

## ECA 모델 기반의 개인 데이터 자동 백업 시스템 구현\*

전 인 하\*\*·이 봉 구\*\*\*·김 영 지\*\*·문 현 정\*\*·우 용 태\*\*\*\*

### *Implementation of Automatic Backup System for Personal Data based on ECA Model*

Jun, In-Ha·Lee, Bong-Goo·Kim, Young-Ji·Mun, Hyeon-Jeong·Woo, Yong-Tae

#### 〈Abstract〉

In this paper, we develop system to backup automatically personal data by real time based on ECA model. This system backups data that is changed in user PC using CDP technology based on ECA model. Proposed ECA model defines file's change such as file creation, update, remove etc. in user PC to event for synchronization between server and user PC. If event is detected about file, proposed system examines condition that is defined and backup the file to server. Backup data that is stored to server can recovery on point of time that user wants. The proposed system can be applied to data management system for business, personal data management system, high-capacity contents management system etc.

Key Words : Automatic Backup, ECA Model, CDP

## I. 서 론

최근에 사용자 중심의 Web 2.0과 UCC(User Created Contents) 환경의 도래에 따라 개인 데이터를 안전하게 관리하기 위한 웹기반의 개인용 백업솔루션이 널리 사용되고 있다. 하지만 기존 개인용 백업 솔루션은 번거로운 업로드 과정, 비용 부담 그리고 중요 데이터를 타사 서버

에 보관하는데 따른 보안상의 문제점을 가지고 있다.

2002년 IDC 보고서에 따르면 기업의 중요 데이터의 75%는 서버가 아닌 개인 PC에 보관하고 있으나 바이러스, 하드웨어 오류, 도난, 파손 등과 같은 장애로 인해 일주일마다 PC 1,000대당 1대의 디스크가 파손되어 백업 소홀로 인한 개인 데이터 손실 피해가 매년 약 180억 달러에 달하고 있다[1].

현재 개인 데이터 백업을 위해 널리 사용되는 방식은 웹 서버에 일정 공간을 빌려쓰는 ASP(Application Service Provider)서비스 방식이다. 하지만 ASP 방식은 사용자의 수동적인 업로드 방식으로 실시간으로 변경되

\* 이 논문은 2007년 지식경제부 IT우수기술지원사업으로 수행된 연구임(A1300-0701-0062)

\*\* 하이브레인넷 부설연구소

\*\*\* (주)카뮤즈 부설연구소

\*\*\*\* 창원대학교 컴퓨터공학과

는 내용을 백업하기가 쉽지 않다.

최근에는 개인 데이터의 안전한 관리를 위하여 CDP(Continuous Data Protection) 기술에 대한 연구가 진행되고 있다. 연구의 일환으로 사용자 데이터의 실시간 백업에 대한 연구 및 개발이 진행되고 있다. 현재까지의 실시간 백업은 주기적으로 데이터의 변경 상태를 체크하여 변경된 데이터 파일만 업로드하거나 데이터 전체를 백업하는 방식으로 중복 저장 관리하는 문제가 있다.

본 연구에서는 ECA(Event-Condition-Action) 모델을 기반으로 개인 데이터를 실시간으로 자동 백업하기 위한 시스템을 구현하였다. ECA 모델은 이벤트 발생시 조건이 만족되는지 판정한 후 동작을 실행하는 것으로[2], 실시간으로 발생하는 다양한 정보를 모니터링 및 분석하여 신속한 대응 방법을 제공할 수 있다[3]. 이러한 특징을 이용하여 데이터의 변경 상태를 실시간으로 감지하고 분석하였다. 본 연구에서 구현한 시스템은 기존 시스템의 수동적인 백업 방식의 개념에서 탈피하여 실시간으로 변경 데이터를 감지하여 백업하는 능동적 백업 방식의 개념을 지원한다.

## II. 관련연구

### 2.1 개인 데이터 백업시스템 기술 동향

현재 국내에서 개인 데이터 백업을 위해 널리 사용되는 방식은 웹 서버에 일정 공간을 빌려쓰는 ASP 서비스 방식이다[4, 5]. 하지만 ASP 방식은 데이터 손실에 대비한 백업 위주의 서비스로 개인적인 비용 부담, 사용자의 수동적인 업로드/다운로드 과정과 중요한 데이터를 타사 서버에 보관하는데 따른 보안상의 문제점이 있다. 최근에는 실시간 백업과 암호화를 지원하는 방식이 서비스되고 있다.

국외에서의 개인 데이터 백업 방식도 ASP 서비스 방식이 대부분이며 최근에 시만텍과 IBM에서 CDP 기술을

이용한 서비스를 제공하고 있으나, 운영체제나 사용 환경이 제한되는 문제점이 있다[6]. 그리고 CDP 기술을 응용하여 실시간 백업 기능을 일부 지원하나 데이터 손실에 대비한 ASP 방식의 백업 서비스 위주로 제공되고 있다[7-9].

### 2.2 CDP 기술

CDP 기술은 파일 생성, 수정 같은 파일의 변화에 대한 백업본을 연속으로 저장하여 블록, 파일 단위로 신속한 복구시간과 유연한 복구지점을 제공하는 기술이다. CDP 기술은 이벤트에 의해 자동으로 백업을 수행하여 관리자나 사용자의 수동적인 조작없이 편리하게 데이터를 관리할 수 있는 신기술로 발전할 전망이다[10, 11]. 국내에서 CDP 기술은 아직까지 초기 단계이며, 외국에서는 시만텍의 백업 이그젝 10d, IBM의 Tivoli CDP for Files와 같은 CDP 제품을 통해 2006년 하반기부터 활성화 단계에 있다[12].

### 2.3 ECA 모델

능동 데이터베이스에서 규칙을 명시하기 위해 사용된 모델을 ECA 모델이라고 한다[2]. ECA 모델은 능동적으로 실행되어야 하는 이벤트, 이벤트 발생시 동작을 수행할지 여부를 결정하는 조건, 조건에서 지정된 행동을 수행하는 동작으로 구성된다[3]. ECA 모델은 이벤트 발생시 조건이 만족되는지 판정한 후 동작을 실행하기 위해 즉시 판정(immediate consideration), 지연 판정(deferred consideration), 분리 판정(detached consideration)과 같은 3종류의 규칙판정 옵션으로 구분된다[2].

### III. Event-driven 방식의 자동백업시스템

본 연구에서는 ECA 모델 기반의 개인 데이터 자동 백업 시스템을 구현하였다. 본 시스템은 ECA 모델 기반의 이벤트 규칙 정의, 이벤트 규칙 처리 엔진, 백업 엔진, 복구 엔진으로 구성된다.

#### 3.1 ECA 모델 기반의 이벤트 규칙 정의

##### 3.1.1 이벤트 정의

이벤트는 PC에서 사용자가 시스템이나 파일관련 작업시 발생한다. 본 연구에서 규정한 이벤트는 크게 시스템 관점에서 자동적으로 감지할 수 있는 시스템 이벤트와 사용자가 임의적으로 지정할 수 있는 사용자 정의 이벤트로 구분된다. 파일 조작이나 운영체제 시작/종료 등과 같은 작업은 시스템 이벤트로 정의하였고, 백업 스케줄링과 같은 작업은 사용자 정의 이벤트로 정의하였다.

##### 3.1.2 조건 정의

본 연구에서는 ECA 모델의 조건 평가시 사용되는 즉시 판정, 지연 판정, 분기 판정 기법을 이용하여 이벤트 별로 처리 조건을 구체적으로 정의하였다.

##### □ 즉시 판정

즉시 판정은 이벤트가 발생하는 즉시 조건을 검사하여 해당 조건이 만족되면 자동 백업을 수행하는 기법이다. 즉시 판정은 <표 1>과 같은 이벤트와 조건을 만족하는 경우에 해당된다.

<표 1> 즉시 판정 조건 정의

이벤트	조건 정의
파일	파일 이벤트가 발생하고 사용자가 지정한 자동백업타이머(timer) 지정이 없는 경우
운영체제	운영체제 이벤트가 발생하는 경우
시간	지정한 유희 시간이 만족하는 경우

##### □ 지연 판정

지연 판정은 이벤트가 발생하고 특정 조건이 만족될 때까지 기다린 후에 조건을 검사하여 해당 조건이 만족되면 자동 백업을 수행하는 기법이다. 지연 판정은 <표 2>와 같은 이벤트와 조건을 만족하는 경우에 해당된다.

<표 2> 지연 판정 조건 정의

이벤트	조건 정의
파일 이벤트	사용자가 지정한 자동백업타이머 시간이 만족될 때까지 지연 주기적인 백업을 위해 사용자가 지정한 스케줄링 시간이 만족될 때까지 지연

##### □ 분기 판정

분기 판정은 이벤트와 상관없이 '사용자가 직접 수동적으로 백업'을 수행하고자 할 때, 조건을 검사하여 해당 조건이 참이면 자동 백업을 수행하는 기법이다.

##### 3.1.3 액션 정의

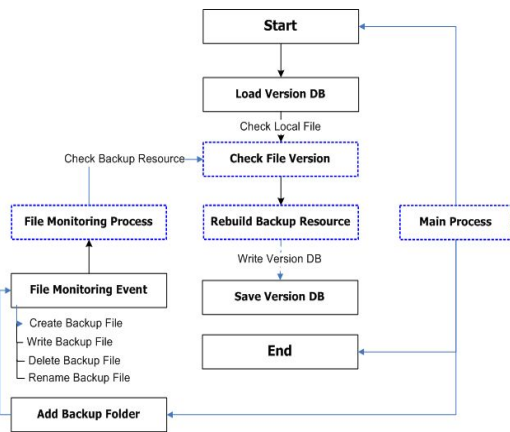
액션은 이벤트별로 정의된 조건들이 만족될 경우, 실행되는 동작을 정의하는 것이다. 본 시스템에서는 이벤트별로 정의된 조건이 만족되는 즉시, '백업파일정보생성', '파일암호화', '업로드'의 순서로 액션을 실행한다.

#### 3.2 이벤트 규칙 처리 엔진

##### 3.2.1 이벤트 규칙 처리 엔진 전체 구성도

이벤트 규칙 처리 엔진은 PC에서 시스템과 파일에 대한 조작 이벤트를 감지하고 파일에 대한 변경과 관련된 정보를 관리하기 위한 기능을 수행한다. 이벤트 규칙 처리 엔진은 크게 파일에 대한 정보를 관리하는 파일 버전 DB, 파일 및 시스템 이벤트를 모니터링하는 파일 모니터링 툴, 파일의 변경 여부를 검사하는 파일 변경 검사 모듈로 구성된다. 이벤트 규칙을 처리하는 과정은 기존의 파일 버전 DB를 로드하고 파일 모니터링 툴에 의해 이벤트가 감지되면 파일 변경 검사 모듈에서 변경된 파일

의 정보와 파일 버전 DB의 내용을 비교하여 파일 버전 DB를 재구성하는 과정으로 이루어진다. <그림 1>은 이벤트 규칙 처리 엔진에 대한 흐름도이다.



<그림 1> 이벤트 규칙 처리 엔진에 대한 흐름도

### 3.2.2 이벤트 규칙 처리 엔진의 구성 모듈

#### 1) 파일 모니터링 툴

파일 모니터링 툴은 PC 데이터의 생성, 변경, 복사 등과 같은 파일 기반 이벤트와 시스템 시작, 종료 등과 같은 시스템 이벤트를 감지하여 로그를 남기는 툴이다. 본 연구에서는 ALFA Corp. 사의 Alfa File Monitor를 구입하여 사용하였다.

#### 2) 파일 버전 DB

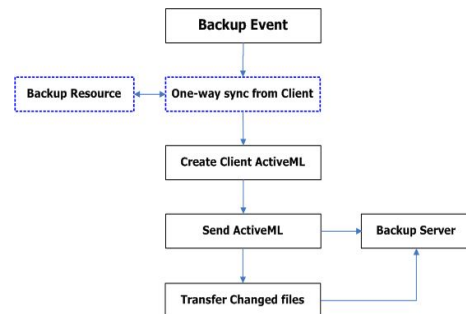
파일 버전 DB는 신규로 생성된 파일이나 변경된 파일에 대한 정보를 관리하기 위한 바이너리 파일이다. 파일 버전 DB는 파일별 ID, 파일명, 파일크기, 파일생성일, 파일 최종 수정일 등의 정보로 구성된다. 파일 버전 DB의 정보는 파일에 대해 조작 이벤트가 발생하면 파일 변경 검사 모듈에 의해 재구성된다. 그리고, 파일 버전 DB는 본 과제에서 정의한 이벤트의 조건이 만족되어 자동백업 액션이 동작할 때 서버로 파일 정보를 전송하기 위해 사용된다.

#### 3) 파일 변경 검사 모듈

파일 변경 검사 모듈은 파일 조작 이벤트 발생시 변경된 파일의 정보와 파일 버전 DB의 정보를 비교하여 새로운 파일 버전 DB를 구축하기 위한 재구성 작업을 수행한다. 먼저, 파일 변경 검사 모듈은 파일 모니터링 툴을 통해 파일 조작 이벤트에 대한 로그를 검사하여 해당 파일의 정보를 메모리 큐에 저장한다. 그리고 기존의 파일 버전 DB와 메모리 큐의 정보를 비교하여 변경 사항이 존재하면 메모리 큐의 정보를 추가할 수 있도록 파일 버전 DB를 재구성한다.

### 3.3 실시간 백업 엔진

본 연구에서는 실시간 백업을 위하여 변경된 파일 정보를 관리하기 위한 ActiveML을 정의하고, 이러한 ActiveML 인스턴스와 파일을 서버에 전송하기 위한 백업 엔진을 구현하였다. <그림 2>는 구현한 백업 엔진에 대한 흐름도이다.



<그림 2> 백업 엔진에 대한 흐름도

#### 3.3.1 ActiveML 정의

ActiveML(Active sync Markup Language)은 사용자 PC의 변경된 파일을 서버에 백업하고 복구하기 위해 정의한 XML 형식의 데이터 구조체이다. ActiveML은 사용자와 세션에 대한 인증 정보, 백업 장치 정보, 백업 방식, 백업 파일에 대한 정보 등의 내용으로 구성된다.

다음 <표 3>은 ActiveML의 엘리먼트와 속성에 대한 정의를 설명한 것이다.

<표 3> ActiveML 엘리먼트 정의

엘리먼트명	설명
ActiveML	ActiveML 스키마의 루트 엘리먼트
ActiveHeader	사용자 정보와 백업/복구 날짜, 방법을 표현하는 Complex형 엘리먼트
ActiveBody	백업/복구 파일에 대한 정보와 파일별 명령어를 표현하는 Complex형 엘리먼트
SessionID	세션 유지를 위한 사용자 세션 키 값을 표현하는 엘리먼트
UserId	사용자 ID를 표현하는 엘리먼트
MacAddress	사용자 디바이스의 Mac Address를 표현하는 엘리먼트
TargetServer	서버의 도메인 또는 IP 주소를 표현하는 엘리먼트
CurrentDate	백업/복구를 요청한 시간(년/월/일 시:분:초)을 표현하는 엘리먼트
SyncData	백업/복구를 요청한 n개의 파일에 대한 정보를 표현하는 Complex 형 엘리먼트
Final	데이터의 끝을 표현하는 엘리먼트
File	각 파일에 대한 정보를 표현하는 Complex형 엘리먼트
Command	각 파일에 대한 명령어를 표현하는 엘리먼트(신규 입력 : 100, 복구 : 200, 파일명 변경/ 파일 수정: 300, 삭제 : 400)
LocalFileId	각 파일의 유일한 해쉬 값을 표현하는 엘리먼트
ServerDir	서버에 실제로 파일이 저장된 물리적인 경로를 표현하는 엘리먼트
ServerFileName	서버에 실제로 저장된 파일 이름을 표현하는 엘리먼트 - LocalFileId + ServerModifyDate 로 구성된 값
FileName	백업/복구를 요청한 파일명을 표현하는 엘리먼트
LastModifyDate	백업/복구를 요청한 파일의 생성날짜 또는 최종 수정날짜(년/월/일 시:분:초) 를 표현하는 엘리먼트
ServerModifyDate	백업/복구 타임스탬프로 파일이 서버에 저장된 최종시간(년/월/일 시:분:초)을 표현하는 엘리먼트
FileSize	백업/복구를 요청한 파일 크기(byte단위)를 표현하는 엘리먼트

### 3.3.2 프로토콜 정의

본 연구에서는 SyncML 동기화 프로토콜[13]의 7가지 동기화 타입 중 백업을 위해 'One-way sync from client only'타입을 이용하였고 복구를 위해 'One-way sync from server only'타입을 이용하였다.

#### 1) One-way sync from client only 방식

One-way sync from client only 방식은 사용자 PC에서 자동 백업을 수행하는 경우에 사용되는 동기화 방식이다. 백업 폴더로 지정된 폴더 이하의 모든 파일의 정보를 서버로 전송한다.

#### 2) One-way sync from server only 방식

One-way sync from server only 방식은 서버에 이미 백업되어 있는 사용자 폴더 및 파일 정보를 PC로 모두 전송한다.

### 3.3.3 백업 엔진 구성 모듈

백업 엔진은 사용자 PC 파일의 변경 파일 정보를 생성하는 ActiveML 생성 모듈, 백업 및 복구 정보를 전송하는 파일 정보 전송 모듈로 구성된다.

#### 1) ActiveML 생성 모듈

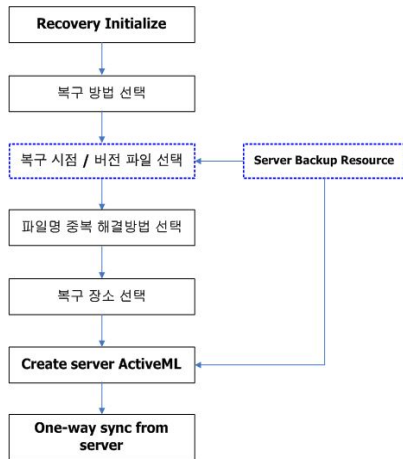
데이터 백업 엔진을 통해 PC와 서버간에 백업 정보를 교환하기 위해 신규, 변경, 삭제된 파일에 대한 ActiveML 기반의 백업 정보 구성이 필요하다. ActiveML 생성 모듈에서는 이벤트 규칙 처리 엔진을 통해 만들어진 파일 변경 정보(Backup Resource)를 이용하여 ActiveML 기반의 XML 인스턴스를 생성한다.

#### 2) 파일 정보 전송 모듈

ActiveML 전송 모듈에서는 백업 및 복구 모드에 따라 'One-way sync from client only' 또는 'One-way sync from server only' 타입으로 ActiveML 인스턴스와 변경 파일을 전송한다.

### 3.4 복구 엔진

본 연구에서 개발한 실시간 백업 시스템은 이벤트가 발생할 때마다 변경된 파일을 서버로 업로드하는 방식으로 동작한다. 따라서 서버에는 하나의 파일명에 대해 시간별로 여러 개의 백업 파일이 생길 수 있으며 이러한 시간대별 백업 파일들을 버전 파일이라 한다. 본 연구에서는 버전 파일을 대상으로 최종 백업 파일 복구, 시간대별 복구, 특정 버전 파일별 복구 기능을 수행할 수 있는 복구 엔진을 개발하였다. <그림 3>은 복구 엔진에 대한 흐름도이다.



<그림 3> 복구 엔진에 대한 흐름도

#### 3.4.1 복구 엔진 모듈

##### 1) 완전 복구 모듈

완전 복구 모듈은 사용자가 모든 데이터를 손실하거나 PC를 교체한 경우에 자동백업시스템의 서버에 저장된 사용자의 모든 파일을 다운로드하여 복구하는 모듈이다. 최초 백업 시점부터 최종적으로 백업된 시점까지의 모든 버전 파일에서 최종 버전의 파일들과 백업된 폴더 구조까지 모두 복구해 내는 방법이다.

##### 2) 불완전 복구 모듈

불완전 복구 모듈은 사용자의 임의적인 선택에 의해 특정 시점 또는 특정 버전을 기준으로 서버에 저장된 데이터를 다운로드하여 PC 데이터의 일부를 복구하기 위한 모듈이다. 불완전 복구 방법은 시점기반 불완전 복구와 버전기반 불완전 복구 방법이 있다.

##### □ 시점기반 불완전 복구

시점기반 불완전 복구 방법은 자동 백업된 날짜와 시간을 검색하여 해당 시간에 백업된 버전 파일을 모두 다운로드하여 복구하는 방법이다. 시점기반 불완전 복구는 변경 내용이 적용된 특정 시점을 알고 있는 경우에 사용할 수 있는 복구 방법이다. 예를 들어, 최종 보고서를 작성하던 중에 특정 페이지를 삭제하고 저장한 후, 계속해서 최종 보고서 작성을 진행하던 중에 수십분 전에 삭제된 특정 페이지의 내용이 필요한 경우, 시점기반 불완전 복구가 필요하다.

##### □ 버전기반 불완전 복구

버전기반 불완전 복구 방법은 자동 백업된 특정 파일의 버전 파일들 중에 필요한 파일을 선택하여 복구하거나 해당 파일의 버전 파일 모두를 복구하는 방법이다. 예를 들어, 최종 보고서 작성 중에 작성 시작일부터 현재까지 날짜별로 최종 버전 파일들을 복구할 필요가 있는 경우에 최종 보고서의 버전 파일들 중에서 백업된 날짜별 최종 버전 파일들을 선택하여 복구할 수 있다.

### 3.5 시스템 구현 화면

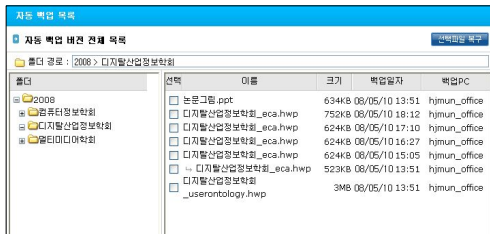
시스템은 클라이언트와 서버 시스템으로 구분된다. 클라이언트 시스템은 실시간 자동 백업이 주된 기능이다. 본 클라이언트 시스템은 윈도우 운영체제 환경에서 사용자 PC에 인스톨하여 자동 백업 대상 폴더를 지정한 후, 파일 생성, 수정, 삭제와 같은 이벤트가 발생할 때마다 변경된 파일을 자동으로 서버로 백업한다. <그림 4>는

클라이언트에서 동작하는 시스템의 메인 화면이다.



<그림 4> 클라이언트 시스템 메인 화면

서버 시스템은 완전 복구와 불완전 복구를 수행할 수 있는 복구 기능, 사용자 PC에서 백업된 정보를 브라우저를 통해 검색할 수 있는 웹 디스크 기능, 사용자별로 디스크 사용 현황 관리 기능으로 구성된다. <그림 5>는 사용자가 선택한 폴더에 대해 자동으로 백업된 결과를 보여주는 화면이다.



<그림 5> 자동 백업 결과 화면

#### IV. 결론

본 연구에서는 ECA모델을 기반으로 개인 데이터를 실시간으로 자동 백업하기 위한 시스템을 구현하였다. 본 연구에서 개발한 시스템은 ECA 모델 기반으로 CDP 기술을 이용하여 사용자 PC 에서 변경된 데이터를 실시

간으로 백업한다. 즉, 기존 시스템의 수동적인 백업 방식의 개념에서 탈피하여 실시간으로 변경 데이터를 감지하여 백업하는 능동적 백업 방식의 개념을 지원한다.

ECA 모델에서는 서버와 PC간의 동기화를 유지하기 위해 사용자 PC에서의 파일의 변경 사항을 이벤트로 정의한다. 파일에 대한 이벤트가 감지되면 제안시스템은 시스템이나 사용자에 의해 정의된 조건을 검사한 후, 변경 파일을 서버로 백업한다.서버에 저장된 데이터는 사용자가 원하는 시점에 복구할 수 있다.

본 시스템은 중소 규모의 기업, 연구소 등에서 업무용 지식 문서를 저비용-고효율의 PC 서버를 이용하여 체계적으로 관리하기 위한 업무용 데이터관리시스템 분야, Web 2.0 시대에서 UCC, 보고서, 계획서, 실험 데이터 등과 같은 개인 지식을 체계적으로 관리하기 위한 개인용 데이터관리시스템 분야, e-Learning, e-Book, 전자상거래 등과 같은 분야에서 대용량 콘텐츠를 체계적으로 관리하기 위한 서비스 분야 등과 같은 다양한 분야에서 활용할 수 있을 것이라 기대된다.

#### 참고 문헌

- [1] “스토리지 SW 시장 동향,” 한국소프트웨어진흥원, 2004. 3
- [2] R. A. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 2007. 3
- [3] 이종희, 현순주, “객체관계형 데이터베이스와 능동 규칙 관리시스템의 설계와 구현,” 한국정보처리학회(SI연구회), 2000. 6
- [4] Webhard, <http://www.webhard.co.kr/>, LG데이콤
- [5] WebBackup, <http://webbackup.paran.com/>, KTH
- [6] Backup Exec, <http://www.symantec.com>, Symantec Corporation

- [7] Carbonite, <http://www.carbonite.com/>, Carbonite, Inc.
- [8] Data Deposit Box, <http://www.datadepositbox.com>, Acpana Business Systems Inc.
- [9] Data Protector, <http://www.ironmountain.com>, Iron Mountain Inc.
- [10] 유진상, “백업 시장 새로운 트렌드 실시간 데이터 복구 솔루션 CDP,” ZDNet Korea, 2006. 5
- [11] 정지원, “Data Protection과 CDP,” 효성인포메이션 시스템, 2006. 9
- [12] 김효정, “차세대 데이터 보호의 시장을 부르는 CDP,” <http://www.zdnet.co.kr/news/enterprise/storage/0,39031208,39149666,00.htm>, ZDNet Korea, 2006. 7
- [13] 이상윤, “동기화 표준 SyncML의 표준화 동향,” ITFIND, 1031, 2002. 1



김 영 지  
Kim Young Ji

2007년~현재 하이브레인넷 책임연구원  
2004년~2007년 고신대학교 초빙교수  
2004년 창원대학교 컴퓨터공학과 (공학박사)  
1999년 창원대학교 전자계산학과(이학석사)  
1997년 창원대학교 전자계산학과(이학사)

관심분야 : 온톨로지마이닝, 추천모델, e-Learning  
E-mail : yjikim@hibrain.net



문 현 정  
Mun Hyeon Jeong

2007년~현재 하이브레인넷 책임연구원  
2004년~2007년 창원대학교 연구교수  
2003년 창원대학교 컴퓨터공학과 (공학박사)  
1996년 창원대학교 전자계산학과 (이학석사)  
1994년 한국방송대학교 전자계산학과(이학사)

관심분야 : KDD, 온톨로지마이닝, 시맨틱웹  
E-mail : hjmun@hibrain.net



우 용 태  
Woo Yong Tae

1987년~현재 창원대학교 컴퓨터공학과 교수  
1995년 경북대학교 전자공학과(공학박사)  
1984년 경북대학교 전자공학과(공학석사)  
1982년 경북대학교 전자공학과(공학사)

관심분야 : 데이터마이닝, 온톨로지마이닝, 시맨틱웹  
E-mail : ytwoo@sarim.changwon.ac.kr

■ 저자소개 ■



전 인 하  
Jun In Ha

2007년~현재 창원대학교 컴퓨터공학과 박사과정 재학  
2005년 고신대학교 전산수학과(이학사)  
2007년 창원대학교 컴퓨터공학과(공학석사)

관심분야 : 온톨로지모델링, 온톨로지 추론시스템, 암호화, 네트워크  
E-mail : ihjun@hibrain.net



이 봉 구  
Lee Bong Goo

~현재 (주)카뮤즈 연구소장  
1995년 창원대학교 전자계산학과(이학석사)  
1993년 창원대학교 전자계산학과(이학사)

관심분야 : 데이터 보안, 네트워크  
E-mail : bglee@kamuse.com

논문접수일 : 2008년 5월 29일, 수정일 : 2008년 6월 12일(1차)  
계재확정일 : 2008년 6월 15일