MOBILE DATA RECOVERY SYSTEM (EXT4)

PROJECT PROPOSAL DOC.

Jin, Hyun-wook
Professor
System Software Lab, Konkuk Univesity

Lim, Min-woo (200910793)

Lee, Young-jun (200911412)

목차 (CONTENTS)

내용

개요 (INTRODUCTION)

개요 (Introduction)

개발 배경

최근 Linux OS 의 일반적인 사용이 확대됨과 동시에 Embedded Platform 에서의 Kernel 사용 또한 일반화 되었다. 전 세계적으로 Linux 의 보편성과 범용성을 인정받으며 OS 시장에서 꾸준한 상승세를 이어가고 있다. 또한, 전 세계인이 사용하는 스마트폰 중 Linux Kernel 기반의 OS 를 사용하는 비율이 상당히 높다. 사례로, Android Platform 은 Linux Kernel 을 기반으로 제작된 Framework 로서 Linux 에서의 File System 인 EXT3 를 주로 사용하고 있다. 이처럼 EXT4 File System 의 이용 또한 증가하는 반면 파일 복구, 즉 지워진 파일을 다시 복구하는 소프트웨어는 현저히 부족하다. 따라서 본 작품을 통해 Linux 환경 뿐만 아니라 스마트폰 내에서의 EXT4 File System 에서의 Recovery System 을 구축한다.

<그림 > Computer Forensics

또한, 최근 디지털 포렌식이 중요해지면서 많이 사용되는 스마트 폰의 보안 또한 중요시 되었다. Android 모바일 디바이스도 EXT4 파일 시스템을 사용함으로써 문자 메시지, 브라우저의 내역 등 내장 Database 인 SQL Lite 를 사용하는 경우가 많다.

휴대폰은 개인의 생활과 가장 밀접한 디지털 기기로 디지털 포렌식 수사 시에 반드시 고려해야 할 대상이 되고 있다. 스마트폰은 피처폰과는 달리 일반 PC 와 유사한 고성능의 운영체제를 사용하면서 다양한 모바일 앱을 통해 사용자에게 여러 가지 기능을 제공하는 특징이 있다.

디지털 포렌식 관점에서 스마트폰의 사용 흔적을 수집하고 분석하는 것이 중요해짐에 따라 전세계적으로 스마트폰 포렌식에 관한 연구가 활발하게 이루어지고 있다.

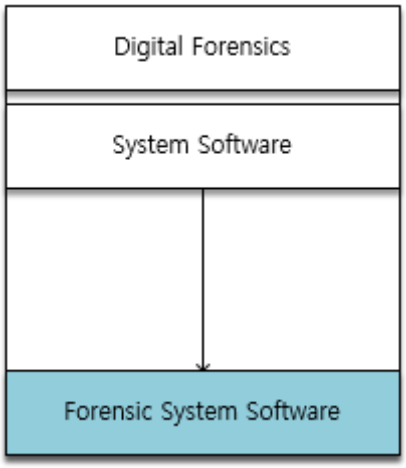
개요 (INTRODUCTION)

따라서 EXT4 File System 을 사용하는 모바일 디바이스, 즉 안드로이드 모바일 장비를 목표로 삼아 문자 메시지, 브라우저 내역 등 SQLite 를 사용하는 앱들의 삭제된 정보를 복구하는 프로젝트를 제안하고자 한다.

개발 목적

EXT4 File System 을 사용하는 Platform (Android Phone)에서의 File Recovery System 을 구축한다. 사용자가 실수로 파일을 삭제했을 때, 혹은 고의적으로 삭제했을 때 해당 Block 이 쓰이지 않았다면, 해당 내용을 다시 복구할 수 있다. 특히 문자 메시지나 브라우저 내역 등을 중점적으로 복구 하고자 한다. Linux System 을 이용하는 사용자의 편의성 향상과 File System 의 안정적인 운용을 위해 복구할 수 있는 파일들을 출력하고, 사용자가 원하는 삭제된 파일을 복구한다.

디지털 포렌식 관점에서 EXT4 File System 상의 삭제 파일 복구 현안이 대두되고 있는 만큼, 현 디지털 시대 흐름에 걸맞는 소프트웨어를 제작하고자 한다. 또한 포렌식 관점의 보안과 시스템 소프트웨어인 File System 의 결합으로 보다 나은 Forensic System Software 를 기대해본다.



<그림 > Forensics System Software

프로젝트 환경 (PROJECT ENVIRONMENT)

프로젝트 환경 (Project Environment)

대상 환경 (TARGET ENVIRONMENT)

Android Platform Mobile Phone (Odroid Series) Linux Kernel 3.0 이상

1.7GHz Exynos4412 Prime Cortex-A9 Quad-Core Processor

2GByte Ram, 8GB SD Card

Android 4.X

EXT4 File System at /data partition mounted

<그림 > Odroid-XU Lite

개발 환경 (HOST ENVIRONMENT)

Ubuntu 12.04 64bit

요구사항 (REQUIREMENT)

요구사항 (Requirement)

기술적 요구사항 (FUNCTIONAL REQUIREMENTS)

1. 사용자가 삭제한 파일 중 SQLite 를 사용하는 데이터만 추출한다.
   A. 문자 메시지 (ex> com.sec.mms)
   B. 크롬 브라우저 내역
   C. 기타 SQLite 사용하는 APP 들의 데이터
2. 추출된 데이터는 SQLite Database Browser 로 확인할 수 있다.
3. Native C 와 C++을 이용해 작성한다.
4. 필요한 경우 Shell Scripting 도 이용할 수 있다.

비기술적 요구사항 (NON-FUNCTIONAL REQUIREMENTS)

1. 해당 디바이스의 Root Privilege 를 얻는 과정은 제외한다. (필요)
2. **삭제된 후 다른 데이터가 덮어 쓰여 데이터의 일부가 남아 있는 경우 복구한다.**
3. **삭제된 영역에 하나의 레코드가 독립적인 상태로 존재하는 경우 복구한다**.
4. 삭제된 영역의 레코드가 2 개 이상 존재하는 경우 복구하지 않는다.
5. 레코드가 손상된 데이터는 복구하지 않는다.
6. 운영체제의 수정(Compile & Build) 없이 이용 가능하다.

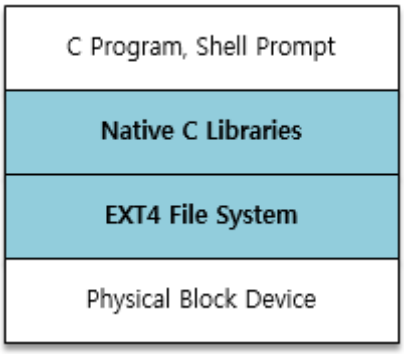
개발 방법 (DEVELOPMENT PROCESS)

개발 방법 (Development Process)

개발은 Linux Ubuntu 12.04 에서 주로 진행되며, C 언어의 Native Functions 을 이용하여 제작한다. EXT4 File System 과의 데이터 연동은 C 언어의 Native 가 관리하며, 사용자가 사용하는 프로그램은 C 언어를 이용한 프로그램으로 사용된다.

기존의 EXT3GREP 과 같은 Utility 의 사용은 일반 사용자들로 하여금 상당한 불편함을 초래하였고, 이를 보완하고자 복구 시스템 소프트웨어를 아래와 같은 방법으로 개발한다. 또한, 포렌식 관점으로 봤을 때 연구가 활발하게 진행되고 있는 이 시점에 본 소프트웨어를 개발한다.

SYSTEM LAYERED ARCHITECTURE

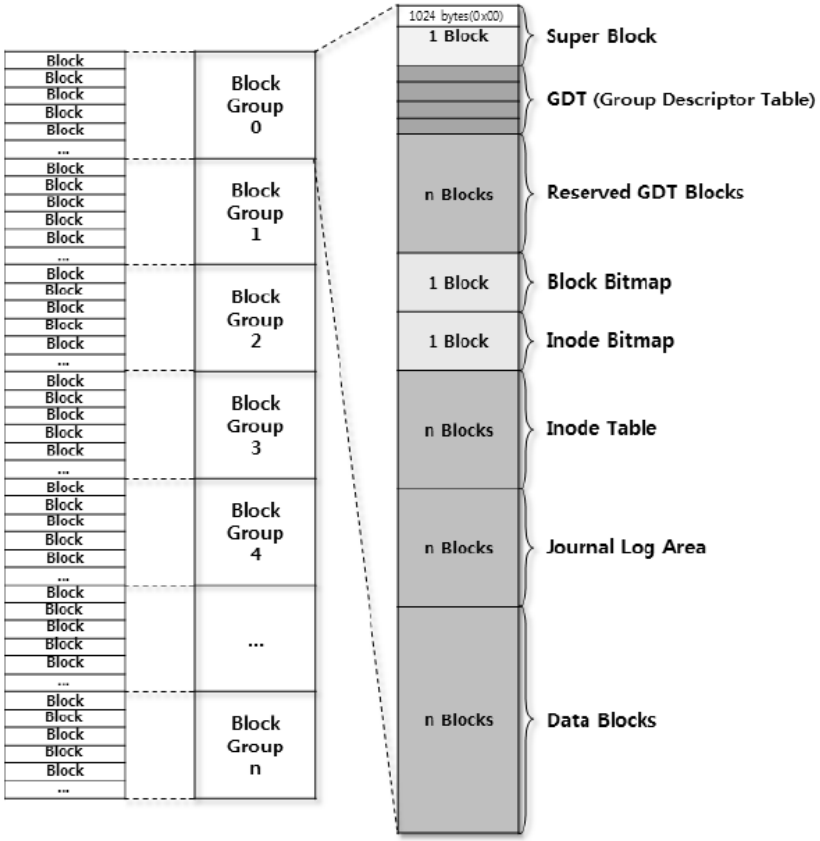


<그림 > System Layered Architecture

위 <그림 1>과 같이 EXT4 File System 의 구조를 파악하고, Linux 에서의 파일 삭제 Routine 과 방법 등을 고려해 Native C Libraries 를 이용해 파일을 복구 할 수 있도록 한다. EXT4 에 관련된 내용은 다음 절인 EXT4 File System 에 명시한다. 본 프로젝트의 가장 주된 Layer 는 Native C Library 로서, 핵심 기능은 모두 해당 레이어 에서 처리한다.

개발 방법 (DEVELOPMENT PROCESS)

EXT4 FILE SYSTEM



<그림 > EXT4 File System Architecture

EXT4 File System 은 데이터 저장의 최소 단위인 블록들로 이루어져 있으며 블록의 크기는 1KB, 2KB, 4KB 중 하나를 선택하여 사용할 수 있지만, 기본적으로 4KB 를 사용한다. <그림 2>와 같이 효율적인 관리를 위해 모든 블록들을 여러 개의 블록 그룹으로 묶어서 관리한다. 하나의 블록 그룹은 최대 32,768 개의 블록을 관리하기 때문에 블록의 크기가 4KB 일 경우 하나의 블록 그룹의 크기는 최대 128MB 이다.

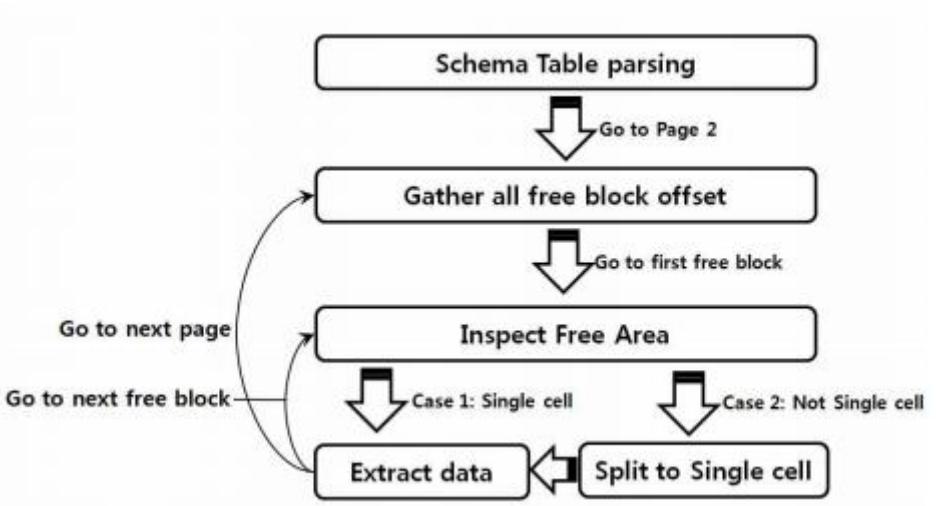
블록 그룹 내부에는 Super Block, Group Descript Table, Block Bitmap, Inode Bitmap, Inode Table, Journal Log Area 등의 메타데이터 영역이 존재한다.

개발 방법 (DEVELOPMENT PROCESS)

SCHEMA TABLE PARSING (스키마 테이블 파싱)

삭제된 레코드(문자 메시지)를 복구하기 위해서는 먼저 스키마 테이블을 통해 테이블의 필드의 개수와 각 필드의 타입을 파악해야 한다. 이후 페이지를 순차적으로 탐색하며 페이지 내의 모든 비할당 영역을 파악하여 데이터를 추출한다. 이전에 반드시 스키마 테이블을 파싱하여 타입과 개수를 파악해 둔다. 이는 후에 Leaf 셀에 존재하는 길이가 가변적인 영역을 구분하는데 이용될 수 있다.

전체적인 삭제데이터 복원 과정은 아래 그림과 같다.



<그림 > 삭제 데이터 복원 과정

Free Space(비할당 영역)을 반복적으로 검사한 후 데이터를 추출한다. 여기서 비할당 영역을 검사하는 이유는, 삭제된 레코드들은 모두 Free Space 로 인식되고, 0 이 아닌 값들이 입력되어 있는, 즉 전 상태를 유지하고 있다는 가정하에 이와 같은 프로세스로 구현한다.

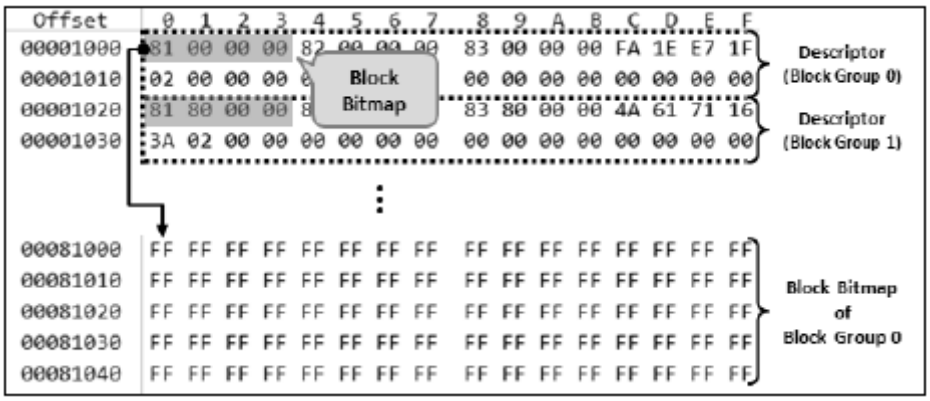
FREE SPACE EXTRACTION(GDT) : 비할당 영역 추출

안드로이드 운영체제는 내부에 여러 개의 파티션들이 존재하며 /sdcard 파티션을 제외한 나머지, /system, /data 파티션 등이 EXT4 file System 을 사용한다. 어플리케이션과 관련 있는, 즉 외부 데이터가 아닌 경우(SQLite, XML 파일 등) /data 파티션에 저장된다.

개발 방법 (DEVELOPMENT PROCESS)

안드로이드 모바일 장치 내의 어플리케이션들이 사용하는 데이터들은 한정적이다. 가장 많은 비율로 점유하고 있는 데이터의 형식은 SQLite Database File 과 어플리케이션 환경 설정 관련 파일인 XML 파일이 있다. 본 프로젝트는 SQLite Database File 을 중점적으로 다루어 문자 메시지, 혹은 크롬 브라우저의 삭제된 내역 등을 추출한다.

/data/data 디렉터리를 분석하면, 어플리케이션 별로 SQLite Database 정보를 추출 할 수 있다. 어플리케이션 초기 설치시 단편화가 일어나지 않지만, Database 의 특성상 변경이 자주 일어나므로 데이터의 조각이 생기고, 또한 페이지의 단편화가 일어난다. 따라서 이러한 단편화된 페이지를 분석하기 위해 아래와 같이 비 할당 영역을 Carving 기법으로 수집한다.



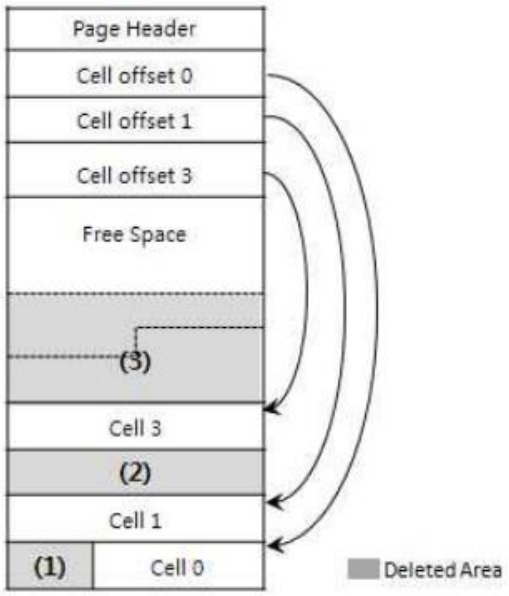
<그림 > Block Group's Block Bitmap Information

비할당 영역(Free Space)를 추출하기 위해서는 먼저 GDT 에서 총 블록 그룹의 개수와 각 블록 그룹당 Block Bitmap 의 위치를 확인해야 한다. 각 블록 그룹에 존재하는 Block Bitmap 의 위치로 차례로 이동하여 Bitmap 을 해석함으로써 비할당 블록을 파악한다.

특정 어플리케이션과 관련 있는 파일들은 같은 블록 그룹에 존재할 가능성이 높고, 파일 단편화가 다른 블록 그룹으로 발생하더라도 최대한 인근의 블록 그룹으로 파일 조각을 할당하기 때문에 단편화된 파일 조각의 재조합 과정에서의 복구율을 높이기 위해서 비할당 영역을 각 블록 그룹별로 따로 수집한다. 이처럼 비할당 영역을 추출한 후, SQLite 부분만을 따로 떼어내 진행한다.

아래 그림은 비할당 영역 중 삭제된 레코드의 그림을 나타낸다.

개발 방법 (DEVELOPMENT PROCESS)

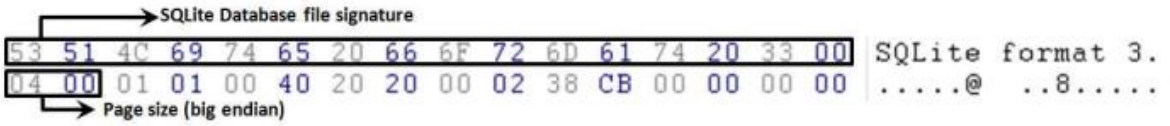


<그림 > 비할당 영역의 삭제된 레코드 영역

(1)과 (2) 만 복구 가능하고[요구사항 참고], 3 번은 2 개의 레코드가 합쳐져 있는 형태로서 복구가 불가능 하기 때문에 복구하지 않는다.

SQLITE DATABASE FILE

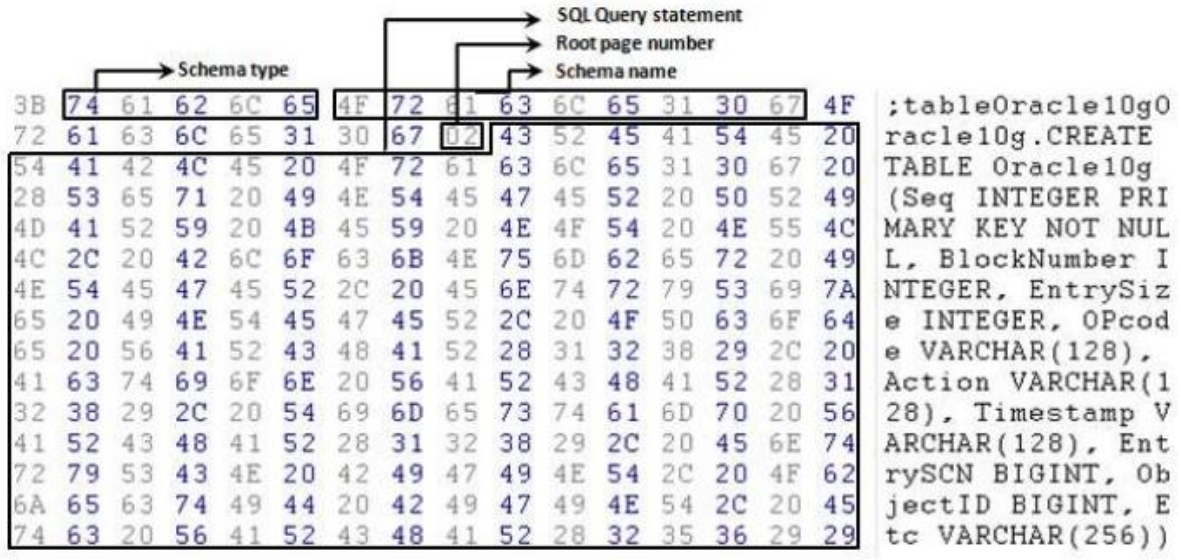
SQLite 파일의 전체 구조는 아래 그림과 같으며 파일의 시그니쳐는 0x53 51 4c 69 74 65 20 66 6f 72 6d 61 74 20 33 00 이다. 따로 확장자가 없기 대문에 시그니쳐를 통해 SQLite Database File 을 구분한다.



<그림 > SQLite Header Structure

헤더 페이지에 존재하는 스키마 테이블에는 응용프로그램에서 SQLite 파일을 구성하는 테이블, 인덱스, 트리거에 관한 정보가 들어있다. 아래 그림은 SQLite 스키마 테이블의 정보이다.

개발 방법 (DEVELOPMENT PROCESS)



<그림 > SQLite Scheme Table

이와 같이 삭제된 레코드를 복구하기 위해서 스키마 테이블 중 테이블 형식의 스키마에서 SQL Query 문을 해석한다. 테이블 이름, 필드의 타입, 이름, 필드의 개수를 추출한다.

SQLite 의 필드 타입은 Integer, Text, Blob, Numeric 네 가지가 존재하며, 4 가지 타입에 관해서만 진행한다.

USER APPLICATION (C)

Shell Script 또는 Native C 프로그램을 연동하여 Android 디바이스 내의 /data 파티션의 비할당 영역을 이미지로 추출(dd) 후 데이터 복원을 시작하여 각 어플리케이션들의 SQLite Database File 을 분석하여 삭제된 데이터 레코드를 복원한 파일을 출력해낸다. User Application 에서는 Mounting 된 /data 파티션의 경로를 입력하면 진행된다.

PREDICTED OUTPUT

ADDRESS	DATE	TEXT	ETC
123123	123123	복구 테스트 1	

개발 방법 (DEVELOPMENT PROCESS)

4567891	45697841	“동양캐피탈” 당일~	
1231233	231231231	우리금융 고객님은~	
1231239	122190	SKT- VIP 한도 초과	
123132	9123912	11 번가 특가 세일	

개발 일정 (DEVELOPMENT SCHEDULE)

개발 일정 (Development Schedule)

작업내용(임민우)	6	7	8	9	10	11
Odroid-XU Rooting 및 /data 파티션을 위한 Application 설치						
/data Free Area Image Extraction (비할당 영역 추출)						
가변 길이 영역의 길이 파악과 레코드 복원						
삭제된 레코드 CSV 파일 최종 추출부분 구현						
최종 테스트 및 디버깅						
작업내용(이영준)	**6**	**7**	**8**	**9**	**10**	**11**
SQLite Database File Analysis (Header, Scheme Relation) (SQLite 구조 분석)						
File Carving (비할당 영역의 SQLite Data 부분 추출)						
Database File 내의 삭제된 영역 파악						
삭제된 레코드 CSV 파일 최종 추출부분 구현						
최종 테스트 및 디버깅						

<그림 > 작업일정

참고문헌 (REFERENCES)

참고문헌 (References)

A Study of Linux File System Evolution, Lanyue Lu, University of Wisconsin, Madison, 2013

[Tech Report] 스마트폰 포렌식과 SQLite, Ahnlab, 2014

EXT4 File system in Linux environment: Features and performance analysis, Borislav Djordjevic, 2012

Research and application of SQLite embedded database technology, chunyue bi, 2009

SQLite 데이터베이스의 비 할당 영역에 잔존하는 삭제된 레코드 복구 기법, 전상준, 고려대학교 정보보호연구원, 2011

https://github.com/jungheum/fragmented-data-forensics, Fragmentation Analysis

Android Forensics Concept, Zlatko Jovanovic, 2011

Advanced file carving approaches for multimedia files, R Poisel, 2011