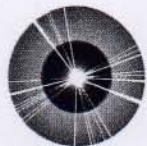


제13권 제1호  
Vol. 13 No. 1



한국정보과학회  
Korean Institute of Information Scientists and Engineers



한국정보처리학회  
Korean Information Processing Society



# 2011

## 한국 소프트웨어공학 학술대회 논문집(안)

(Preliminary)

Proceedings of 2011 Korea Conference on  
Software Engineering

- 일시 : 2011년 2월 9일(수)~11일(금)
- 장소 : 한화 위닉스 파크(강원도 평창)

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학 소사이어티  
한국정보처리학회 소프트웨어공학 연구회  
한국전자통신연구원

후원 : 삼성 SDS, 비트컴퓨터, (주)모아소프트, (주)이웨이파트너즈,  
단국대학교SERC(소프트웨어공학연구센터), 고려대학교 고신뢰  
융합소프트웨어 연구센터, 서강대학교 SW 요구 및 검증공학기술  
연구센터, 숭실대학교 모바일서비스 SW공학센터, 포항공대  
융합소프트웨어개발 연구센터

요약 해석을 이용한 정적 실행시간 분석을 위한 실행 경로의 외부 입력 의존성 분석--171  
김윤관, 신 원, 장천현 (건국대학교)

CBMC 모델검증 도구를 이용한 엔터프라이즈 소프트웨어의 검증-----179  
김성희, 권기현 (경기대학교)

## C1: 임베디드 소프트웨어 아키텍처 및 설계 I

MDD 기반의 임베디드 소프트웨어 재구조화-----189  
엄석환, 김진규, 강성원 (KAIST)  
이민광, 김태호, 김진혁 (삼성전자)

멀티코어 기반의 멀티미디어 어플리케이션 성능 향상을 위한 소스 코드 클러스터링  
기법-----202  
정진우, 박수진, 박수용 (서강대학교)

임베디드 시스템의 관점지향 모델링을 위한 SysML 의 확장-----210  
이재욱, 김두환, 김종필, 홍장의 (충북대학교)

PPC 를 활용한 DRAM SSD 기반 서버의 안정성 향상-----219  
박성범, 유영진, 신동인, 엄현상, 엄현영 (서울대학교)

## C2: 소프트웨어 시험 및 검증 I

NMS 환경의 효율적 운영 검증을 위한 테스트 관리 시스템 구축-----231  
김명완, 김우열, 손현승, 김영철 (홍익대학교)

Applying a Rule based Classification Technique with Feedback to Find Bugs in Source  
Code-----235  
Tao Zhang, Byungjeong Lee, Jaeho Lee (서울시립대학교)

테스트 자동화를 통한 K 社의 테스트 프로세스 개선 사례 연구-----243  
서정덕, 민상윤 (KAIST)

금융 부가가치 망 환경의 실시간 트랜잭션 검증을 위한 테스트케이스 연구-----249  
김규원, 김우열, 손현승, 김영철 (홍익대학교)

# 금융 부가가치 망 환경의 실시간 트랜잭션 검증을 위한 테스트 케이스 연구

김규원\*, 김우열, 손현승, 김영철

(주) KOVAN\*

홍익대학교 컴퓨터정보통신 소프트웨어공학전공

[kkw1206@kovan.com](mailto:kkw1206@kovan.com)\*,

{john, son, bob}@selab.hongik.ac.kr

**요약:** 본 논문에서는 금융 거래를 일종의 연속적인 실시간 트랜잭션으로 정의하고, 발생 가능한 연속적인 트랜잭션의 수집과정, 각 단위 트랜잭션에 대한 정의, 연속적인 트랜잭션 체인에 대한 동적인 분석 진행한다. 이 과정을 통해 얻어낸 명세를 통해 Use Case 다이어그램으로부터 테스트 시나리오에 관련된 정보를 추출하고, 이러한 과정을 거쳐서 최종으로 테스트 케이스를 생성한다.

**핵심어:** 트랜잭션, 테스트케이스, 테스트시나리오, Use case 다이어그램, VAN(Value Added Network)

## 1. 연구배경

VAN사는 결제 시스템을 이용하는 가맹점과 카드사 및 여타 금융기관을 연결해주는 중계기관이다. 신용카드 결제 시스템은 신용카드 단말기, VAN(Value Added Network)사, 카드사들이 서로 연동되어 있다[1]. 전화선과 단말기를 이용한 거래승인, POS 시스템, HOST 서버를 이용한 거래승인 등 다양한 결제 모델이 나타났다[2].

다양한 결제모델들의 수용으로 VAN사는 시스템도 점점 복잡해 졌다. 또한, 가맹점들과 금융기관의 요구사항을 모두 수용해야 하는 구조이기 때문에 시스템의 유지보수가 빈번하게 이루어진다[3]. 시스템의 변경되는 횟수가 많은 만큼 시스템에 대한 테스트도 같이 진행된다. 적용된 프로세스에 대한 테스트진행은 개발자들이 테스트의 역할을 같이 한다. 테스트 케이스는 새로운 요구사항과 기존 시스템의 관계를 고려하여 작성되어야 한다. 하지만, 테스트 케이스 작성을 위한 체계적인 방법이 없을 때, 개발자의 경험과 역량에 따라 테스트 케이스는 다르게 작성 된다[4,5]. 이것은 테스트의 신뢰성과

연결되기 때문에 중요한 문제이다. 또한, 이런 테스트 케이스를 이용하여 테스트를 수행하는 것은 테스트의 역량에 의존하게 된다[6].

본 논문은 VAN(Value Added Network) 환경에서 발생하는 실시간 트랜잭션을 검증하기 위한 테스트 케이스를 생성하는 방법을 제안하는데 목적이 있다. 테스트 케이스를 생성하기 위하여, VAN 환경에 대해 조사하고, 트랜잭션의 종류별로 시나리오를 구성하고 모델링 한다. 모델링 한 다이어그램들로부터 테스트 케이스를 생성한다. 본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구로서 VAN에 대한 설명을 한다. 제 3장은 VAN 시스템의 실시간 트랜잭션에 대한 보여준다. 제 4장에서는 제안된 테스트 기법으로 테스트 케이스를 생성하는 과정을 보여준다. 마지막으로 5장에서는 결론 및 향후 연구 과제를 언급한다.

## 2. 관련 연구

국내에서 VAN(Value Added Network)이 의미하는 보편적 의미는 신용카드에 관련된 통신업무 및 정보처리 관련 업무를 하는 사업자를 말한다. VAN서비스의 구성은 가맹점, VAN사, 카드사가 연계되어 이루어진다. 신용카드 단말기, 인터넷거래, 대형점의 HOST 서버 등은 전화선, 인터넷, 전용선 등 다양한 회선으로, 각각의 통신전문으로 VAN사의 승인 시스템으로 전송하면, VAN사는 각각의 요청전문에 대해서 발급한 카드사로 전송을 하여, 거래승인을 이루어지게 해준다.

VAN(Value Added Network)은 가맹점, VAN사 승인시스템, 카드사 3가지 다른 기관이 연동하여 서비스가 이루어진다. 가맹점은 카드 소지자가 카드를 가져오면, 자신이 가지고 있는 신용카드 결제 시스템(단말기, POS 등)에 거래승인을

\* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2010-0012117)과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

요청한다. 요청한 거래승인 데이터는 PSTN, INTERNET, TCP/IP 전용망 등을 통해서 VAN사로 전달된다. VAN사는 거래 요청정보의 재 가공을 통해 금융기관과 금융거래를 완료하고 가맹점으로 결과를 전송한다.

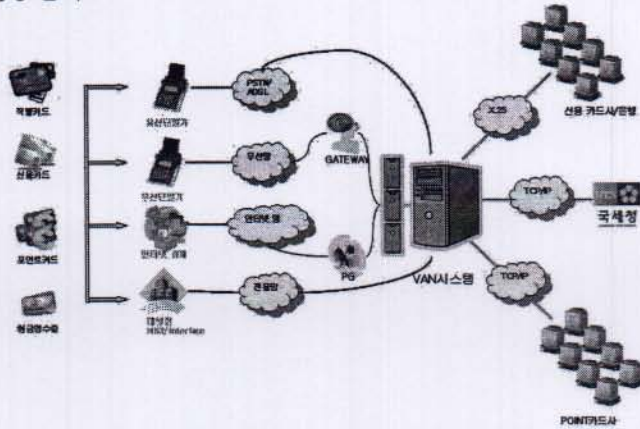


그림 1 VAN 서비스 구성도

그림 1은 VAN사의 서비스의 구성도를 보여준다. 금융거래가 가능한 카드들과 가맹점, 통신회선, VAN 시스템, 금융기관으로 구성되어 있다.

### 3. VAN 시스템의 실시간 트랜잭션

금융 VAN의 트랜잭션은 실시간 트랜잭션과 후 처리 트랜잭션으로 나누어진다. 실시간 트랜잭션은 가맹점에서 발생한 금융거래 결과를 확인하기까지의 과정이다. 후 처리(Post Processing) 트랜잭션은 실시간 트랜잭션이 완료된 후, VAN 시스템에서 카드사와 배치(Batch)작업으로 이루어지는 트랜잭션이다.

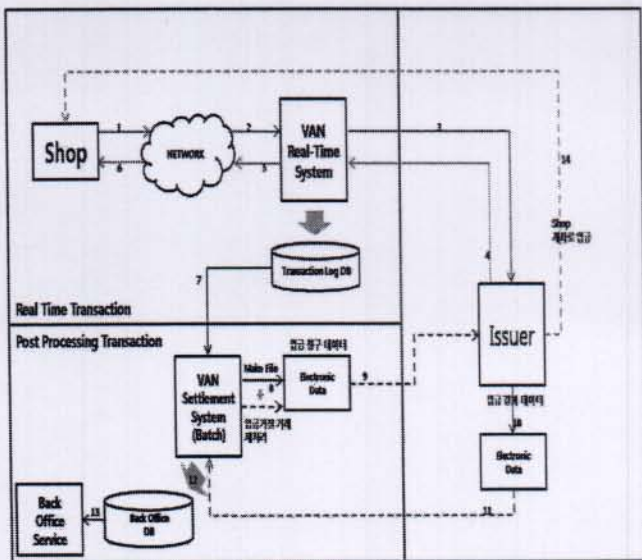


그림 2 금융 VAN 트랜잭션 처리과정

VAN 시스템에 기록된 실시간 트랜잭션 데이터들은 파일로 생성되어 카드사로 전송된다. 카드사는 수신 받은 파일에 대한 금융업무 처리를 한다. 그림 2는 금융 VAN 에서 발생하는 두 가지 트랜잭션의 처리 과정을 나타낸 그림이다.

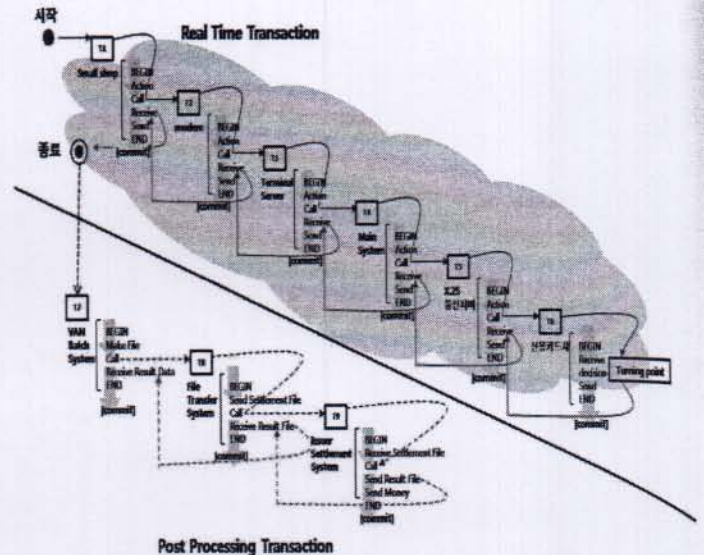


그림 3 금융 VAN 트랜잭션 체인

그림 3은 실시간 트랜잭션과 후 처리 트랜잭션의 관계를 나타낸다. 실시간 트랜잭션 영역에서 트랜잭션이 완료된 것을 시작으로 후 처리 트랜잭션이 시작되어 VAN의 배치(batch)시스템에서 파일을 생성하고 파일 전송 시스템에서 파일을 카드사로 전송하고, 카드사가 수신된 파일의 결과를 VAN으로 다시 송신하고, VAN이 수신하는 것으로 후 처리 트랜잭션이 완료된다. 가맹점에서의 입금은 VAN사와 별도로 카드사 자체적으로 진행 된다.

### 4. 제안된 테스트 기법으로 테스트 케이스를 생성

VAN환경에서 발생하는 포인트 카드 트랜잭션에 대하여 적용한다. 포인트 조회, 포인트 적립, 포인트 사용, 포인트 사용 취소 4가지의 트랜잭션이 있다. 포인트 카드 트랜잭션에 적용하기 위해서는 테스트 카드 정보와 테스트 가맹점 정보가 필요하다. 이것을 위해서 가상의 테스트 카드번호, 가맹점번호를 설정하고 이것을 이용하여 실제 테스트 케이스를 생성해 본다. 테스트 가맹점들은 POS 장비를 사용하는 Middle Shop으로 설정한다. 본문에서는 포인트 조회에 대해서 테스트 케이스를 생성해 보도록 하겠다.

그림 4는 가맹점에서 발생한 포인트 트랜잭션 중 포인트 조회 트랜잭션을 나타낸 그림이다. 포인트 트랜잭션은 5개의 단위 트랜잭션으로 구성된다.

TS(Transaction-Shop) 트랜잭션은 가맹점에서 발생하는 트랜잭션이다. 포인트 조회 정보를 입력하고, 포인트 거래요청을 하고, 응답을 통해 결과를 확인하면 트랜잭션이 완료된다.

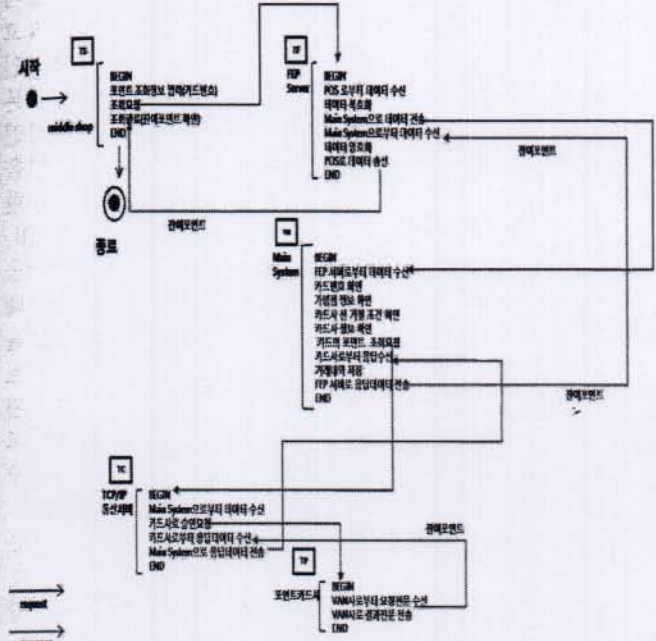


그림 4 포인트 조회 트랜잭션 동적 분석

TF(Transaction-FEP) 트랜잭션은 FEP 서버에서 발생하는 트랜잭션이다. 시작조건은 TS 트랜잭션이 거래요청을 하는 것이다. TF 트랜잭션은 수신 받은 데이터의 복호화, Main System으로 데이터 전송, Main System으로부터 데이터 수신, 데이터암호화, 가맹점에서의 데이터 송신, 데이터 수신으로 구성되어 있다. TM(Transaction-Main system) 트랜잭션은 TF 트랜잭션이 데이터를 전송하면서 시작 된다. TM 트랜잭션은 카드 정보, 가맹점 정보, 선 거절 조건 등을 검사 한 후에 포인트 조회 요청을 한다. 카드사로부터 응답을 수신 받고, 거래내역 저장 후 FEP서버로 응답데이터를 전송하면 트랜잭션은 완료된다.

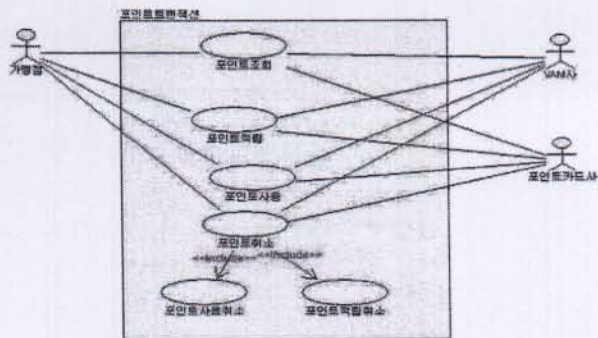


그림 5 포인트 트랜잭션 Use Case

TC(Transaction-Comm.)은 TCP/IP 통신서버에서 발생하는 트랜잭션이다. Main System 으로 부터 데이터를 수신 받아서, 포인트 카드사로 거래 요청을 하고, 응답을 수신 받아서 Main System으로 전송하면 거래가 트랜잭션이 완료된다. TP(Transaction-Point Issuer) 는 포인트 카드사 트랜잭션이다. 포인트 카드사는 VAN사로부터 결과를 수신 받고, 결과를 전송하면 트랜잭션이 완료된다. TS->TF->TM->TC->TP->TC->TM->TF->TS의 흐름으로 이루어진다. 트랜잭션이 완료되기 위해서는 다음 단계의 트랜잭션이 먼저 완료가 되어야 한다.

그림 5는 포인트의 대표적인 트랜잭션인 포인트 조회, 적립, 사용, 취소에 대한 Use Case 다이어그램이다. 표 1은 포인트 조회 Use Case 시나리오이다.

표 1 포인트 조회 Use Case 시나리오

Use Case Name	포인트 조회
Participating Actor	가맹점, VAN 사, 포인트 카드사
Flow Event	<ol style="list-style-type: none"> <li>1. 테스트 가맹점에서 테스트 카드 번호를 사용해 포인트 조회를 요청한다.</li> <li>2. VAN 사는 요청 가맹점 정보와 카드정보를 확인 후 포인트 카드사로 조회 요청을 보낸다.</li> <li>3. 포인트 카드사는 VAN 사로부터 수신 받은 카드의 잔여 포인트를 VAN 사로 전달한다.</li> <li>4. 잔여 포인트 내역을 수신 받은 VAN 사는 거래내역을 저장하고 테스트 가맹점으로 응답을 전송한다.</li> <li>5. 거래내역을 수신 받은 테스트 가맹점은 포인트 점수를 확인한 후 거래는 종료된다.</li> </ol>
Entry Condition	테스트 가맹점에서 테스트 카드로 포인트 조회 요청
Exit Condition	잔여 포인트 확인

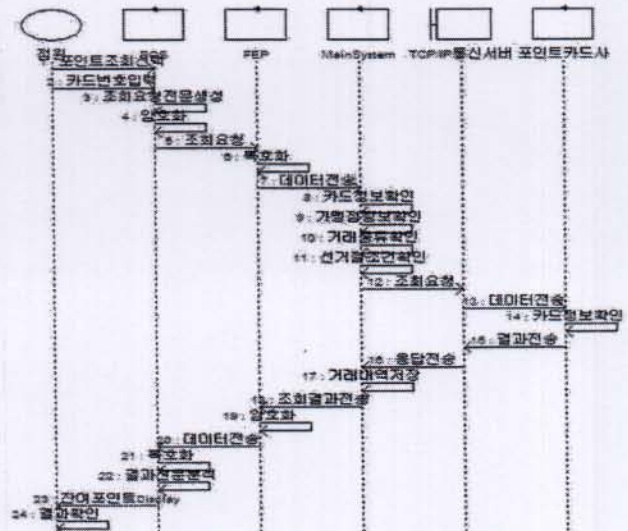


그림 6 포인트 조회 순차 다이어그램

그림 6는 포인트 조회 트랜잭션을 순차 다이어그램으로 표현한 것이다. 순차 다이어그램은 ECA 결정테이블과 ECA 모델링으로 변환 할 수 있다. 순차 다이어그램 메시지의 번호로 E(event), C(condition), A(Action)를 구성한다.

표 2 포인트 조회 테스트 시나리오

트랜잭션	테스트 시나리오
점원(TS)	1-점원은 포인트 조회선택을 한다. 2-카드번호를 입력한다. Pos 가 잔여 포인트를 display 한다(TP commit) 24-포인트 조회 결과를 확인한다.(TS commit)
POS(TP)	점원이 포인트 조회선택과 카드번호를 입력한다.(시작조건) 3-포인트 조회요청 전문을 생성한다. 4-요청전문의 암호화를 수행한다. 5-FEP 로 조회 요청을 한다. FEP로부터 응답을 수신한다. ( TP commit) 21-응답 데이터를 복호화 한다. 22-결과 전문을 분석한다. 23-결과를 점원에게 Display 한다.(TP commit)
FEP(TF)	POS로부터 조회요청을 받는다(시작조건). 6-복호화를 수행한다. 7-MainSystem 으로 데이터를 전송한다. MainSystem 으로부터 결과를 수신한다. (TM commit) 19- 암호화를 수행한다. 20- 결과를 POS 에 전송한다. (TF commit)
MainSystem (TM)	FEP로부터 데이터를 수신 받는다(시작조건). 8,9,10,11-카드정보, 가맹점정보, 거래종류, 선 거절 조건을 확인한다. 확인결과가 false 이면 거래내역을 저장하고 거절결과를 POS 로 전송한다.(false 일 경우, transaction 은 commit) 12-확인결과가 true 이면 TCP/IP 통신서버로 조회요청을 한다. TCP/IP 로부터 응답을 수신 받는다. (TC commit) 17-거래내역을 저장한다. 18-결과를 FEP 에 전송한다. (TM commit)
TCP/IP 통신서버 (TC)	MainSystem 으로부터 조회요청을 받는다(시작조건) 13-포인트카드사로 데이터를 전송한다. 포인트 카드사로부터 결과를 수신 받는다. (TI commit) 16-MainSystem 으로 응답을 전송한다. (TC commit)
포인트 카드사 (TI)	TCP/IP 통신서버로부터 데이터를 수신 받는다.(시작조건) 14-카드의 정보를 확인한다. 15-결과를 전송한다. (TI commit)

표 2는 포인트 조회의 테스트 시나리오다.

테스트케이스 번호	단위테스트케이스	테스트케이스	테스트케이스(가맹점)	테스트케이스(카드사)	시작조건	ACTION	형성결과
001	TS	TS01	TS0101	TS010101	카드번호 입력	카드번호를 입력한다	카드번호가 입력된다
002	TP	TP01	TP0101	TP010101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
003	TP	TP02	TP0201	TP020101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
004	TP	TP03	TP0301	TP030101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
005	TP	TP04	TP0401	TP040101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
006	TP	TP05	TP0501	TP050101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
007	TF	TF01	TF0101	TF010101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
008	TF	TF02	TF0201	TF020101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
009	TF	TF03	TF0301	TF030101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
010	TF	TF04	TF0401	TF040101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
011	TF	TF05	TF0501	TF050101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
012	TF	TF06	TF0601	TF060101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
013	TF	TF07	TF0701	TF070101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
014	TF	TF08	TF0801	TF080101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
015	TF	TF09	TF0901	TF090101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
016	TF	TF10	TF1001	TF100101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
017	TF	TF11	TF1101	TF110101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
018	TF	TF12	TF1201	TF120101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
019	TF	TF13	TF1301	TF130101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
020	TF	TF14	TF1401	TF140101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
021	TF	TF15	TF1501	TF150101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
022	TF	TF16	TF1601	TF160101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
023	TF	TF17	TF1701	TF170101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
024	TF	TF18	TF1801	TF180101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
025	TF	TF19	TF1901	TF190101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
026	TF	TF20	TF2001	TF200101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
027	TF	TF21	TF2101	TF210101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
028	TF	TF22	TF2201	TF220101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
029	TF	TF23	TF2301	TF230101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
030	TF	TF24	TF2401	TF240101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
031	TF	TF25	TF2501	TF250101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
032	TF	TF26	TF2601	TF260101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
033	TF	TF27	TF2701	TF270101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
034	TF	TF28	TF2801	TF280101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
035	TF	TF29	TF2901	TF290101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다
036	TF	TF30	TF3001	TF300101	카드번호를 입력	카드번호를 입력한다	카드번호가 입력된다

그림 6 포인트 조회 테스트 케이스

그림 6는 테스트 시나리오 기반으로 생성한 포인트 조회 테스트 케이스이다.

### 5. 결론

VAN 환경상의 테스트는 요구사항 명세에 따른 기능적인 테스트가 이루어진다. 테스트 케이스는 요구사항을 기반으로 기능적 테스트 위주로 작성이 된다. 테스트 케이스를 생성하는 방법론을 가지고 있지 않는 회사가 많기 때문에, 누적된 경험적인 것을 바탕으로 작성하는 경우가 많다. 본 논문에서는 VAN의 금융거래를 일종의 연속적인 실시간 트랜잭션으로 정의하고, 트랜잭션 분석을 통한 정보를 기반으로 트랜잭션 모델링을 통한 테스트 케이스 생성을 제안했다. 트랜잭션의 PATH, ACTION 분석을 통하여 트랜잭션들의 동적인 분석을 하였고, 테스트 케이스를 생성한다. 이 과정을 통해 생성된 테스트 케이스는 연속적인 트랜잭션 체인들 간의 Action 관계를 통해서 더 세부적인 테스트 케이스의 생성이 가능했다. 제안된 방법으로 생성된 테스트 케이스는 요청에 대한 결과는 물론 구성하고 있는 단위 트랜잭션들의 Action도 테스트 할 수 있으므로, 더 많은 커버리지를 가질 수 있다.

향후 연구 과제로서는 본 논문에서 다루지 않은 병렬적인 트랜잭션, 동시 다발적으로 발생하는 트랜잭션에 대한 테스트 케이스 연구를 진행 할 예정이다.

### 참고문헌

- [1] 황삼생, 양재영, 문희철, "VAN 을 기반으로 한 컴퓨터와 네트워크에 의한 전자상거래에 관한 연구", 한국국제경영관리학회, 학술발표대회 논문집 2001.11 pp. 159-185,2001
- [2] 이기현,곽홍희, "우리나라 금융 VAN 의 현재와 미래", <<정보과학회지>>, 제 6 권 제 5 호 pp. 15-22,1988
- [3] 김규원,박보경,장우성,문소영,김영철, "역공학 기반 금융 VAN 연동 서비스 모델링을 통한 시스템 구현 연구", 한국정보과학회, 제 37 권 제 1호(B) 2010.6, pp. 92-96,2010
- [4] 서광익,최은만, "객체지향 소프트웨어를 위한 주요 블랙박스 테스트 기법들의 비교", 정보과학회, 제 33 권 제 1 호, 2006.1 pp. 1-17,2006
- [5] 이지현, 노혜민, 유철중, 장옥배, 이준옥, "상호운용성 테스트를 위한 테스트케이스 생성 기법", 한국정보과학회, 제 33 권 제 1 호, 2006.1, pp. 44-58,2006
- [6] 김재련,김성태: "Software 의 Quality 를 향상시키기 위한 체계적인 testing 기법", 한국정보과학회, 제 10 권 제 2 호, 1983.10 pp. 48-57,1983