

## 모바일 PST를 위한 효율적 PA 기법 설계 및 구현

심재성<sup>1</sup> · 박석천<sup>2\*</sup>

### Design and Implementation of Efficient PA Technology for Mobile PST

Jae-Sung Shim<sup>1</sup> · Seok-Cheon Park<sup>2\*</sup>

<sup>1</sup>Department of IT Convergence Engineering, Gachon University, Gyeonggi-do, 461-701, Korea

<sup>2</sup>Department of Computer Engineering, Gachon University, Gyeonggi-do, 461-701, Korea

#### 요 약

PST는 제품 컨설팅을 위해 제품 카탈로그를 시스템화 한 것으로 대다수의 모바일 PST(Premium Sales Tool)는 DA(Dynamic Allocation)기법을 사용한다. 그러나 모바일 기기는 수용할 수 있는 자원이 한정되어 있어 고해상도의 콘텐츠를 연속적으로 질의 할 경우 요청한 리소스가 대기상태로 쌓여 CPU의 과부하가 발생할 수 있다.

따라서 본 논문에서는 사용자의 제스처에 따라 페이지 할당과 해제를 주기적으로 제공함으로써 부하발생에 대한 문제를 개선하여 향상된 PST 시스템을 설계 및 구현하였다. 구현한 시스템의 성능을 테스트 한 결과 기존 DA 기반 시스템 보다 제안한 PA(Pattern Allocation) 기반 시스템이 페이지 할당과 해제 빈도수에 대한 부하가 평균 약 78%의 감소율을 보였으며, CPU 점유율은 평균 약 72% 감소한 것을 확인하였다.

#### ABSTRACT

PST stands for the Premium Sales Tool for Product Consulting is customized product catalog system. In other words, the product classification categories of content for consulting and sales to customers by providing useful information for the maximization of an overall system. These problems with mobility and flexibility of updating the solution to integrate mobile with mobile devices, but high-performance high-resolutions era content data continuously, but still it is difficult to handle.

In this paper, to solve this problem, depending on the user's gesture by allowing periodic allocation and release of a problem for load generation enhanced by improving the PST system has been designed and implemented. Implementation was a result of the test the performance of the system, that PA system that has been provided by the traditional DA system show the allocation and the average 78% of the rate of decrease the load is released in the frequency of page, CPU utilization was reduced an average 72% I was confirmed.

**키워드** : 카탈로그, 모바일, DA기법, PA기법, PST

**Key word** : Catalog, Mobile, DA Techniques, PA Techniques, PST

접수일자 : 2014. 10. 14 심사완료일자 : 2014. 10. 28 게재확정일자 : 2014. 11. 14

\* **Corresponding Author** Seok-Cheon Park(E-mail:scpark@gachon.ac.kr, Tel:+82-31-750-5328)

Department of Computer Engineering, Gachon University, Gyeonggi-do, 461-701, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2014.18.12.3049>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

PST는 Premium Sales Tool의 약자로 제품 컨설팅을 위해 제품 카탈로그를 시스템화 한 것으로 주로 종이와 전자 카탈로그를 이용하고 있다. 특히 전자 카탈로그는 상품 갱신이 용이하며, 갱신 제품정보의 신속한제공과 동영상, 소리 사용이 가능하여 제품정보 질을 높일 수 있는 장점이 있으나 컴퓨터가 없으면 이용불가하다[1].

이에 최근에는 모바일 기기 보급을 통하여 인터넷 접속할 수 있으며, 이동이 자유롭고 업데이트가 가능한 다양한 모바일 기기를 통해 기존 카탈로그 개선을 위한 다양한 연구가 활발하게 진행 중이다[2, 3].

그러나 모바일 기기는 수용할 수 있는 데이터 자원이 한정되어 있어 고해상도의 콘텐츠 데이터를 연속적으로 질의 했을 경우 요청한 리소스가 대기상태로 쌓여 CPU의 과부하가 발생하는 치명적인 문제점이 있다.

따라서 본 논문은 이를 해결하기 위해 사용자의 제스처에 따라 페이지 할당과 해제를 주기적으로 관리함으로써 CPU부하 발생에 대한 치명적인 문제를 개선하기 위한 향상된 PST 시스템을 설계 및 구현하였다.

본 논문의 구성은 1장 서론에 이어 2장에서는 PST, 데이터 스트림, 앱 샌드박스를 분석하고, 3장에서 가로 세로 페이지 처리를 위한 UI와 데이터를 효율적 관리를 위한 PA를 설계하였다. 그리고 4장에서는 이를 구현 및 테스트 하고, 마지막 5장에서 결론을 기술하였다.

## II. 관련연구

### 2.1. PST(Premium Sales Tool)

PST는 제품 컨설팅을 위하여 제품 카탈로그를 시스템화한 것으로, 제품 컨설팅을 위한 콘텐츠 범주를 분류하여 고객에게 정보를 제공하기 위한 시스템이다.

제품 카탈로그는 주로 종이나 전자 카탈로그를 이용하고 있다. 이는 제품이 새로 출시될 때 마다 카탈로그의 양이 늘게 되고 매번 업데이트를 해야 한다.

또한 전자 카탈로그를 사용한다고 하더라도 노트북이나 PC를 이용해야 되기 때문에 이동성이 확보되었다고 보기도 힘들다. 이에 따라 업무의 복잡성이 발생하고 사용자가 고객에게 컨설팅하기 어려운 문제점이 발생한다.

### 2.2. 데이터 스트림 관리

데이터 스트림에서는 입력되는 데이터에 따라 실시간으로 대응함으로, 이를 저장하기 위한 스트림 큐는 그 저장 공간이 동적으로 변해야 할 필요가 있다.

일반적으로 동적인 메모리 관리를 위해 메모리 풀을 사용하는데, 메모리 풀은 생성될 때 대응량의 메모리를 할당 받고, 이 메모리를 고정된 크기의 페이지로 분리하는 역할을 수행한다.

또한 메모리 풀에서는 페이지 할당 요청에 따라서 여분의 페이지를 요청한 만큼 제공하거나 페이지 해제를 요청하면, 페이지를 여분 페이지로 돌려 관리한다.

즉, 데이터 스트림 관리 시스템에서는 메모리 풀을 사용하여 스트림 큐를 동적으로 관리하는 것을 말하며, DA기법이란 이러한 메모리 풀로부터 새로운 데이터 페이지를 할당 받고 사용하지 않으면 즉시 해제하는 작업이 즉각적으로 이루어지는 기법이다[4].

### 2.3. App Sandbox Storage

iOS 플랫폼은 앱이 메모리 사용 및 데이터 저장소의 측면에서 서로에게 간섭받지 않도록 관리한다. 따라서 각각의 앱은 데이터를 저장할 수 있는 파일 시스템에 제한을 받는다. 또한 iOS는 앱들이 각 앱의 Documents와 Temporary 디렉터리에만 읽고 쓰기를 허용한다. 이를 앱 샌드박스라고 부른다[5]. 그림 1은 앱 샌드박스의 레이아웃이다.

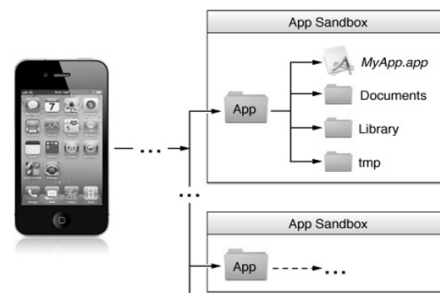


그림 1. 앱 샌드박스 레이아웃  
Fig. 1 App Sandbox Layout

앱 샌드박스는 악성 코드가 앱에 침입했을 경우 앱 샌드박스에서 도난, 손상, 삭제된 사용자 데이터에 대한 보안을 제공한다. 또한 앱 또는 프레임워크에서 코딩의 에러에 대한 손상을 최소화하는 역할도 한다.

### III. Mobile PST 시스템의 설계

#### 3.1. 설계 시스템의 개요

클라이언트에서는 서버에 요청한 데이터를 앱 고유 저장소 샌드박스에 저장하고, 콘텐츠 정보를 가지고 있는 XML 데이터를 파싱하여 iPad에 적용하여 사용자에게 직관적 UI를 제공한다. 그림 2는 사용자에게 다양한 콘텐츠 제공을 위한 대용량 데이터 관리구조이다.

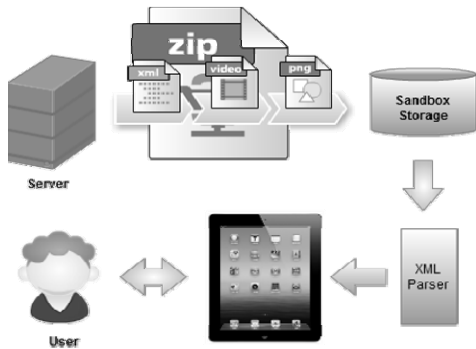


그림 2. PST 시스템 개념도  
Fig. 2 PST System Concept

#### 3.2. Mobile PST 설계 시 고려사항

Mobile 기반 앱은 웹 서버에서 콘텐츠를 전송 받아 처리하고, 고해상도 사진, 동영상 등의 대용량 데이터를 다루기 위한 요구사항을 고려해야 한다[6, 7].

메모리 관리를 위해 이미지의 포지션에 따라 주기적으로 사용하는 이미지와 간헐적으로 사용하는 이미지를 분류하여 각각 처리하였다. 또한 유연성을 위해 수정, 추가 사항에 대한 요청에 유연하게 대처할 수 있도록 콘텐츠를 서버에서 별도로 관리하고, 콘텐츠 정보를 담고 있는 XML을 이용하여 자료의 재사용성을 높였다. 그리고 간편한 UI를 위해 최소한의 컨트롤과 콘텐츠를 각 기능에 따라 분류하였다. 각 기능에 따른 내용은 가로부로, 세부적인 내용은 세로부로 분류하여 상하 좌우 페이지징 가능한 UI로 설계 및 구현하였다.

#### 3.3. Mobile PST 동작절차

본 논문에서 제안하는 시스템의 동작절차는 그림 3과 같다. 썸네일 화면①에서 'B' 사진을 터치 했을 때 선택된 'B' 화면이 확대되고, 그 화면은 ②와 같다.

②화면을 우측에서 좌측으로 플리킹(Flicking : 손가

락으로 튀기는 동작)하면 ③과 ②는 화면에서 사라지고 우측에 있던 'C'가 화면에 출력된다. ④는 ③을 아래에서 위로 플리킹하면 출력되는 화면이며, 'C'를 밀어내고 'F'가 화면에 출력되는 것을 확인 할 수 있다.

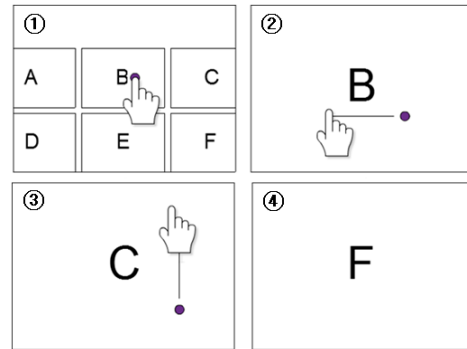


그림 3. 모바일 PST 동작 절차  
Fig. 3 Mobile PST Operating Procedures

#### 3.4. Image loading 설계

본 논문에서 제안하는 시스템 UI는 사용자 편의에 따라 최소입력으로 화면에 이미지를 출력하고, 연속적 질의에 따라 샌드박스에 저장된 콘텐츠를 사용자에게 출력하여야 한다. DA기법은 요청에 따라 페이지 할당과 즉시 해제를 반복하여 데이터 손실, 명멸이 생긴다. 이러한 문제점을 개선하여 데이터 가로세로에 따라 연관성에 따라 분류하고, 사용자 제스처 패턴을 분석하여 콘텐츠 할당과 해제를 재설계하였다.

#### 3.5. 뷰의 라이프 사이클

뷰의 라이프사이클은 제스처 이벤트가 요청시 큐 영역의 객체 개수를 확인하여 3이하인 경우 뷰를 생성해 큐에 저장한다. 현재 생성된 뷰가 사용자에게 의해 요청된 페이지와 다음 출력될 페이지를 구분하고, 조건에 따라 삭제될 뷰 번호를 반환한다. 삭제된 뷰는 큐의 객체와 비교하여 이 또한 제거 한다. 기존 시스템에서는 한 화면에서 가로, 세로 중 하나의 페이지만 수행되어 사용자가 전·후 두 상황만 고려하면 되지만 콘텐츠를 연속적으로 질의하게 되면 응답 시간이 지연되고, 부하가 심하면 강제종료 될 수 있다. 본 논문에서는 이러한 문제점 해결을 위해 Image loading 설계를 기반으로 뷰 라이프사이클 동작 알고리즘을 제안하였다. 그림 4는 뷰 라이프사이클 동작 알고리즘이다.

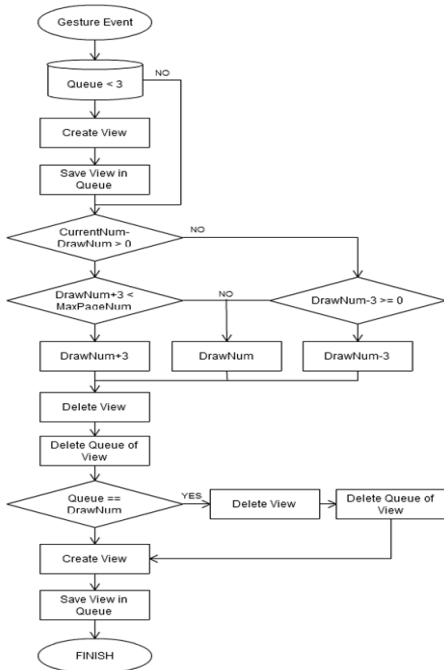


그림 4. 뷰 라이프사이클 동작 알고리즘  
Fig. 4 View Life-Cycle Operation Algorithm

### 3.6. Data Flow Diagram

제안 시스템은 앱의 유연성을 고려하여 로컬에서 콘텐츠 데이터를 관리하는 것이 아니라 서버에서 별도로 관리하여, 최신 데이터를 간편하게 받아 볼 수 있다. 그림 4는 제안하는 Mobile PST의 DFD이다.

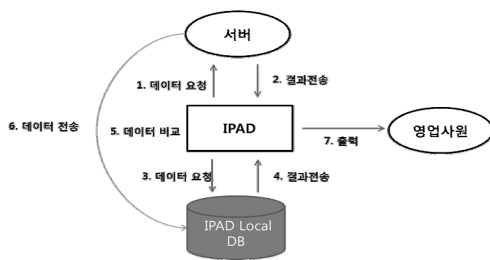


그림 5. 모바일 PST 데이터 플로우 다이어그램  
Fig. 5 Mobile PST Data Flow Diagram

iPad는 최신 정보에 대한 기록 정보를 서버에 요청하고, 서버로부터 받은 정보와 로컬 DB의 정보와 비교하여, 전송 받은 데이터를 서버에 요청한다. 클라이언트는 요청한 압축파일을 받아 압축을 해제한다.

## IV. Mobile PST 시스템 구현 및 테스트

### 4.1. 콘텐츠 뷰의 XML 데이터 맵핑

본 논문에서 구현한 Mobile PST는 Mac OS 운영체제로 XCode를, 개발언어는 Objective-C를 사용하였다.

콘텐츠 뷰는 이미지, 동영상 콘텐츠 등을 제공하는 가장 화려하고 활동적인 뷰로, 각 콘텐츠 뷰는 XML에 정의한 컨트롤과 타겟 이벤트정보와 맵핑하여 보이게 된다. 그림 6은 각 뷰의 이미지 네임, 크기 등 콘텐츠의 모든 정보를 XML 데이터로 정의하고 있다.

```

<PSTRoot>
<Tab Number="1" PageCount="2">
  <Page>
    <InitAttribute>
      <PageNum>1</PageNum>
      <Type>Image</Type>
    </InitAttribute>
    <BackgroundImage>BackgroundImage.png</BackgroundImage>
    </InitAttribute>
    <BodyAttribute Type="MoviePlay">
      <Content>
        <Target>
          <ContentTag>1</ContentTag>
          <ContentPath>MoviePlay.mp4</ContentPath>
          <MovieType>0</MovieType>
        </Target>
        <Control>
          <ControlTag>100</ControlTag>
          <ControlPath>ControllImage.png</ControlPath>
          <Frame>
            <Origin_x>0</Origin_x>
            <Origin_y>199</Origin_y>
            <Size_width>576</Size_width>
            <Size_height>389</Size_height>
          </Frame>
        </Control>
      </Content>
    </BodyAttribute>
  </Page>
</Page>
<...>
</Page>
</Tab>
<Tab Number="1" PageCount="5">
  <...>
</Tab>
</PSTRoot>
    
```

그림 6. Mobile PST의 XML 데이터 정의  
Fig. 6 The XML Data Definition of Mobile PST

그림 7은 그림 6의 XML에서 얻어온 데이터를 파싱하여 매핑코드를 구현하였다.

XML로 뷰의 콘텐츠 관련 정보를 정의하고, 정보를 바탕으로 이벤트를 실행하기 때문에 자료 재사용과 업데이트에 대한 유연성을 확보할 수 있다.

```
(void)setContentsWithTab:(NSInteger)tab withPage:(NSInteger)page
withData:(NSDictionary*)dict {
    // Initialization code.
    tapNumber = tab;
    pageNumber = page;
    NSArray *keys = [dict allKeys];
    for(NSString *key in keys)
    {
        if([key isEqualToString:@"InitAttribute"])
        {
            [self makeBackgroundImageView:[dict objectForKey:key]];
        }
        for (NSDictionary *dictNew in imgelInfo)
        {
            NSDictionary *target = [dictNew objectForKey:@"Target"];
            NSDictionary *control = [dictNew objectForKey:@"Control"];
            [self makeChangelmagePath:target];
            [self makeButton:control withType:key];
        }
    }
    else if([key isEqualToString:@"MoviePlay"])
    {
        NSDictionary *imgelInfo = [dict objectForKey:key];
        for (NSDictionary *dictNew in imgelInfo)
        {
            NSDictionary *control = [dictNew objectForKey:@"Control"];
            NSDictionary *target = [dictNew objectForKey:@"Target"];
            [self makeMoviePath:target];
            [self makeButton:control withType:key];
        }
    }
}
}}}
```

그림 7. XML 맵핑 절차  
Fig. 7 XML Mapping Procedure

4.2. 성능평가

본 논문에서 제안하는 시스템이 UI의 메모리관리에 적합함을 증명하기 위해 기존 일반페이지 할당 기법으로 스트림 큐 공간을 초과할 때 마다 메모리 풀로부터 새로운 데이터 페이지를 할당 받고 사용하지 않는 페이지를 즉시 해제하는 방법(DA)과 제안기법(PA)의 페이지 할당 및 해제 빈도수와 질의 별 CPU 점유율 비교 평가를 통해 병목현상이 감소되는 것을 확인한다. 표 1은 제안하는 시스템의 단위테스트 절차이다.

표 1. Mobile PST 단위테스트 절차  
Table. 1 Mobile PST Modules Test Process

테스트 시나리오	입력데이터	기대결과
낮은 해상도 이미지부터 한 장씩 상하좌우 페이지 하여 할당-해제 빈도수와 CPU 점유율 확인	해상도 A 230x173	한 장씩 페이지 할 때 할당-해제가 감소되는지 확인
	B 685x514 C 2014x768	한 장씩 페이지 할 때 CPU 점유율이 감소하는지 확인

구현한 시스템은 연속질의를 100으로 제한하고, 입

력데이터는 콘텐츠의 해상도를 변경 하면서 테스트 했으며, DA와 PA의 할당 및 해제 빈도수와 CPU점유율을 비교하였으며 결과는 표 2와 같다.

표 2. DA와 PA의 실행 결과  
Table. 2 The Results of DA and PA

구분	할당-해제 빈도수			CPU 점유율		
	A	B	C	A	B	C
DA	61.1	61.8	60.9	3.1	15.2	31.7
PA	13.4	14.1	12.7	1.2	4.2	5.1
결과(%)	78.07	77.18	79.15	61.29	72.37	83.91

연속질의를 통한 성능평가 테스트 결과, 할당과 해제 빈도수는 해상도에 관계없이 기존 DA시스템은 평균수치인 61.27보다 제안한 PA시스템 13.40으로 약 78.13%의 감소율을 보였다. 또한 CPU 점유율도 향상된 결과를 보였으며, 해상도에 따라 다른 성능 비를 보였다. 각 콘텐츠별 CPU점유율은 해상도가 제일 낮은 ‘콘텐츠A’가 61.29%의 감소율을 보였으며, ‘콘텐츠B’는 72.37% 감소율을, 고해상도인 ‘콘텐츠C’는 83.91%의 높은 감소율을 보였다. 따라서 본 논문의 Mobile PST는 고해상도 콘텐츠에 제안하는 PA기법을 적용하여, 할당과 해제의 빈도수와 CPU점유율을 감소시켜 병목현상이 사라지고 부드러운 플리킹과 끊김 없는 페이지 실행됨을 확인 하였다.

V. 결론

기존 대표적인 모바일 PST의 DA 기법은 메모리 풀로부터 새로운 데이터 페이지를 계속적으로 할당 받고 페이지를 사용하지 않으면 즉시 해제하는 방법이다.

그러나 DA 기법을 이용한 시스템은 잦은 할당과 해제로 부하가 발생하여 연속질의가 쌓이면서 데이터를 손실하거나 플리킹이 느려지고, 페이지가 끊기는 현상이 발생하는 문제점이 있다. 이러한 문제점은 이동성과 업데이트의 유연성을 가진 모바일과의 통합으로 해결이 되었으나 모바일 고해상도 콘텐츠 데이터를 연속적으로 처리하는데 있어 어려움이 따른다.

따라서 본 논문에서는 Mobile PST의 UI는 사용자의

편의를 고려하여 설계하였고, 복잡한 기능은 축소하고 제품정보의 질을 향상시키기 위해 버튼이나 컨트롤들은 배제하였으며, 많은 정보를 일관성 있게 분류하여 적절한 레이블을 부여하였다.

또한 작동 방식을 쉽게 배우고 이해할 수 있도록 많은 동작은 제거하고, 각 페이지마다 사진 뿐 아니라 동영상, 애니메이션 등을 삽입하여 사용자의 흥미를 가질 수 있게 하였다. 이와 같은 모든 상황을 고려하여 상하 또는 좌우 플리킹 UI가 가능하도록 설계하였다.

실제한 시스템을 구현하기 위해 Mac OS X Lion 운영체제를 이용하였고, 구현 언어로는 Objective-C를 사용하였으며, 시스템의 구성은 콘텐츠 업데이트의 유연성을 위해 서버에서 콘텐츠를 관리하고 iPad는 최신 데이터를 앱 샌드박스에 요청하여 사용자에게 출력한다.

실험 데이터는 해상도를 변경하면서 테스트 하였다. 고해상도는 한 개의 콘텐츠 당 1MB 이상의 PNG 파일을 사용했으며, 사용자의 제스처에 따라 콘텐츠 데이터를 연속해서 질의하였고, 조건 패턴 할당과 해제 빈도수를 측정하였다. 그 결과 할당과 해제 빈도수는 해상도에 관계없이 기존 DA 시스템 보다 제안한 PA 시스템이 페이지 할당과 해제 빈도수에 대한 부하가 평균 78.13%의 감소율을 보였으며, CPU 점유율도 평균 72.52% 감소한 것을 확인하였다.

본 연구는 Mobile기반 가로, 세로 플리킹이 가능한 페이지처리에 따른 고해상도 콘텐츠 데이터 관리로 카탈로그뿐만 아니라 대용량 콘텐츠 관련한 모든 앱에 적용할 수 있을 것으로 사료된다.

## REFERENCES

- [1] Ig-hoon Lee, Jonghoon Chun, "System and Utilization for E-Catalog Classifier", in *Journal of KIISE : Computing Practices and Letters*, vol. 14, no. 9, pp. 876-883, 2008.
- [2] Kim Il-Heun, "The Study for Problems and Improvements of a Electronic Catalog's Usage", M.S. dissertation, Hanyang University, 2003.
- [3] Gab-jin Lee, Sung-cheol Goh, "A Guidelines for Establishing Mobile App Management System in Military Environment - focus on military App store and verification system", *Journal of Korea Institute of Information and Communication Engineering*, vol. 17, no. 3, pp. 525-532, 2013.
- [4] Baek Sung-Ha, Lee Dong-Wook, Chung Weon-il, Bae Hae-Young, "Adaptive Memory Management Method based on Utilization Ratio to Process Continuous Query", *Journal of Korea Spatial Information Society*, vol. 11, no. 2, pp. 79-88, 2009.
- [5] Goo Bon-Min, Kim Ju-Young, Lee Tae-Rim, Shin Sang-Uk, "Collection and Analysis of the Digital Evidence for Android and iOS Smart Phones", *Journal of the Korea Institute of Information Security and Cryptology*, vol. 21, no. 1, pp. 167-175, 2011.
- [6] Choi Jun-ho, Lee Hye-jin, Chang Eun-ji, "The Conceptual Components of Simplicity in the Smartphone UX Design : A Case of iPhone", *Journal of Korean Society of Design Science*, vol. 23, no. 4, pp. 207-216, 2010.
- [7] Kim Byung-Taek, Kim Young-hun, "A Study on Emotional Interface Design Based on Each Smart-phone Application Category", *Journal of Korea Design Knowledge*, vol. 20, pp. 181-192, 2011.



**심재성(Jae-Sung Shim)**

2011 평생교육진흥원 컴퓨터공학과 공학사  
2013 가천대학교 일반대학원 전자계산학과 공학석사  
현 재 가천대학교 일반대학원 IT융합공학과 공학박사과정 중  
※관심분야 : 통신이론, 의료정보 서비스, 네트워크 시큐리티



**박석천(Seok-Cheon Park)**

1977 고려대학교 전자공학과 공학사  
1982 고려대학교 컴퓨터공학과 공학석사  
1989 고려대학교 컴퓨터공학과 공학박사  
현 재 가천대학교 컴퓨터공학과 정교수  
※관심분야 : 차세대 인터넷, 멀티미디어 통신, 모바일 통신, 네트워크 시큐리티