한국지리정보학회지 12권 1호 : 26~35(2009)

디자인 패턴을 활용한 LBS기반 모바일 시스템 구현*

이홍로[1※] · 백정호[2] · 문영채[2]

Implementation of Mobile System based on LBS using Design Pattern*

Hong-Ro LEE[1※] · Jeong-Ho BAEK[2] · Young-Chae MUN[2]

요 약

본 논문은 LBS기반 모바일 시스템 구현을 위하여 GoF의 디자인 패턴을 활용한 효율적인 설계 방법을 제안하고자 한다. 본 논문에서 제안된 설계기법은 사용자와 모바일 시스템 관점에서 디자인 패턴의 생성패턴, 구조패턴, 행위패턴을 이용하여 모바일 시스템 인터페이스를 설계하고 구현한다.

디자인 패턴을 활용한 설계 방법은 소프트웨어 개발에 있어서 모듈 및 소프트웨어 재사용성 증가, 개발시간을 단축, 유지보수 최소화, 시스템의 메모리 절감 효율성을 보여준다. 따라서 본 논문은 모바일 환경에서 지리정보시스템을 접목한 미들웨어를 개선하기 위해 디자인 패턴을 활용하였다.

본 논문은 모바일 시스템에 디자인 패턴을 적용하여 높은 안정성과 신뢰성을 보여주고, 모듈 및 소프트웨어의 재사용과 확장성을 높이며 효율적인 시스템을 개발하는데 기여할 것이다.

주요어 : 디자인 패턴, 지리정보시스템, 위치기반서비스, UMPC

ABSTRACT

This paper will propose the efficient method that should be designed to utilize the design pattern of GoF(Gang of Four) for implementation of LBS(Location Based Service) mobile system. The design method proposed by this paper can design and implement system interface using creational patterns, structural patterns and behavioral patterns of design pattern from the user and mobile system viewpoint.

The design method using design pattern can show increasing reusability of module and software, abridging the development time, minimizing the maintenance and decreasing the efficiency of memory of system to software development. Therefore, we used a design pattern in order to change the middleware applying geographic information system from mobile environment.

2009년 1월 22일 접수 Received on January 22, 2009 / 2009년 3월 1일 수정 Revised on March 1, 2009 / 2009년 3월 5일 심사 완료 Accepted on March 5, 2009

* 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신 인력양성사업으로 수행된 연구결과임.

1 군산대학교 컴퓨터정보공학과 부교수 Dept. of computer Information Engineering, Kunsan University

2 군산대학교 컴퓨터정보공학과 박사과정 Dept. of computer Information Engineering, Kunsan University

※ 연락저자 E-Mail : leehongro@kunsan.ac.kr

Finally, the mobile system applied to design pattern shall contribute to showing high stability and reliability, improving reusability and scalability of module and software and developing efficient LBS mobile system.

KEYWORDS : Design Pattern, GIS, LBS, UMPC

서 론

현재까지 객체지향 설계유형에서 디자인 패턴을 이용하여 시스템을 설계 및 효율성을 분석하는 연구들이 진행되고 있다(B.Schulw 등, 1998; T.H. Ng와 S.C. Cheung, 2005). 객체지향 디자인 패턴 설계 유형을 적용하여 지오센서 모델 설계 및 인터페이스 구축 그리고 인터페이스의 연산 효율성을 비교하였다(백정호와 이홍로, 2006). 또한 디자인 패턴 정보의 시각화와 디자인 패턴 개체의 구조적인 속성들을 자동으로 검사하는 연구가 제안 되었다(Chengwan He 등, 2008). 디자인 패턴을 응용하여 소프트웨어의 재사용성과 품질 향상을 목적으로 Shopping Mall Component를 설계하였다(Jing Gang Chu와 Jia Chen, 2008). 뿐만 아니라 디자인 패턴들의 전개를 통하여 클래스와 객체간의 효율성을 분석 및 평가하였다(T.H. Ng와 S.C. Cheung, 2005).

그리고 유비쿼터스 기술과 GIS, 위치기반서비스(LBS : Location Based Service)등을 활용하여 PDA, UMPC(Ultra-Mobile Personal Computer) 등과 같은 모바일 장치를 이용한 연구가 진행되고 있다(김종수 등, 2006; 이기영 등, 2002).

그러나 모바일 장치에 시스템을 구축하는데 객체지향 설계 방법을 적용한 사례는 매우 미비한 실정이다. 모바일 장치에 적용하는 소프트웨어는 메모리의 한계를 극복하고, 개발의 안정성과 재사용성을 증가시키기 위해 체계적인 분석과 설계 방법을 이끌어 낼 수 있어야 한다. 그러나 하드웨어의 성능을 무시하고 특정 목적을 달성하기 위해 설계된 소프트웨어는 시스템에 적용될 때 하드웨어의 모든 성능을 차지하게 되는 문제가 생긴다. 따라서 체계적인 분석과 디자인 패턴 설계를 효율적으로 시스템에 적용하여 시스템의 오류 발생 비율을 줄이고 분석 능력을 향상시키는 것이 필요하다(김운용과 최영근, 2002).

본 논문에서는 객체지향 설계를 위해 디자인 패턴을 이용한 모바일 시스템을 구현하였다. 소프트웨어 개발의 안정성과 재사용성, 메모리의 한계를 극복하기 위한 효율적인 접근 방법은 객체지향 설계 유형의 디자인 패턴을 적용하여 객체지향 시스템을 보다 효율적으로 설계하고 구축하는 것이다. 이러한 설계 방법을 적용함으로써 요구사항 변경을 사전에 예방하여 개발시간을 단축하고, 시스템의 오류 발생 비율을 줄이고 분석능력을 향상 시킬 수 있다.

본 논문은 모바일 장치에 시스템 구현을 위하여 GoF(Gang of Four) 디자인 패턴 중 생성패턴, 구조패턴, 행위패턴 3가지 측면에서 6가지(Abstract Factory, Prototype, Facade, Proxy, Command, Mediator pattern)를 활용하여 설계하였으며, 구현된 모바일 시스템을 Quest Software사의 JProbe 툴을 이용하여 실행시간에 따라서 메모리 성능과, 시스템의 효율성을 보여준다.

본 논문은 객체지향 디자인 패턴 기반 모바일 시스템 모델을 설계, 객체지향 모바일 시스템 구현, 성능평가 및 검토, 결론 및 향후 연구방향으로 구성되었다.

LBS기반 모바일 시스템 모델 설계

본 논문에서는 GIS기반 GPS를 이용한 농작물 작황 조사 모바일 시스템 구축 내용을 기반으로 LBS기반 모바일 시스템을 설계하고, 구현하였다(문영채와 이홍로, 2008).

1. LBS기반 모바일 시스템 개요

LBS기반 모바일 시스템은 현장에서 사용자가 모바일 장치를 이용하여 현장의 농작물 중 벼의 생육 정보 및 생산 정보를 조사하고, GPS를 이용하여 사용자의 위치 정보 및 벼의 위치 정보를 수치지도 상에 보여주고, 조사된 농작물 정보와 위치정보를 서버에 있는 DB에 저장하는 시스템이다. 사용자의 권한별로 DB의 접근을 제한 할 수 있는 사용자관리 모듈, 작물의 정보를 DB에 삽입/삭제/수정/검색 할 수 있는 작물 관리 모듈, 수치지도 상에 사용자와 벼의 위치정보를 보여주는 맵 모듈, GPS로부터 수신된 위치정보를 변환 하기위한 위치정보관리 모듈, GPS와 UMPC, UMPC와 DB간 데이터 전송 및 수신을 위한 통신 모듈 등 5가지로 구성되어 있다(문영채와 이홍로, 2008).

2. 디자인 패턴을 이용한 LBS기반 모바일 시스템 설계

2.1 Abstract Factory 패턴

Abstract Factory 패턴은 새로운 구현 내용이 추가 된다 하더라도 새로운 하위 클래스만 정의해서 사용하면 기존 소스코드의 변경을 최소화 하면서 새로운 구현 내용을 적용할 수 있게 된다.

본 연구에서 사용자관리를 위한 설계의 경우 그림 1과 같이 UserFactory 클래스를 FarmManager 클래스, FieldManager 클래스, AllUser 클래스로 구분하여 UserFactory 클래스의 하위 클래스로 두고, 시스템 내에서는 UserFactory의 자료형과 인터페이스를 사용하게 만든다. 이 같은 구조에서 DB에 접근 할 때 ConnectDB 클래스를 정의하고, OracleDB 클래스, MssqlDB 클래스, OtherDB 클래스를 ConnectDB 클래스의 하위 클래스로 추가로 정의해서 사용하면 기존 소스코드의 분석이나 수정 없이 독립적으로 실행이 가능한 객체들을 생성할 수 있게 된다.

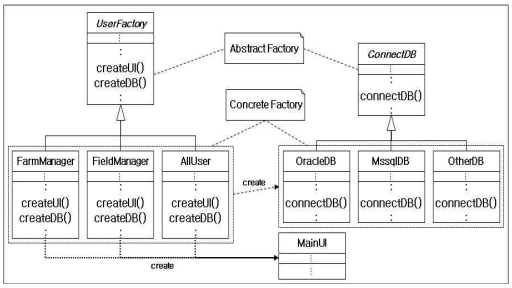

FIGURE 1. 사용자관리를 위한 Abstract Factory 패턴

2.2 Prototype 패턴

Prototype 패턴은 기존의 객체를 복제해서 새로운 객체를 생성한다. 이 경우 새롭게 생성되는 객체는 기존의 객체와 동일한 자료형을 가지기 때문에 객체를 생성할 때마다 객체의 자료형을 정해줄 필요가 없다. 따라서 자연스럽게 생성할 객체의 자료형을 판별하기 위한 비교 문장이 불필요하다. 또한 새로 생성되는 객체는 기존 객체와 동일한 자료형을 가지기 때문에 모든 객체는 자료형이 정해져야 한다는 조건도 만족한다.

그림 2에서 Management 클래스의 하위 클래스들은 모두 clone()이라는 멤버함수를 가지는데 이를 통해 자신과 동일한 객체를 복제해 주게 된다.

본 논문에서 작물관리를 위한 설계의 경우 clone() 멤버함수를 이용하여 객체를 새로 생성하지 않아 메모리의 효율성을 높이고, 재사용성을 높였다.

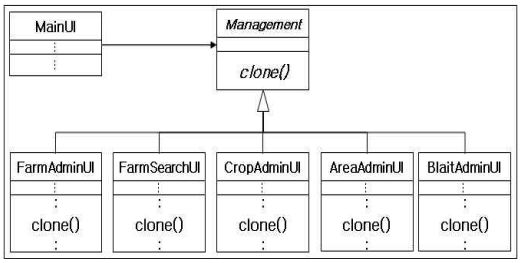

FIGURE 2. 작물관리를 위한 Prototype 패턴

2.3 Facade 패턴

일반적으로 프로젝트를 수행할 때 하나의 목적을 위해 여러 개의 객체들이 유기적으로 연결되게 된다. 이때 객체들을 묶을 수 있다고 판단이 되면, 별도의 객체를 새롭게 만들어 사용자들에게 새롭게 만든 객체의 인터페이스를 이용하여 다른 객체까지 모두 사용할 수 있게 만들 수 있다. Facade 패턴은 대표인터페이스를 이용하여 다른 여러 객체를 사용할 수 있게 만드는 패턴이다.

그림 3은 복잡성을 줄이기 위해서 시스템을 서브시스템으로 구조화 한다. 일반적으로 설계 단계에서의 목표는 서브시스템 간의 연결성과 종속성을 최소화하려는 것이다. 위치정보관리를 위해서 GPS와 연결하고, 좌표를 변환하는 일련의 과정들을 하나의 서브시스템으로 구분하고 PositionManage 클래스로 복잡한 인터페이스를 단순화된 인터페이스로 제공한다.

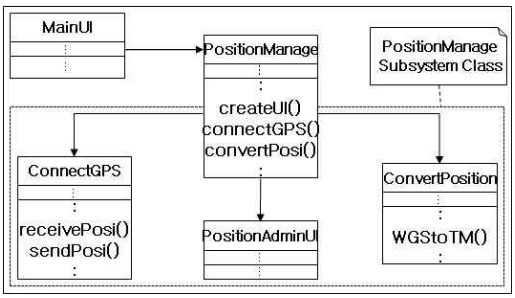

FIGURE 3. 위치정보관리를 위한 Facade 패턴

2.4 Proxy 패턴

시스템이 작업을 처리하는 가운데, 시스템의 처리 과정을 효율적으로 만들기 위해 처리를 분산시킬 필요가 있다. Proxy 패턴은 원하는 작업을 대신해서 처리하도록 만드는 패턴이다. 처리과정의 일부분을 다른 프로세서에 대행시키고자 할 때 Proxy 패턴을 적용하면, 시스템의 구조를 어렵지 않게 효율적으로 변경 시킬 수 있다.

그림 4는 객체를 사용하는 시점까지 생성과 초기화에 드는 비용 및 시간을 연장하기 위해 객체로의 접근을 제어한다.

DBConnect 클래스는 DB로부터 데이터를 불러오는 과정을 대행하는 클래스이다. 그리고 FarmAdminUI, FarmSearchUI, CropAdminUI, AreaAdminUI, BlaitAdminUI 클래스에 공통적인 인터페이스를 정의하고 데이터를 요청하는 곳에 DBConnect 클래스를 사용하여 데이터를 줄 수 있게 한다.

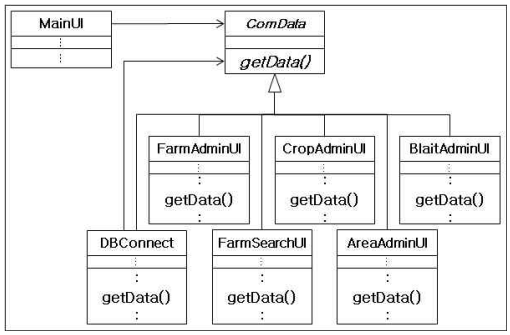

FIGURE 4. DB 데이터 관리를 위한 Proxy 패턴

2.5 Command 패턴

서로 다른 요청을 객체화 하여 클라이언트에게 파라미터로 넘겨줄 수 있게 한다. 요청받은 오퍼레이션이 무엇이며, 이를 처리할 객체가 누구인지에 대한 아무런 정보 없이도 임의의 객체에게 메시지를 보내야 할 때가 있다. 요청과 그 요청을 처리할 객체를 중계하기 위한 클래스 상속 구조를 Command 패턴이라고 한다.

그림 5는 MenuAction 클래스는 요청을 처리할 객체를 내부적으로 저장, 관리하면서 요청이 주어졌을 때 요청을 처리할 객체의 멤버 함수를 불러주는 역할을 한다. 요청을 처리할 객체는 MenuAction 클래스의 객체가 생성되는 시점에 지정되며, MainUI에서 요청을 전달하는 방식은 MenuAction 클래스 및 그 하위 클래스들이 제공하는 공통된 인터페이스를 통해 이루어진다. MenuAction 클래스 및 그 하

위 클래스들이 이처럼 공통된 인터페이스를 제공하는 이유는 MainUI 클래스 입장에서는 구체적으로 어떤 클래스의 객체가 사용되는지 상관없이 다형성을 적용해서 프로그램 동작이 일어날 수 있게 만들기 위해서이다.

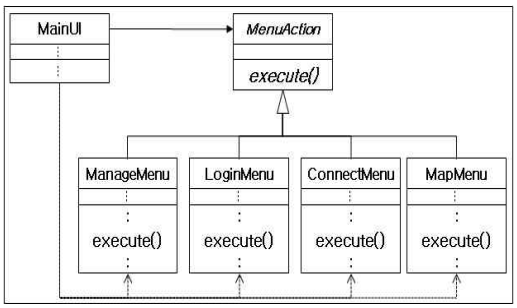

FIGURE 5. UI 메뉴 실행을 위한 Command 패턴

2.6 Mediator 패턴

객체지향 개발 방법론에서는 행위를 여러 객체들에게 분산시켜 처리하도록 권하고 있다. 이러한 행위의 분산으로 객체 모델에 정의한 객체들끼리 수많은 상호작용을 갖게 된다. Mediator 패턴은 각 객체가 관련성을 갖는 다른 객체에 대한 참조 관계를 직접 정의하기 보다는 이를 독립된 다른 객체가 관리하게 한다.

그림 6은 MapMenu 클래스는 하위의 어떤 클래스 객체든지 MainUI 클래스 객체와 연동해서 동작할 수 있게 만들어 주기 위한 것으로 MainUI 클래스 객체가 공통적으로 다룰 수 있는 자료형을 제공해 주는 역할을 한다.

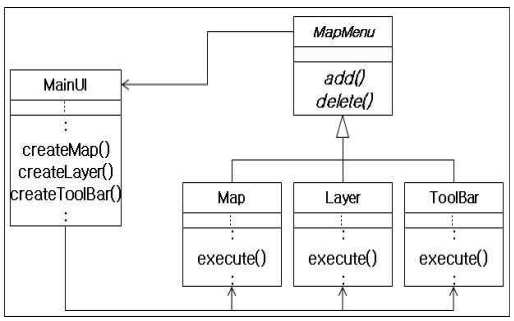

FIGURE 6. 맵 관리를 위한 Mediator 패턴

LBS기반 모바일 시스템 인터페이스 구현

1. LBS기반 모바일 시스템 인터페이스

객체지향 모바일 시스템은 모듈별로 구현하여 하나의 통합적인 시스템을 구현 하였다. 본 시스템은 대용량, 빠른 처리속도, 넓은 화면, 휴대성을 가진 UMPC 기기 환경에서 JAVA 언어를 사용, JBuilder Tool을 이용하여 구현하였고, 맵 컴포넌트는 ESRI사의 Map Objects 2.3 API를 이용하여 구현하였고, Oracle을 이용하여 DB를 구축하였다.

1.1 사용자관리 모듈 구현

사용자관리 모듈은 Abstract Factory 패턴을 적용하여 구현하였다. 이 패턴을 적용함으로써 기존 소스코드의 변경을 최소화하면서 사용자의 권한별로 새로운 구현 내용을 적용할 수 있다. 그리고 DB에 접근할 때 독립적으로 실행이 가능한 객체들을 생성하여 해당 DB환경으로 접근할 수 있다.

사용자관리 모듈 인터페이스는 그림 8에서 Log In 창과 같으며, 아이디와 패스워드를 입력하여 사용자 로그인을 하면 실행화면 상단 제목 표시줄에 사용자의 권한이 보여 지고 권한에 맞는 작업을 할 수 있다.

1.2 작물관리 모듈 구현

작물관리 모듈은 Prototype 패턴을 적용하여 구현하였다. 이 패턴을 적용함으로써 작물관리의 각각의 인터페이스에 있는 공통적인 컴포넌트 객체들을 복제하여 새로 생성하지 않아 메모리의 효율성 및 객체의 재사용성을 높였다.

그림 7은 현장관리자, 작업관리자가 작황정보, 작물정보, 재배정보, 날씨정보, 수량정보, 지역정보, 병충해 방제정보, 위치정보, 사용자정보를 작황 DB에 삽입, 삭제, 수정, 검색을 할 수 있는 인터페이스이다.

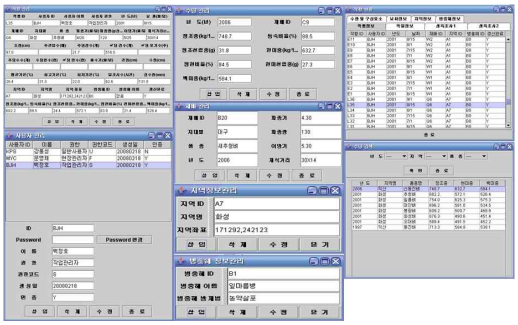

FIGURE 7. 작물관리 화면

1.3 맵 모듈 구현

맵 모듈은 Mediator 패턴을 적용하여 구현하였다. 이 패턴을 적용함으로써 사용자 인터페이스 상에서 Map과 Layer와 Toolbar를 연동하여 공통적으로 다룰 수 있다.

그림 8은 사용자가 메인 화면에 작황의 위치정보나 사용자의 위치정보를 시각적으로 보기 위하여 맵을 추가하거나 삭제 할 수 있고, 맵 상에서 축소, 확대, 이동 등을 할 수 있는 인터페이스이다.

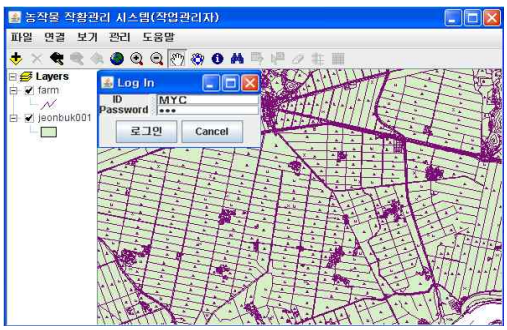

FIGURE 8. 사용자관리 및 맵 모듈 화면

1.4 위치정보관리 모듈 구현

위치정보관리 모듈은 Facade 패턴을 적용하여 구현하였다. 이 패턴을 적용함으로써 GPS 연결 및 GPS로부터 수신된 좌표변환 과정을 하나의 서브시스템으로 구분하여 단순화된 인터페이스로 제공한다.

그림 9는 위치정보관리 모듈 인터페이스로 GPS 연결 설정 후 GPS 연결을 하면 모듈 내에서 좌표변환을 통하여 맵 상에 사용자의 현재 위치를 빨간색으로 칠해진 원으로 표시 되고, GPS 수신좌표와 변환된 좌표를 확인할 수 있다.

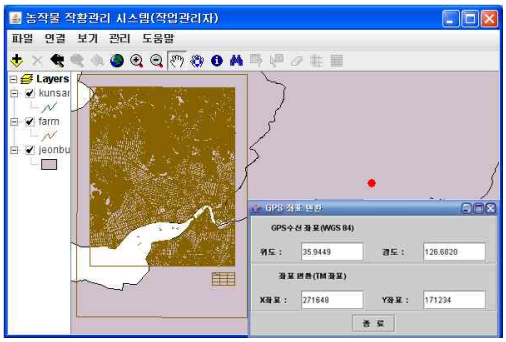

FIGURE 9. 위치정보관리 화면

1.5 통신 모듈 구현

통신 모듈은 DB 설정 및 연결, GPS 설정 및 연결로 구성되었다. DB 설정 및 연결은 Proxy 패턴을 이용하여 DB로부터 데이터를 불러오는 과정을 대행 하는 클래스를 정의하였다. 데이터를 요청하는 클래스에서 대행하는 클래스를 사용하여 데이터를 전달한다. 대행하는 클래스를 사용함으로써 시스템의 구조를 효율적으로 변경 시킬 수 있다.

그림 10은 DB설정 및 GPS설정 인터페이스로 연결 메뉴에 DB설정, DB연결, GPS설정, GPS연결을 이용하여 GPS와 서버에 있는 DB에 연결하여 데이터를 송 · 수신 할 수 있다.

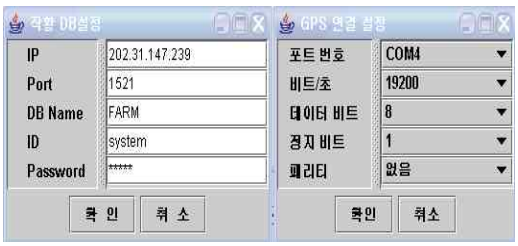

FIGURE 10. 통신모듈 접속설정 화면

성능평가 및 검토

객체지향 모바일 시스템 인터페이스는 표 1의 모바일 환경(UMPC)에서 Java JBuilder로 구현 하였다. 통신환경은 GPS와 UMPC간의 Bluetooth 통신, UMPC와 서버의 DB간의 WLAN 및 TCP/IP 통신을 하였다.

TABLE 1. UMPC 사양

항 목	사 양
OS	Windows XP Home Edition
CPU	Intel Processor STANLEY 800MHz
메모리	1 GB
HDD	60GB(4200rpm)

1. 객체지향 모바일 시스템 인터페이스 성능 평가

본 논문에서 제안하는 LBS기반 모바일 시스템 인터페이스는 GoF의 디자인 패턴 중에서 Abstract Factory, Prototype, Facade, Proxy, Command, Mediator 패턴을 이용하여 모듈을 통합하는 기법으로 GIS 인터페이스, 작황관리, 사용자 위치정보관리를 위한 모듈을 설계하고 구현하였다.

이에 시스템을 설계 및 구현하는데 많은 문제점이 생겼다. 특히, 중복 되는 객체의 활용하는 방법, 복잡한 시스템을 구조화 하는 방법, 서로 다른 객체 간의 중계 방법 등 많은 문제점이 생겼다.

이를 해결하기 위해 기존 소스코드의 변경을 최소화 하면서 새로운 구현 내용을 적용할 수 있는 Abstract Factory 패턴을 사용하였다. 또한 기존의 객체를 복제해서 새로운 객체를 생성하는 Prototype 패턴, 복잡성을 줄이기 위해서 시스템을 서브시스템으로 구조화 하는 Facade 패턴, 처리과정의 일부분을 다른 프로세서에 대행시킴으로써 시스템의 구조를 효율적으로 변경 시킬 수 있는 Proxy 패턴, 객체간의 요청과 그 요청을 처리할 객체를 중계하기 위해 Command 패턴을 사용하였다. 그리고 각 객체가 관련성을 갖는 다른 객체에 대한 참조 관계를 직접 정의하지 않고 독립된 다른 객체가 관리하게 하는 Mediator 패턴을 사용하였다.

본 논문은 디자인 패턴을 적용하지 않은 기존 객체지향 프로그래밍에서 개발한 인터페이스와 디자인 패턴을 적용한 모바일 인터페이스의 연산 효율성을 살펴보았다.

그림 11에서 기존 객체지향 프로그래밍에서 개발한 인터페이스의 연산 개수를 100%라 하였을 때, Abstract Factory 패턴과 Prototype 패턴의 경우 20%, Facade 패턴의 경우 10%, Proxy 패턴의 경우 15%, Command 패턴과 Mediator 패턴의 경우 각각 10%씩의 절감 효과를 보였다. 그리고 백정호 등(2006)은 지오센서 인터페이스 구현에서 디자인 패턴을 적용하여 Abstract Factory 패턴 20%, Composite 패

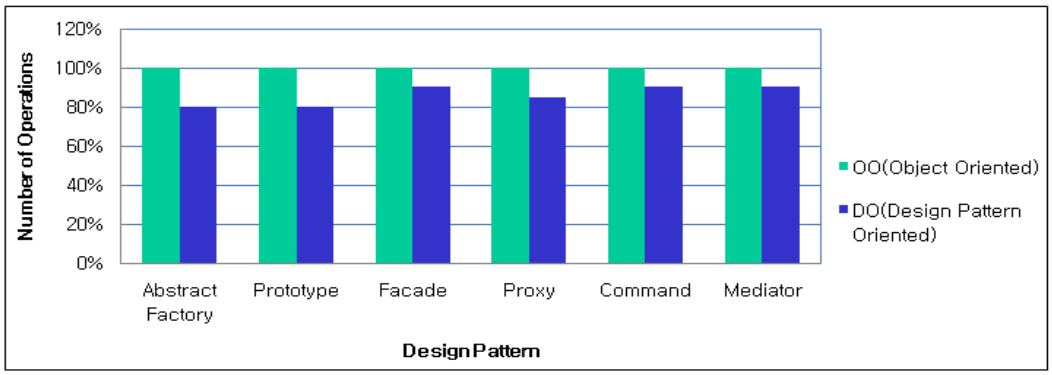

FIGURE 11. 객체지향 모델과 디자인 패턴 모델의 인터페이스 연산 수 비교

턴 5%, Adapter 패턴 0%, Factory Method 패턴과 Template Method 패턴 각각 10%씩의 절감 효과를 보였다(백정호와 이홍로, 2006).

본 논문에서 제안한 모바일 인터페이스와

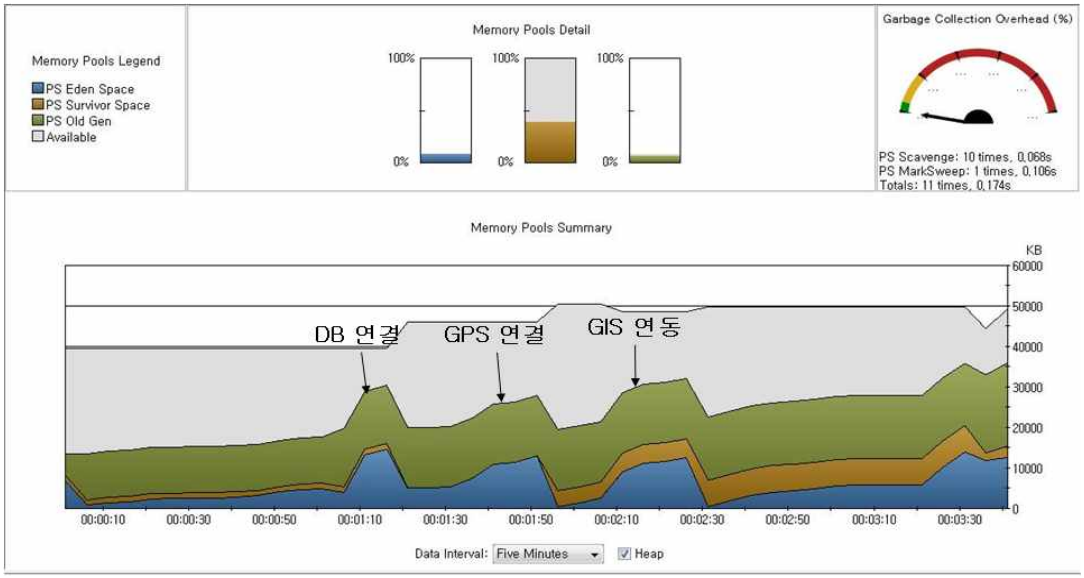

FIGURE 12. 실행 시간에 따른 메모리 성능 비교

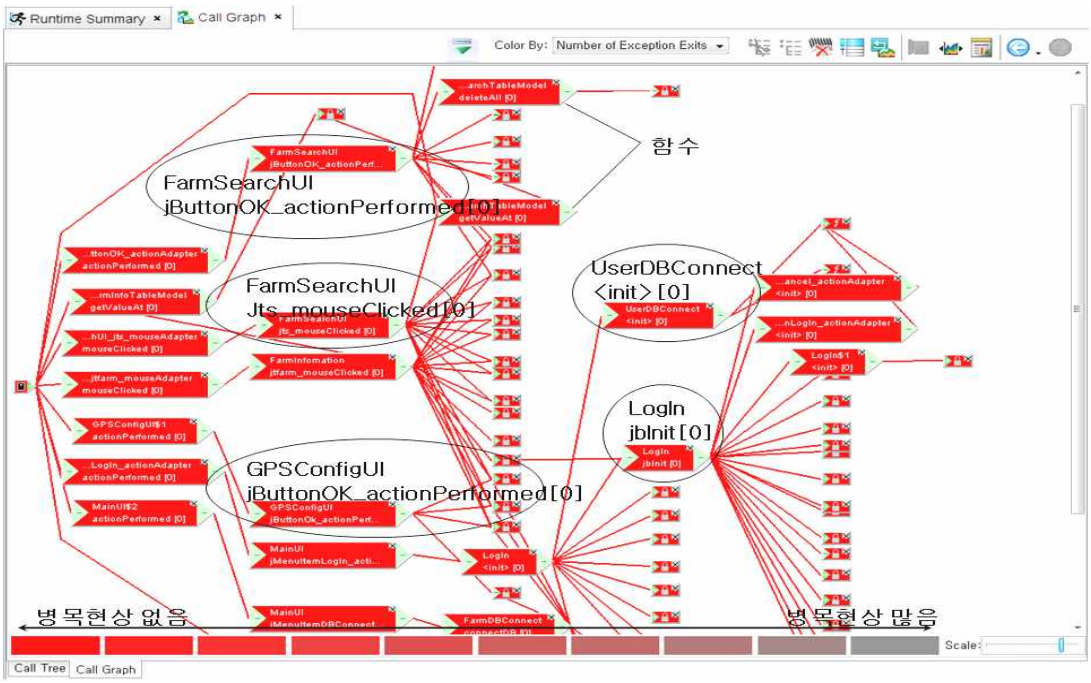

FIGURE 13. 함수 호출 시 예외 발생에 따른 병목현상 상태 그래프

지오센서 인터페이스를 비교하면 모바일 인터페이스는 평균 14%, 지오센서 인터페이스는 9%의 절감 효과를 볼 수 있다. 그리고 시스템의 구현 환경 및 적용된 디자인 패턴마다 시스템의 성능이 다를 수 있음을 확인할 수 있다.

2. 모바일 인터페이스 효율성 검토

객체지향 소프트웨어 개발의 주요 목적은 재사용성과 개발 시간의 단축, 비용의 절감 및 소프트웨어 품질 향상에 있다. 본 논문은 모바일 인터페이스의 효율적인 설계로 개발시간 단축 및 유지보수 최소화, 재사용성 증가 등 많은 이점이 있다.

그림 12와 그림 13은 Quest Software사의 JProbe 툴을 이용하여 객체지향 설계유형에 따른 모바일 인터페이스의 실행시간에 따라서 메모리 성능과, 인터페이스 실행 시 예외발생에 따른 병목현상을 그래프로 나타내었다.

그림 12에서 Eden Space는 새로 생성된 모든 객체들이 위치하는 메모리 영역, Survivor Space는 Eden 영역의 객체 중 활성화된 객체가 위치하는 메모리 영역, Old Gen은 Full Garbage Collection의 대상이 되는 객체가 위치하는 메모리 영역, Available은 사용가능한 메모리 영역이다. Eden Space, Survivor Space, Old Gen은 프로그램 상에서 데이터를 저장하기 위해 동적으로 할당하여 사용하는 Heap 메모리 안에 위치한다. 시간이 갈수록 Supervisor Space가 늘어나는 것을 볼 수 있는데, Supervisor Space가 활성화된 객체들이 증가하면서 재사용 되는 객체가 증가하여 메모리의 사용 효율을 높였음을 보여준다.

그림 13은 함수 호출 시 예외 발생에 따른 병목현상 상태 그래프로 여러 개의 상자는 함수 이고 함수 사이 연결된 선은 함수 호출을 의미한다. 그리고 상자와 선의 색이 빨갛게 될수록 병목현상이 없고, 회색에 가까울수록 병목현상이 심한 것을 볼 수 있다.

프로그램 실행 시 발생하는 예외가 많을수록 병목현상이 자주일어 나는데 본 논문의 모바일 시스템 인터페이스는 프로그램 실행 시 발생하는 예외가 매우 적어서 병목현상이 일어나지 않음을 보여준다. 이것은 디자인 패턴기반으로 객체간, 객체내 관계를 명확하게 설계하여 시스템의 안정성 및 신뢰성을 높여 준다.

본 논문은 사용자와 시스템 관점에서의 분석과 객체지향 설계유형인 디자인 패턴을 이용한 설계로 소프트웨어 개발에 있어서 개발시간을 단축, 소프트웨어 재사용성을 증가, 비용을 절감 및 안정성과 신뢰성에 확신을 갖게 한다.

결 론

본 논문에서는 디자인 패턴 설계 방법을 모바일 시스템에 적용 하였다. 구체적인 클래스를 지정하지 않고 관련성을 갖는 객체들의 집합을 생성하거나 서로 독립적인 객체들의 집합을 생성할 수 있는 인터페이스를 제공하였다. 인스턴스를 이용해서 생성할 객체의 종류를 명세하고 복사해서 새로운 객체를 생성하여 재사용성을 높였다. 서브시스템에 있는 인터페이스 집합에 대해서 하나의 통합된 인터페이스를 제공하고, 다른 객체로의 접근을 통제하기 위해서 다른 객체의 대리자 또는 다른 객체로의 정보 보유자를 제공하여 객체들 간의 상호작용을 캡슐화하였다.

LBS기반 모바일 시스템 구현에 있어서 디자인 패턴을 이용한 설계 기법은 사용자와 시스템 관점에서 소프트웨어 개발에 있어서 개발시간을 단축, 유지보수를 최소화, 모듈 및 소프트웨어의 재사용성을 증가, 비용을 절감 및 안정성과 신뢰성에 확신을 갖게한다.

앞으로 모바일 환경뿐만 아니라 서버나 클라이언트 등 일반 PC환경에서도 효율적인 설계를 하여 확장된 시스템을 구축할 것이다. 또한 이 연구를 기반으로 USN이나 Embedded 시스템에 적용할 기반 구축 연구도 진행할 것이다. KAGIS

참고 문헌

김운용, 최영근. 2002. 디자인 패턴지향 소프트웨어 개발 지원도구. 정보과학회 논문지 29(8):555-564.

김종수, 김태석, 권오준. 2006. 디자인 패턴을 적용한 네트워크 게임의 DB관리 APIs 설계. 멀티미디어학회 논문지 9(1):101-108.

문영채, 이홍로. 2008. GIS기반 GPS를 이용한 농작물 작황 조사 모바일 시스템 구축. 한국지리정보학회지 11(4):1-10.

백정호, 이홍로. 2006. 객체지향 설계 유형에 의한 지오센서 인터페이스 구현. 한국지리정보학회지 9(3):193-206.

이기영, 노경택. 2002. Mobile GIS를 위한 클라이언트 인터페이스의 설계 및 구현. 한국컴퓨터정보학회 논문지 7(4):16-23.

Benedikt Schulw, Thomas Genbler, Berthold Mohr, Walter Zimmer. 1998. On the computer aided introduction of design patterns into object-oriented systems. Technology of Object-Oriented Languages TOOLS 27. Proceedings 22-25 Sept. 1998, pp.258-267.

Chengwan He, Zheng Li, Kequing He. 2008. Identification and Extraction of Design Pattern Information in Java Program. Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing, Ninth ACIS international Conference on 6-8 Aug. 2008, pp.828-834.

Jing Gang Chu, Jia Chen. 2008. Research and Application of Design Patterns on Shopping Mall Component Design. Control and Decision Conference, 2-4 July 2008, pp.697-700.

T. H. Ng, S. C. Cheung. 2005. Enhancing class commutability in the deployment of design patterns. Information and Software Technology Vol. 47, Issue 12, 1 September 2005, pp.797-804.