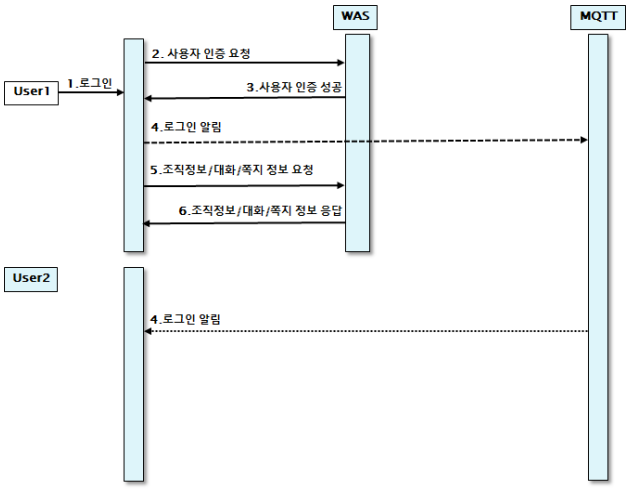

Fig. 9. Log-In Process Procedure

다음으로, 대화 처리 절차는 클라이언트 User1이 WAS에 대화 요청을 하면 WAS는 대화 처리 절차를 수행하고 MQTT 서버는 User1으로부터 메시지가 정상적으로 수신되었음을 User2에게 알리고 User2는 메시지를 WAS로부터 읽고 이에 대한 처리 내용을 MQTT 서버에게 알려준다. 그림 10은 대화 처리 절차를 나타낸 것이다.

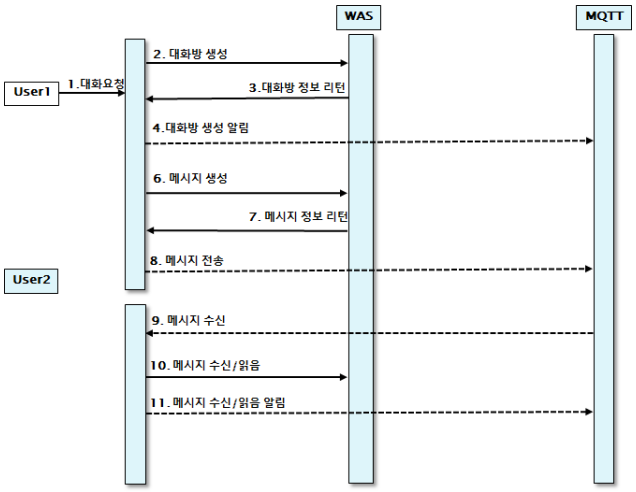

Fig. 10. Talk Process Procedure

마지막으로, 쪽지 처리 절차는 클라이언트 User1이 쪽지를 작성하여 WAS에게 쪽지를 전송하면 WAS는 쪽지 처리 절차를 수행하고 MQTT 서버는 User1으로부터 쪽지를 정상적으로 수신하고 MQTT서버는 User2에게 쪽지 수신을 알리고 User2는 WAS에게 쪽지정보를 요청하고 해당 정보를 받는다. 그림 11은 쪽지 처리 절차를 나타낸 것이다.

www.kci.go.kr

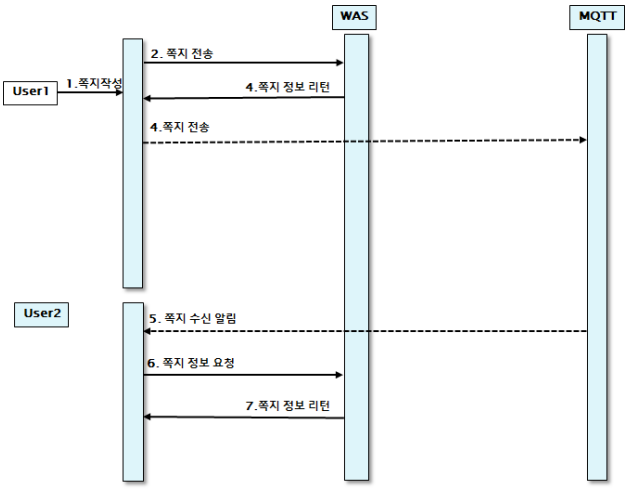

Fig. 11. Instant Message Process Procedure

그룹웨어 메신저 기능 개선에 있어서 반응형 웹 시스템 또는 모바일 웹앱 시스템에 비해 다음과 같은 장점들이 있다.

- 기술 측면 : 서버는 MQTT 등 최신기술을 활용하여 최소 사양에 최대의 성능으로 구현하였으며, 클라이언트는 WPF Framework을 활용하여 디자인과 개발이 분리된 구조로 구현되었다.
- 업무 측면 : 하나의 서버를 통하여 PC메신저와 모바일메신저와 통합 개발되었고 서버의 로직을 공유하여 구현하였다. 이로 인해 PC와 모바일 간에 동기화가 잘 처리되어 데이터 불일치 제거되었다.
- 확장 측면 : 통합 Push Framework로 구현되어 업무적으로 알림이 필요한 부분에 대하여 통합적인 알림을 구현하여 안정적이며 신속한 Communication 채널이 구현되었다.

Fig. 12. Screen of the Proposed Messenger

그림 12는 본 논문에서 제안된 메신저의 화면을 나타낸 것으로 로그인 화면과 대화 리스트 화면과 대화창이다.

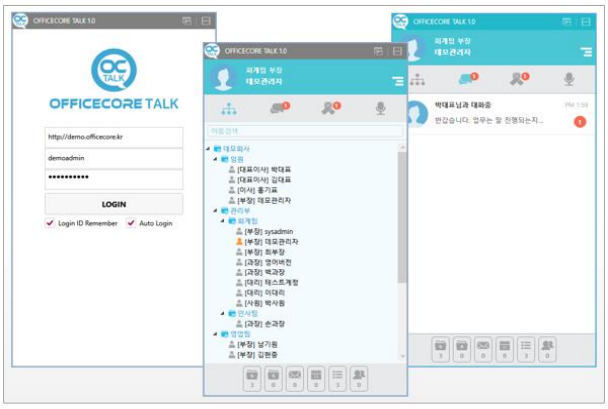

Fig. 13. Instance Message Window of the Proposed Messenger

그림 13은 본 논문에서 제안된 메신저의 쪽지 기능 창을 나타낸 것으로 대화를 하면서 실시간으로 쪽지 내용을 확인할 수 있다.

2. Deployment and Result

본 논문에서 제안된 신규 메신저 시스템이 기존 문제점들을 해결할 수 있는지를 확인하기 위해 기존 상용 시스템에 적용하였다. 신규 시스템을 적용한 시스템은 '(주)아**앤컴퍼니'이고 클라이언트 수는 400명이다. 상용 적용 후 1개월 동안 메모리 누수 문제가 있는지를 모니터링하고 클라이언트 윈도우 메모리 충돌이 발생하는지를 확인하였다. 성능 측면에 있어서는 쪽지 데이터를 수신하여 데이터를 바인딩하는데 걸리는 시간을 기존 시스템과 제안된 신규 시스템에서 측정하고 비교 분석하였다.

2.1 Memory Leakage improvement

메모리 누수 문제에 대해서는 상용 적용 후 일주일마다 메모리 사용량 측정을 통해 개선되었는지를 확인하였다. 메모리 사용량은 MQTT 서버와 WAS에서 확인하였으며 큰 변화 없이 안정적으로 운용됨을 확인할 수 있었다. 표 5는 메모리 사용량을 일주일 단위로 모니터링한 결과이다.

Table 5. Memory Usage per Week in the Proposed Messenger

Day	MQTT Server Memory Usage	WAS Memory Usage
First (6-Dec.)	1.5 MB	203 MB
Second (13-Dec.)	1.5 MB	173 MB
Third (20-Dec.)	1.5 MB	181 MB
Fourth (29-Dec.)	1.5 MB	166 MB

2.2 Client Window Memory Crash improvement

클라이언트 윈도우 메모리 충돌은 서식있는 텍스트 입력 및 표시 컨트롤을 하는 RichTextBox 핸들러에 의해 발생하는 것을 확인하였고 이를 개선하기 위해 단순화된 대화 로직 및 윈도우 컨트롤 핸들러 TextBox로 변경하여 처리하고 응용 계층

www.kci.go.kr

에서 예외 처리를 해 줌으로써 클라이언트 윈도우 메모리 충돌은 더 이상 발생하지 않음을 확인할 수 있었다.

2.3 Performance improvement

신구 메신저 시스템 간 처리 속도 성능 측정은 쪽지 데이터를 수신하여 데이터를 바인딩하는데 걸리는 시간을 기준으로 비교 분석하였다. 각 시스템에서 3번에 걸쳐 쪽지 데이터에 대해 바인딩을 시도하였다. 표 6은 두 시스템의 바인딩 시간을 나타낸 것이다.

기존 메신저에서는 바인딩하는데 걸리는 시간이 평균 0.5초 소요되었고 신규 메신저에서는 최대 0.1초~ 최소 0.01초로 처리 속도가 개선됨을 알 수 있었다.

Table 6. the time to bind an instance message (ms)

Number of Attempts	the Existed Messenger	the Proposed Messenger
1	677 ms	101 ms
2	485 ms	58 ms
3	513 ms	15 ms

그림 14는 기존 메신저에서, 그림 15는 본 논문에서 제안된 메신저에서 쪽지 데이터를 바인딩할 때 로그를 캡쳐한 것이다. 바인딩 시간 측정은 로그 데이터의 시작(Start)과 종료(End) 시각의 차를 계산한 것이다. 데이터 바인딩 후에 데이터를 읽어 화면에 출력하는 부분에 대해 성능 측정을 시도해 보았다. 이전 시스템과 개선된 시스템 사이에 큰 차이가 없음을 확인하였다.

```
2018-01-08 14:01:13,259 GetMemoBox Start
2018-01-08 14:01:13 936 ResultGetMemoBox End
2018-01-08 14:01:31,569 GetMemoBox Start
2018-01-08 14:01:32 054 ResultGetMemoBox End
2018-01-08 14:01:33,743 GetMemoBox Start
2018-01-08 14:01:34 256 ResultGetMemoBox End
```

Fig. 14. Client Log Data of the Existed Messenger

```
2018-01-08 13:16:40,837 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST Start
2018-01-08 13:16:40,938 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST End
2018-01-08 13:17:43,719 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST Start
2018-01-08 13:17:43,771 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST End
2018-01-08 13:17:43,821 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST Start
2018-01-08 13:17:43,836 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST End
2018-01-08 13:18:28,410 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST Start
2018-01-08 13:18:28,486 [9] DEBUG OcTalkClient.ViewModel.ViewModelBaseOC - SERVER_GET_MEMO_LIST End
```

Fig. 15. Client Log Data of the Proposed Messenger

IV. Conclusions

그룹웨어는 모든 기업 내에서 업무를 효율적이고 신속하게 처리할 수 있는 수단을 제공하고 기업 내 자원을 효율적으로 관리하는데 큰 기여를 하고 있다. 그룹웨어에는 다양한 기능이 있는데, 본 논문에서는 그 기능들 중에 두 번째로 많이 사용되고 있는 메신저 기능에 대해 검토 분석해 보고 해당 기능에 어떤 문제점들이 있는지를 파악하고 이를 해결하기 위해 기존 메신저 시스템을 어떻게 개선하였는지를 확인하였다.

먼저, MQTT 기반으로 서버와 클라이언트 간 메시지를 주고받을 수 있도록 변경하며 또한 닷넷 4.5로 업그레이드를 통해 시스템을 보다 안정화시켰다. 새로운 설계 구조에 의해 구축된 메신저 시스템을 기존 상용 시스템에 적용하고 모니터링하여 기존 문제점들이 해결되었는지를 확인해 보았다.

메모리 누수에 있어서는 1개월 동안 해당 시스템을 모니터링하여 메모리가 증가되지 않음을 확인할 수 있었다. 또한, 성능 측면에서는 쪽지 목록을 클라이언트에 바인딩하는 시간을 측정하여 기존 시스템에 비해 최소한 5배 이상 향상되었음을 알 수 있었다. 클라이언트 윈도우 메모리 충돌에 있어서도 클래스 내의 사용하는 멤버 함수를 대체하여 더 이상 문제가 발생되지 않을 확인하였다. 현재, 기존 모든 상용 시스템에 신규 메신저 기능을 업그레이드 한 상태이며 고객들로부터 좋은 반응을 피드백 받고 있다.

향후, 그룹웨어의 메신저는 기업에서만 사용되는 특화된 기능이므로 기능 측면에서 카카오특과의 비교 분석을 통해 카카오톡만의 장점 기능들을 수용하여 보다 향상된 메신저가 될 수 있도록 연구개발이 필요할 것이다.

REFERENCES

[1] Groupware Usage Survey Results, *http://gamefocus.co.kr/detail.php?number=39761,* 2014.

[2] ITCrew ltd., "Officecore - Solution proposal for collaboration and communication," Jan. 2017.

[3] Byung Kwon Kwak, Seung-Ho Lee, Jong Gyun Lee, Jong-Soo Song, "Web-based Groupware Solution," Journal of The Korea Information Processing Society, Vol. 6, No. 3, pp. 101-109, May. 1999.

[4] Joon Ho Yoon, Sun Young Cho, Won Suk Choi, "The Present and Future of Groupware ASP Businesses - Focused on Biz meka Groupware," Journal of The Korea Information Processing Society, Vol. 10, No. 6, pp.35-41, Nov. 2003.

[5] Chang Je Cho, "Groupware Technology," Comm -unications of the Korean Institute of Information Scientists and Engineers, Vol. 25, No. 8, pp.22-25, Aug. 2007.

[6] Hyun-Sun Ryu, Neung-Hoe Kim, Dong-Hyun Lee, Hoh Peter In, "An Empiricial Study on Factors Affecting the Adoption of Software as a Services(SaaS) for Integrating

www.kci.go.kr

Group of Companies," Prodeedings of Korea Multimedia Society, pp. 652-655, Nov. 2010.
[7] Chang Gi Kim, Lee Sang Do, Jeong Dong Seo, Jeong Min Seo, "A Study of Smart Phone Based Social Service Groupware System," Proceedings of the Korean Society of Computer Information Conference, Vol. 22, No. 2, pp. 443-445, Jul. 2014.
[8] Yong gu Ji, "A Study of the Work Efficiency and Job Satisfaction in the Utilization of Mobile Groupware," Graduate School of Communication and Information Dongguk University, 2016.
[9] soon-young Hwang, "Increase efficiency and safety by merging log groupware integration plan," Proceedings of the Korea Information Science Society, pp. 1145-1147, Jun. 2016.
[10] ITCrew ltd., "Comparision of Officecore and Other Groupware Solutions" Dec. 2017.

Authors

Hyung Soo Park received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 1992, 1995 and 2008, respectively Dr. Park joined a researcher of R&D Center at LGE, ltd. and ntelia, ltd. Anyang, Korea, in 1995 and 2004. He is currently a Professor in the Department of Computer Software Engineering, Dongyang Mirae University. He is interested in mobile network, internet and mobile computing, and Information Security.

Hoon Ki Kim received the B.S., M.S. and Ph.D. degrees in Electronic Engineering from Hanyang University, Korea, in 1988, 1990 and 2002, respectively After receiving his M.S. degree, he was with Core Network Research Labs., LG Electronics Inc., Korea, from 1990 to 2001. He has been engaged in research of switching, CDMA cellular, PCS, IMT-2000, NGN system. Dr. Kim joined the faculty of the Department of Computer Science at Dongyang Mirae University, Seoul, Korea, in 2001. He is currently a Professor in the Department of Computer Science, Dongyang Mirae University. He is interested in embedded system, communication software and wireless communication networks.

Woo Jong Na received the M.S. degrees in Management Information System from SUNGKYUNKWAN University, Korea, in 2001. Mr. Na joined a Developer of Development Center at Hyundai Information Technology, Korea, in 1996. He is currently a CEO in the ITCrew ltd. He is interested in Collaboration Tool and Package S/W Solution for company.

www.kci.go.kr