Lightning Talk - 제12회 소프트웨어무결점연구센터 워크샵 (2015년 1/26~29일)

강건한 안드로이드 앱을 위한 예외 처리

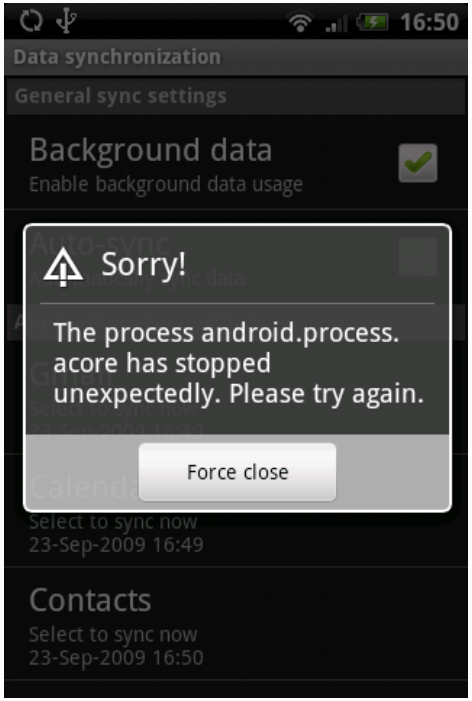
최 광 훈
(공동연구: 창병모)

모바일 소프트웨어 연구실
(http://mobilesw.yonsei.ac.kr)

컴퓨터정보통신공학부
연세대학교 원주캠퍼스

문제: 안드로이드 앱이 곧잘 죽는다.

- 예기치 않은 비 정상 종료 (심지어 시스템 리부팅)



원인: 안드로이드 앱이 곧잘 죽는 이유?

- 안드로이드 구조와 무관한 순수 자바 예외에 의존
  - 예) Activity1에서 Activity2를 띄울 때
    그럴듯하지만, 쓸모 없는 try-catch 블럭

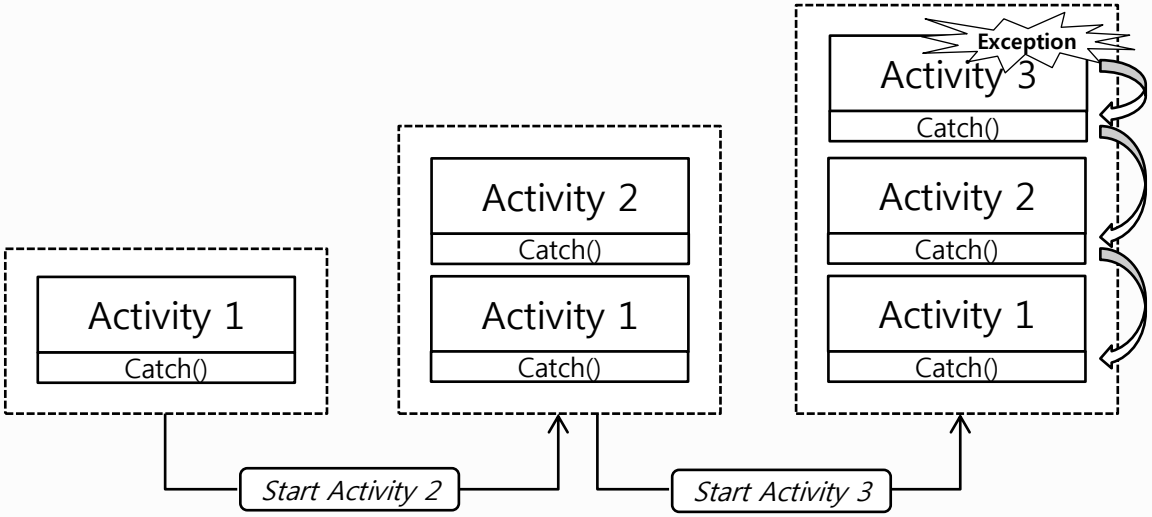
```
// Activity1          (*) Activity => 안드로이드 컴포넌트 (화면을 담당)
try {
    startActivity(   intent{target=Activity2}   );
}
catch(Exception e) {
    ...
}
```

해결책: 안드로이드 예외 (Not 자바 예외)

- 안드로이드 컴포넌트 구조를 반영한 예외 전파 및 처리 기능을 탑재한 안드로이드 컴포넌트 Lib
  - 각 컴포넌트에 "Catch 블럭" 도입
  - 피호출 컴포넌트에서 호출 컴포넌트로 예외 전파

Intra-Component Exception Handling

Inter-Component Exception Handling



실험: 9개 안드로이드 벤치마크 (7만여 LOC~)

- 장점
  - 안드로이드 예외로 앱의 강건성 향상
    (30가지 비정상 종료 케이스를 방어)
  - 안드로이드 플랫폼을 수정할 필요가 없음
  - 앱의 클래스와 메소드 구조를 그대로 유지

- 오버헤드
  - 예외 Lib 기반 Rewriting (132 LOC, 4.5%)
  - 앱의 바이너리 크기 증가 (4~5Kbytes, 8%)
  - 예외 레이어로 인한 시작 시간 지연(0.03초, 7%)

강건한 안드로이드 앱을 위한 예외 처리

- 더 자세한 이야기
  - 안드로이드 예외 Lib 상세 구현 방법
  - Activity 확장 및 다른 컴포넌트에 대해 적용
    - 프레그먼트
    - 서비스, 방송수신자, 컨텐트프로바이더
  - 벤치마크 실험 상세 결과
- 참고 논문
  - Kwanghoon Choi & Byeong-Mo Chang, **A Lightweight Approach to Component-Level Exception Mechanism for Robust Android Apps**, Jan. 2015.
    (Submitted for Publication)

안드로이드 시스템에서 SW 안전성 연구(2011~)

• 진행 한 일 • *진행하고 있는 일*

모바일SW (안드로이드)	PL 설계 기술 응용	프로그램 자동 분석	프로그램 검증
안드로이드 프로그램 분석	• 안드로이드 의미 정의 (액티베이션 흐름)	• 안드로이드 액티베이션 흐름 분석 • *안드로이드 악성코드 검출 분석*	
안드로이드 앱의 Robustness 향상 방법	• 안드로이드 예외 의미 • 안드로이드 예외 Lib 설계		• *예외 Lib의 무결성 검증*
	• 인텐트 스펙 설계	• *인텐트 스펙과 코드의 일관성 분석*	• 실행시간 인텐트 검증
NFC 플랫폼	• NFC 프로그램 메시지 (vs. 기존 NFC 메시지)		• NFC 프로그램 실행시간 모니터링 • *실행 전 NFC 프로그램 검증*

해결책1: 컴포넌트 안에서 예외 처리

- 예외 기능 탑재한 컴포넌트 클래스
  - 예) Activity => ExceptionActivity
- 컴포넌트 콜백 메소드를 감싸는 try-catch 담장
  - 예) onCreate 메소드 => OnCreate 메소드
  - 예) ExceptionActivity의 onCreate 메소드
    - try { OnCreate(); } catch(Throwable exn) { ... }
- 각 컴포넌트에 Catch 메소드 도입
  - 컴포넌트 별 “Catch 블럭”

해결책2: 컴포넌트간 예외 전파

- Activity1에서 Activity2를 띄우고, Activity2에서 발생한 예외를 Activity1에서 처리하는 방법

```
// Activity1
TryActivity ( intent{target=Activity2},
        new Catch() {
          boolean handle(Throwable   exn) {
              if (  exn  instanceof   Exception  ) {
                    ...
                    return true;
              }
              else return false;
          }
} );
```