

임베디드 기기에 저장된 사용자 로그 데이터 자동 분석 프로세스

김봉준¹⁰, 김성숙², 김종철¹, 박기진³

¹ 아주대학교 일반대학원 산업공학과

² 아주대학교 교육대학원 e-Learning 전공

³ 아주대학교 산업정보시스템공학부

e-mail : {bongjun⁰, sungsook87, kbellfe, Kiejin}@ajou.ac.kr

Process of Automatic User Log Data Analysis Stored in Embedded Device

Bongjun Kim¹⁰, Sungsook Kim², Jongcheol Kim¹, Kiejin Park³

¹ Department of Industrial Engineering, Graduate School of Ajou University

² Department of e-Learning, Graduate School of Education, Ajou University

³ Division of Industrial and Information Systems Engineering, Ajou University

요 약

임베디드 기기의 사용 이력을 정확히 파악하는 방식으로는 기기 내부에 자동으로 저장된 로그 데이터를 분석하는 방법이 있으며, 저장된 로그 데이터를 분석하기 위해서는 데이터 정제 및 변환 과정이 필요한데, 이 과정에서 임베디드 기기에 저장된 방대한 양의 로그 데이터로 인해 많은 시간과 인력이 소요되고 있다. 이에 본 논문에서는 임베디드 기기의 로그 데이터를 데이터베이스로 관리하고 이로부터 사용 이력 분석 데이터 셋을 입력, 출력, 기능부분으로 나누어 추출하는 일련의 프로세스를 설계하였고 또한 이 과정이 자동화가 되도록 구현하였으며, 이를 통해 임베디드 기기의 로그 데이터를 분석할 때 시간과 인력을 절약할 수 있다는 가능성을 확인하였다.

1. 서론

최근 임베디드 시스템(Embedded Systems) 분야에서는 기존의 산업 제어 시스템은 물론, 개인의 편리성을 충족시킬 수 있는 다양한 기기가 지속적으로 개발되고 있다[1]. 임베디드 시스템이란 미리 정해진 특정한 기능을 수행하기 위하여 소프트웨어와 하드웨어를 조합한 컴퓨터 시스템을 말하며 응용프로그램에 따라 여러 기능을 수행할 수 있는 기존 컴퓨터 시스템(예: PC)과는 구별되는 ‘내장된’ 시스템의 성격을 지닌다[2]. 이러한 임베디드 기기는 자동화 기기의 제어 장치에서부터 가전 기기와 개인용 휴대 단말기에 이르기까지 그 적용 하드웨어 범위가 다양해지고 있으며, 더불어 각각의 하드웨어와 연동되는 소프트웨어도 그만큼 복잡하고 다양하게 개발되고 있는 추세이다.

하지만 개발자 측면에서 보면 이러한 첨단 기술의 발전과, 임베디드 기기의 기능적 복잡함이 임베디드 기기의 하드웨어와 소프트웨어 연동성 부족을 일으키는 원인이 되었고, 이는 사용자 측면에서도 기기 사용에

대한 불편함을 초래하는 부분이 되었다. 이와 같은 복잡한 기능을 갖는 임베디드 기기의 소프트웨어/하드웨어 연동성(예: 사용성)문제를 해결하기 위해서는 사용자가 실제로 기기를 사용하는 행태에 대한 자료가 필수적이다[3, 4, 5].

본 논문에서는 이와 같은 사용자의 사용 행태를 조사하는 한 방법인 로그(Log) 데이터 분석을 수행하였고, 많은 시간과 인력을 절약하기 위한 원시 로그 데이터 추출에서부터 기기 사용에 대한 분석 결과를 출력하기까지의 모든 로그 데이터 분석 과정을 자동화하는 프로세스를 설계 및 구현하였다.

본 논문은 총 5장으로 구성되어 있으며, 2장에서 관련 연구와 기존연구 동향에 대하여 기술하였고, 3장에서는 로그데이터 분석 프로세스 전반에 대하여 서술하였으며 4장에서는 실행 사례와 결과에 대하여 살펴보았다. 마지막으로 5장에서는 분석 프로세스를 통한 임베디드 기기의 사용성 분석의 의의 및 향후 후속 연구에 대해 언급하였다.

2. 관련 연구

오늘날 임베디드 기기들은 지속적인 기술 발전으로 인해 무수한 첨단 기능을 선보이고 있으나, 편리성이

“본 연구는 2007년 산학연 공동기술개발컨소시엄 사업의 지원을 받아 수행되었음.”

높은 만큼 기기 사용자들은 그러한 고 기능성 기기를 사용하는데 있어 많은 어려움을 호소하게 되었다. 그 이유는 인터페이스 부분에서 매우 혼란스러운 상황 즉, 기능은 다양한데 실제로는 사용자에게 필요가 없는 경우, 또는 좋은 기능이 있음에도 불구하고 사용자 입장에서 인터페이스가 설계되어 있지 않아 사용상의 어려움으로 인해 사용할 수 없는 경우 등이 계속 되고 있다. 이로 인해 임베디드 기기를 보다 쉽고 효과적으로 사용할 수 있게 하는 사용성 분석이 매우 중요한 요인으로 자리잡아 가고 있다.

기존 임베디드 기기의 사용성 분석에서는 설문지 조사, 인터뷰, 전화조사와 같은 방법을 이용하였는데, 이러한 방법들에서는 사용자가 솔직하지 않게 답할 가능성이 있고, 또한 자신의 행동패턴을 정확히 기억하기 힘든 면이 있기 때문에 객관적인 자료의 측정이 어려울 수 있다. 이러한 부분을 해결하기 위하여 다양한 분석 방법들이 개발되고 있는데, 임베디드 기기에 내장된 로그 데이터를 이용하여 사용자의 기기 접근 패턴을 분석하는 방법이 그 중 하나이다[3]. 결국 이 방법을 사용함으로써 기기의 사용 로그 데이터를 이용하여 실험실 혹은 인위적인 특수한 상황에서의 분석이 아니라 자연스러운 환경에서의 사용자 행동 패턴을 객관적으로 파악하여 분석할 수 있게 된다.

[3]에서는 디지털 TV의 사용성 향상을 위한 로그 데이터 분석으로써, 디지털 TV에 자동 저장된 사용 로그 데이터로부터 항목별 사용 빈도를 도출한 후, 통계 분석 소프트웨어를 활용하여 메뉴 항목간 사용 패턴을 파악하기 위한 연관성 분석을 실시하였다. 하지만 위 논문에서는 로그 데이터가 임베디드 기기에서부터 추출된 후, 통계 소프트웨어로 분석되기 전까지 수행되어야 할 데이터 정제 및 변환 과정에 대한 언급이 전혀 없었다. 실제로 이러한 임베디드 기기의 로그 데이터 분석 과정에 있어서, 대용량 데이터 파일에서 의미 있는 데이터 셋을 추출하기 위해서는 많은 시간과 인력이 소요되고 있다.

최신 임베디드 기기의 로그 데이터는 그 기능의 다양성과 많은 사용량으로 인해 데이터의 크기가 매우 커지고 있는 실정이다. 참고로 유럽 지역의 DVD 플레이어를 구매한 30가구의 3개월 분량의 접속 로그 데이터의 크기는 그 레코드 수만 해도 15만개를 넘어선다[6]. 이처럼 방대한 양의 데이터를 쉽고 빠르고 안전하게 관리하기 위해서는 데이터베이스(이하 DB)의 활용이 필수적이다[7, 8].

따라서 본 연구에서는 임베디드 기기에서 추출된 로그 데이터를 DB에 저장하고, 이들로부터 임베디드 기기의 하드웨어/소프트웨어 사용 기록 분석에 필요한 모든 과정을 자동으로 처리하는 프로세스를 설계 및 구현하였다.

3. 로그 데이터 분석 프로세스

본 분석 프로세스는 임베디드 기기 안에 내장된 사용자의 로그 데이터를 이용하여 기기 사용의 빈도 분석 및 사용 패턴 분석을 자동으로 수행하는 과정이다. 먼저 데이터를 변환 및 정제하는 준비 과정으로써 하나의 대용량 사용자 원시 로그 데이터를 임베디드 기기에서 추출하여 DB에 저장한다. 이렇게 저장된 DB 테이블은 SQL 프로그램을 통하여 처리되고 다양한 분석 결과 데이터 셋이 추출된다. 특히 본 프로세스는 다양한 임베디드 기기의 분석을 위한 일반화를 위하여 결과 데이터 셋 도출 시 입력부분, 출력부분, 기능부분으로 나누어 각각의 사용 이력에 대한 빈도 분석이 자동으로 출력되도록 수행하였다. 원시 로그 데이터의 추출에서부터 결과 데이터 셋이 나오기까지의 전체적인 프로세스는 그림 1과 같다.

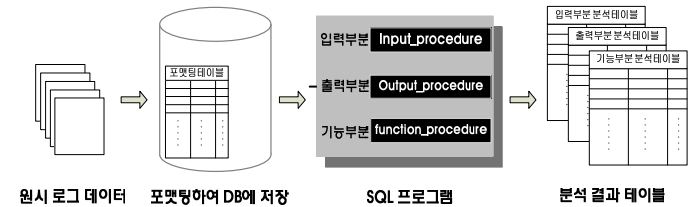


그림 1. 로그 분석 프로세스

3.1 원시 로그 데이터

로그 데이터 파일이란 임베디드 시스템이 장착된 기기의 특정 작업에 대한 사용 이력이 기록된 파일을 뜻하고, 사용자의 노력 없이도 자동으로 생성된다(예: 비행기 블랙박스). 즉, 시스템 입장에서 보면 사용자가 입력한 행동에 대한 즉각적인 반응으로써 각각의 작업 수행에서 사용된 중앙처리장치(CPU)의 작동시간, 입출력 장치의 사용시간, 수행시킨 명령어 그리고 작업의 시작과 종료 시간 등 컴퓨터 시스템의 운용에 대한 모든 기록을 의미한다.

일반적으로 로그 파일은 그 정보의 양이 많으나 실질적으로 분석의 대상이 되는 정보를 가지고 있는 레코드 수는 매우 적기 때문에, 분석목적에 따라 로그 파일 중에서 원하는 항목을 선택하여 분석해야 한다.

3.2 포맷 변경 후 DB에 저장

먼저 분석 작업의 자동화와 효율 향상을 위하여 임베디드 기기에서 추출된 로그 데이터 자료를 DB 서버로 불러오기(Import), 분석을 위한 DB 테이블로

포맷 변경하여 저장하는 작업을 한다. 이는 다양한 원시 데이터들을 본 분석 프로세스의 DB 포맷에 맞추는 작업이며 분석 프로세스의 자동화와 일반화에 반드시 필요한 과정이다.

3.3 SQL 프로그램

로그 데이터 분석을 위해 추출해야 할 데이터 셋을 입력부분, 출력부분, 기능부분으로 나누어 정의하였고, 이 3가지의 데이터 셋을 얻기 위한 DB 질의언어 (SQL: Structured Query Language)를 작성하였다.

```

CREATE PROCEDURE 'LogAnalysis'. 'Input_Procedure'
:
DECLARE Temp_crs cursor FOR SELECT temp_input,
temp_input_count from temp_input_table

SET @num=3;
Add_loop:LOOP
CALL temp_input_table_procedure();
SET @str = concat('alter table input_table add
month_',@num,'int');

PREPARE add_stmt from @str;

IF @num=5 then leave add_loop;
END IF;

EXECUTE add_stmt;
DEALLOCATE PREPARE add_stmt;
SET @num=@num+1;
:

SET l_sql = concat('UPDATE', intable, 'SET',
in_column2, '=' ,in_new_value, 'WHERE' ,in_where);

SET @sql=l_sql';
PREPARE stmt from @sql;
EXECUTE stmt;
:

END

CREATE PROCEDURE 'LOGAnalysis'. 'Temp_Input_
Table_Procedure'( 1 int, mon int, days int)
:

CREATE TABLE temp_input_table AS
SELECT Input as temp_input count(Input) as temp_input_count
FROM Logtable
WHERE month(dates)=mon AND dayofmonth(dates) = days
GROUP BY Input;

END
    
```

그림 2. 입력부분 SQL 프로그램 중요 구문

그림 2는 본 연구에서 작성된 SQL 프로그램의 한 예로 입력부분 데이터 셋을 얻기 위한 SQL 프로그램 코드의 주요 부분이다. 결과 테이블을 생성하는 프로시저를 실행하기 앞서, 먼저 첫째 컬럼이 임베디드 기기의 각 입력 항목들로 채워진 Result_Input 테이블을 생성한다. Temp_Input_Table_Procedure는 분석하고자 하는 로그 데이터를 요구되는 월, 일에 따라 정리하여 임시 테이블인 Temp_Input_Table을 생성하는 프로시저이며 메인 프로시저에서 호출되어 쓰인다. 메인 프로시저인 Input_Procedure에서는 CURSOR와 LOOP 명령어를 이용하여 Result_Input 테이블에 파악하고자 하는 월의 컬럼들을 자동적으로 생성한 후, Result_Input 테이블의 첫째 컬럼에 있는 기기 입력 항목들과 Temp_Input_Table에 정리된 빈도 값들을 비교하여 각 입력 항목에 따른 월별 사용 빈도수를 Result_Input 테이블의 새로운 월 컬럼에 자동적으로 채우게 된다.

3.4 분석 결과 테이블

입력, 출력, 기능부분에 대한 질의 완료된 DB 테이블을 얻은 후, 추후에 다른 분석 도구(예: 통계 소프트웨어)에 범용으로 적용될 수 있도록 일반적인 데이터 셋 파일(예: Excel, 텍스트)로 변환한다. 입력, 출력부분 작업 시에는 각 부분에 따라 빈도 분석이 되어야 하는 모든 값들을 SQL 프로그램으로 분석하기 전에 미리 파악하여, 결과 테이블에 빈도 분석 대상 항목을 미리 적어 놓음으로써 사용자가 일정 기간 동안 전혀 쓰지 않은, 그래서 로그 데이터에 기록되지 않은 값들까지도 결과 테이블에 표현할 수 있도록 한다(그림 3 참조).

| | |
|-------|---|
| 기기-기기 | <ul style="list-style-type: none"> - 핸드폰: 기본 키, 특수 키(카메라, MP3, DMB) - 일반적인 전자기기: 리모콘 키, 본체에 있는 키 - 셋톱박스: 리모콘 키, front 키 - 에어컨: 리모콘 키, 본체에 있는 키 - PDA: 터치펜, 터치스크린, 특수 키 등 |
| 생체-생체 | <ul style="list-style-type: none"> - 핸드폰: 여러 메뉴화면과 서브화면들, 기기에서 제공되는 다양한 음성메세지 - 일반적인 전자기기: 액정화면, 경고음, 음성메세지 - 셋톱박스: 메뉴화면, 서브화면 들 - 에어컨: 액정화면 - PDA: 액정화면, 경고음, 음성메세지 등 - 컴퓨터: 모니터화면, 경고음, 음성메세지 등 |
| 기기-기기 | <ul style="list-style-type: none"> - 기기 안에서 일어나는 소프트웨어 적인 변화 - 일반 기능의 변화 - 기기 사용 동작의 변화 - 사용자의 입력 행위에 대응하여 기기 내부에서 일어나는 동작 |

그림 3. 각종 임베디드 기기의 부분별 내용

기능부분 작업에 있어서는 임베디드 기기 내의 모든 이벤트들의 내용을 파악해야 하기 때문에 그 결과 테이블이 로그 데이터의 어느 한 부분만 결과 데이터로 추출되는 것이 아니라 로그 데이터 전체 즉, 사용자가 행한 모든 행위에 대한 결과가 모두 포함되어야 한다. 예를 들면 MP3 플레이어에서 사용자의 입력에 의해 발생한 재생 이벤트 혹은 기기 자체에서 발생한 자동 꺼짐과 같은 이벤트에 대한 빈도를 모두 포함한다. 기능부분 분석에서는 입력부분과 출력부분 분석에서 다루지 않은 이벤트에 대한 분석이 가능한데 이는 로그 데이터 간의 연관성 분석을 보다 세밀히, 기능적인 측면에서 수행 가능하게 하고 임베디드 기기의 하드웨어에 따른 소프트웨어 개발 시 메뉴 및 기능 선정, 배치 문제에 많은 도움을 줄 수 있다. 그림 3에서 각종 임베디드 기기의 입력, 출력, 기능부분에 대한 내용을 몇 가지 예로써 파악할 수 있다.

4. 분석 수행 사례

본 논문에서 제안한 로그 데이터 분석 프로세스를 적용하여 DVD 플레이어의 사용자 로그 데이터를 분석하였다. 원시 로그 데이터를 입력, 출력, 기능 부분으로 나누어 데이터 파일 세 개를 생성했으며 각 부분에 대해 전체 빈도수뿐만 아니라, 이를 각 월마다 나누어 추가 표현하였다.

4.1 대상 로그 데이터

본 연구에서 사용한 로그 데이터의 구조는 이벤트 발생 ID, 이벤트 발생 시간, 입력 종류, 기능 카테고리, 기능 이동, 세부 내용에 대한 6개의 컬럼으로 이루어져 있다(표 1 참조). Event ID 컬럼에는 각 이벤트의 고유 번호가 들어가며 하나의 이벤트가 발생할 때 마다 순차적으로 1씩 올라간다. Event Time 컬럼에는 각 이벤트의 발생 시간이 년 단위부터 초 단위까지 표현되어 있다. Input 컬럼은 사용자의 입력 정보를 나타내고, Functional Category 컬럼은 발생 이벤트를 그 기능에 따라 크게 카테고리화한 것이고 Functional Movement 컬럼은 기능의 이동 정보를 나타낸다. 마지막 컬럼인 Details는 각 이벤트의 세부적인 내용을 나타낸다. 표 1의 Event ID 1을 예로 들면, 이 로그 파일을 TV의 로그 파일이라고 볼 때 Event ID 1 이벤트의 의미는 2007년 3월 27일 15시20분40초에 사용자가 리모컨으로 TV 채널을 하나 위로 올린 것을 의미한다. 이 이벤트는 기능 카테고리 측면에서는 TUNER에 속하고 CH_UP이라는 기능 동작을 수행한 것이다. 또한 Details 컬럼에는 현재 채널이 2번이라는 자세한 정보가 나타나있다.

표 1. 임베디드 기기의 사용자 로그 데이터 테이블

| Event ID | Event Time | Input | Functional Category | Functional Movement | Details |
|----------|---------------------|------------|---------------------|---------------------|---------|
| 1 | 2007-03-27:15:20:40 | KEY_MOVEUP | TUNER | CH_UP | 2 |
| 2 | 2007-03-27:15:20:43 | KEY_HOME | HOME | ENTER | |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

4.2 분석수행결과

입력부분에서는 로그 데이터로부터 사용자의 임베디드 기기에 대한 직접적인 입력(예: 리모컨 버튼)이 몇 번 수행되었는지를 파악할 수 있다. 예를 들어 TV 로그 데이터를 분석한다면 입력부분 작업을 통해 사용자의 TV 리모컨 버튼(그림 1의 Input 컬럼의 값)의 사용 빈도를 파악할 수 있다.

DVD 플레이어에서의 입력부분인 리모컨 버튼에 대한 분석으로써 사용자가 입력 가능한 리모컨의 모든 버튼들을 파악한 후, 그에 따른 각 버튼에 대한 빈도 분석을 하였다(표 2 참조).

표 2를 보면 사용자가 DVD 플레이어 리모컨의 [POWER] 버튼을 세 달 동안 총 417번 눌렀고, 그 중 둘째 달의 사용량이 186으로 가장 많은 것을 확인할 수 있으며, 또한 숫자 [1] 버튼을 76번 눌렀다는 것을 확인할 수 있다.

표 2. 입력부분 결과 데이터 셋

| 입력 값 | 전체 | 첫째 달 | 둘째 달 | 셋째 달 |
|------------|-----|------|------|------|
| [POWER] | 417 | 49 | 186 | 181 |
| [DVD] | 148 | 22 | 43 | 82 |
| [HDD] | 208 | 34 | 114 | 59 |
| [AV/INPUT] | 467 | 79 | 224 | 144 |
| [0] | 11 | 2 | 7 | 2 |
| [1] | 76 | 23 | 29 | 24 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

출력부분 작업은 로그 데이터의 Functional Category, Functional Movement 그리고 Details 컬럼의 값들을 활용한다. 출력부분 빈도 분석 작업을 위해서는 입력부분 작업과 마찬가지로 미리 출력 값들에 대한 전체 항목들이 파악되어야 하며, 하나의 출력 항목을 뜻하는 로그 데이터 값들의 조합 또한 정의되어야 한다.

DVD 플레이어의 출력부분 분석에 있어서는 메뉴 화면에 대한 분석을 수행하였고, 이를 통해 사용자가 어떠한 메뉴 화면에 몇 번 진입했는지를 파악하였다. 사용자가 진입할 수 있는 모든 메뉴 화면들을 파악한

후, 각 출력 화면에 대한 빈도 분석을 수행하였다(표 3 참조).

표 3. 출력부분 결과 데이터 셋

| 화면 구분 | 출력 화면 | 전체 | 첫째달 | 둘째달 | 셋째달 |
|------------------|--------------------|-----|-----|-----|-----|
| Title list (HDD) | [Search]-[Chapter] | 7 | 0 | 4 | 3 |
| | [Rename] | 15 | 0 | 6 | 9 |
| | [Delete] | 21 | 0 | 3 | 18 |
| | [Edit]-[Combine] | 173 | 1 | 53 | 90 |
| | [Edit]-[Divide] | 10 | 0 | 3 | 7 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

표 3은 DVD 플레이어의 HDD에 영화가 저장 되어 있을 때 그것들을 재생 및 수정, 삭제할 수 있는 메뉴 화면들에 대한 로그 데이터 분석 결과의 일부분이다. 저장된 영화 파일을 지우는 작업을 총 21회 수행했으며, 파일 이름을 변경하는 작업을 총 15회 수행한 것을 확인할 수 있다.

해당 기기의 로그 데이터는 각 컬럼의 특성상 Functional Category, Functional Movement 그리고 Details 컬럼이 인덱스 구조(Functional Category가 최상위, Details가 최하위)를 이루기 때문에 기능부분 빈도 분석의 결과 테이블은 세 개의 컬럼을 인덱스 형식으로 정렬시킨 형태를 띄게 된다.

표 4는 DVD 플레이어의 세부 기능의 발생에 대한 분석을 수행한 결과이며, 이를 통해 기기 내부에서 일어나는 모든 이벤트들에 대한 빈도를 파악할 수 있다. DVD 플레이어의 오디오 기능을 쓸 때 음악 CD를 감지한 이벤트가 총 22회 일어났고, 음악이 재생되는 동안 앞으로 빨리감기 기능이 총 3회 일어난 것을 확인할 수 있다.

표 4. 기능부분 결과 데이터 셋

| 기능구분 | 세부기능 | 전체 | 첫째달 | 둘째달 | 셋째달 |
|-----------|------------|----|-----|-----|-----|
| Audiolist | Detect_CD | 22 | 1 | 17 | 4 |
| | F.SCAN(x2) | 3 | 0 | 3 | 0 |
| | F.SKIP | 45 | 1 | 33 | 11 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Movielist | Play_OK | 27 | 1 | 12 | 14 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

5. 결론 및 후속연구

본 연구에서는 사용자의 이용 행태 분석을 위하여 임베디드 기기에서 추출한 원시 로그 데이터를 DB

기반 하에서 분석하는 프로세스를 설계 및 구현하였으며, 이러한 프로세스의 자동화를 통해 기존의 로그 데이터 분석에 있어서 문제점으로 제기되었던 복잡한 대용량의 자료를 처리하는데 소요되는 인력과 시간을 상당부분 줄일 수 있는 가능성을 확인하였다.

또한 수집된 방대한 로그데이터를 입력부분, 출력부분, 기능부분으로 세분화하여 결과 데이터 셋을 추출하는 프로세스는 임베디드 기기의 소프트웨어와 하드웨어를 분석하는데 있어서 복잡한 기기의 연동성 분석 패턴의 일반화를 보여주었다.

후속 연구로는 이번 자동화 프로세스를 통하여 얻어진 데이터들간의 관계를 시간 변수와 연관 지어 파악 분석하는 작업이 남아 있다. 또한 소프트웨어와 하드웨어의 연동성 일반화 모델 구축 작업의 일환으로 초기 원시 로그 데이터 수집 시 효과적인 분석을 위한 로그 데이터 포맷 일반화 연구가 뒤 따라야겠다.

참고 문헌

- [1] W. Hongxing and W. Tianimiao, "Curriculum of Embedded System for Software Colleges," *Mechatronic and Embedded Systems and Applications, Proceedings of the 2nd IEEE/ASME International Conference on*, pp. 1-5, Aug. 2006.
- [2] S. Li, Z. Xiong, and T. Li, "Distributed Cooperative Design Method and Environment for Embedded System," *Computer Supported Cooperative Work in Design - Proceedings*, Vol. 2, pp. 956-960, 2005.
- [3] 하윤, 최고운, 김효선, 안정희, "가전 제품의 사용 Log 분석을 통한 사용자 중심 인터페이스 디자인에 관한 연구," *HCI 학술대회 논문집*, pp. 179-184, Feb. 2007.
- [4] 변재형, 서종환, "휴대폰의 물리적 상용성 평가 방법 개발 및 적용," *디지털디자인학회논문지*, Vol. 13, No. 0, pp. 97-106, 2005.
- [5] 박정순, "컨조인트 분석을 이용한 정보 가전 OSD의 사용성 평가," *한국콘텐츠학회논문지*, Vol. 2, No. 2, pp. 53-63, June 2002.
- [6] U2 Systems, DVD Log Analysis, Report, July 2007.
- [7] G. Harrison, *MySQL Stored Procedure Programing*, O'Reilly, 2006.
- [8] 이상협, 김성진, 이상호, "데이터베이스 튜닝을 위한 로그 분석 도구," *정보처리학회논문지D* 제11-D권, 제5호, Oct. 2004.