친밀한 관계 분석을 위한 안드로이드 기반 액티비티 매니저 개발

김바울*, 김경덕**, 김상욱*
*경북대학교 전자전기컴퓨터학부
**위덕대학교 컴퓨터공학과
e-mail:bwkim@woorisol.knu.ac.kr

A Development of Activity Manager based on Android for Analysis of Close-relationship

Paul Kim*, Kyungdeok Kim**, Sangwook Kim*
*School of Electrical Engineering and Computer Science,
Kyungpook National University
**Dept. of Computer Engineering, Uiduk University

요 약

모바일 사용자의 주변 상태를 자동으로 인지하는 상황 인지 기술은 사람과 컴퓨터의 상호작용에 있어서 중요하다. 모바일 환경에서 사용자의 상황 인지를 하기 위해서는 플랫폼단에서 상황 정보를 수집하고 처리하는 미들웨어와 대량의 정보를 저장하고 분배해주는 서버가 필요하다. 특히 모바일 사용자간 친밀한 관계를 인지하기 위해서는 동시다발적으로 발생하는 상황 정보들을 효율적으로 처리하는 시스템이 필요하며 전체 성능에 영향을 적게 주어야 한다. 따라서 본 논문에서는 상황 인지 플랫폼에서 다양한 상황 정보들을 처리하는 액티비티 매니저 모듈을 구현한다. 제안하는 모듈은 상황 정보를 플랫폼간 통신이 가능한 형태로 변환하는 기술을 포함하며 상황 정보의 알림, 요청 작업을 실시간으로 처리하여 공유함으로써 사용자간 친밀한 관계를 분석할 수 있다.

1. 서론

사용자의 주변 상황 정보는 동시다발적으로 무수히 발생하고 의미 없이 사라진다. 이런 상황 정보는 사용자의 현재 상황을 설명할 수 있는 정보로 사용자의 감정, 활동, 친밀도 같은 고수준 상황 정보를 추론할 수 있다. 모바일 기기는 이런 고급 정보를 실시간으로 인식함으로써 사용자와 컴퓨터의 상호작용에 더 나은 컴퓨팅 환경을 제공해 줄 수 있다. 또한 서비스 제공자나 개발자에게 API 형식으로 사용자의 상황 정보를 제공하여 다양한 서비스를 창출할 수 있다. 그에 따라 상황 정보를 수집하고 공유하며 인식하는 기술이 많이 연구되고 있으며 이 정보들을 소셜 네트워크와 접목시키는 연구로 발전하고 있다.

따라서 본 연구에서는 스마트폰 플랫폼 차원에서 상황 정보를 서비스할 수 있는 미들웨어로 액티비티 매니저 모듈을 제안한다. 제안하는 모듈은 친밀한 관계를 분석하기 위해 상황 정보들을 수집하며 동시에 여러 개의 정보가 발생하더라도 전체 성능에 영향을 덜 미치며 효율적으로 처리할 수 있게 설계하였다. 또한 플랫폼간 인터랙션을 지원하여 사용자 지인들의 상황 정보를 수집하여 사용자 상황 정보로 흡수할 수 있다.

2. 관련 연구

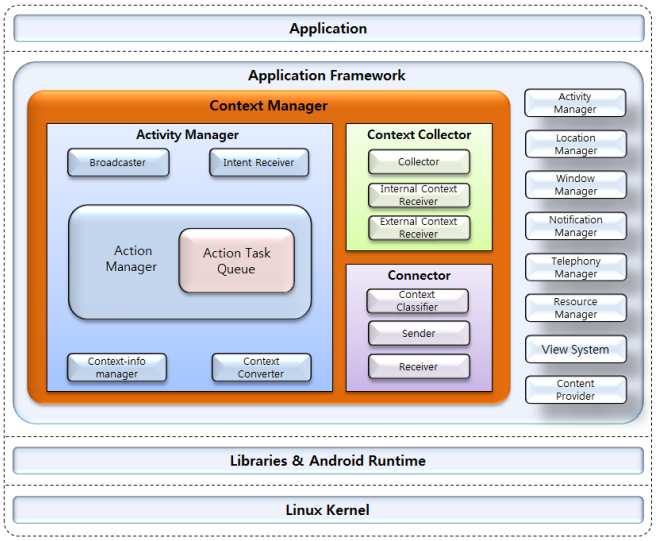
Raento 등은 상황 인지가 가능한 스마트폰 플랫폼과 그 플랫폼을 이용한 사용자의 상황 분석 방법을 연구하였다. 또한 이 모바일 미들웨어를 이용하여 사용자와 지인들의 상황 정보를 화면에 표시하는 ContextContacts를 구현하였다[2]. Beach 등은 사용자의 정보를 검색하고 보여주기 위해서 소셜 네트워크와 모바일 플랫폼을 고려한 에코 시스템을 제안하였다[3]. 이런 시스템들은 스마트폰에 내장된 GPS, 가속도 센서, WiFi 등의 센서 값들을 수집하여 다른 사용자에게 전달하는데 초점이 있다. 하지만 센서 값 같은 데이터는 활용 범위가 낮고 정보의 수준이 낮은 상황 정보들이다. 이런 문제점들을 해결하기 위해 고수준 컨텍스트를 추론하는 연구들이 많이 진행되었다. Andersen 등은 사용자의 Activity, Status, Relation, Vicinity를 추론하는 모바일 시스템을 연구하였다[4]. Ankolekar 등은 사용자와 사용자의 전화번호부에 있는 사람간의 관계를 기반으로 모바일 소셜 네트워킹 어플리케이션을 구현하였다[5]. Sorathia 등은 웹의 소셜 네트워킹 서비스를 모바일 환경에서 지원하고 그에 따른 여러 상황을 추론하는 시스템을 제안하였다[6]. 하지만 이런 연구들은 플랫폼 차원에서 고수준 상황 정보 수집 및 공유를 지원하지 않으며 사용자의 정보를 입력 값으로 받아야하는 연구들이 대부분이다. 또한 상황 인식 기능을 포함하는 시스템들은 아직 상황 공유에 머물러 있다. 따라서 본 연구에서는 친밀도 분석을 위해 플랫폼 차원에서 상황 정보 수집과 공유를 지원하는 액티비티 매니저를 제안한다. 이를 이용하여 친밀 관계 모델을 제안하고 분석한다.

3. 상황 인지 플랫폼

스마트폰 플랫폼 차원에서 상황 정보를 실시간으로 수집하여 공유하려면 그 기능이 플랫폼에 포함되어 유기적으로 동작해야 한다[8]. 따라서 본 연구에서는 플랫폼간 상황 정보 공유 미들웨어인 액티비티 매니저를 포함하는 안드로이드 기반 상황 인지 플랫폼을 개발하였으며 동시 다발적으로 발생하는 상황 정보를 효율적으로 처리하는 상황 공유 메커니즘을 구현한다.

3.1 시스템 구조

상황 인지 플랫폼은 안드로이드를 기반으로 하고 그림 1과 같이 리눅스 커널, 라이브러리, 어플리케이션 플레임워크, 어플리케이션 계층으로 이루어져 있으며 상황 인지를 하기 위한 컨텍스트 매니저 모듈이 프레임워크 계층에 포함되어 있다.

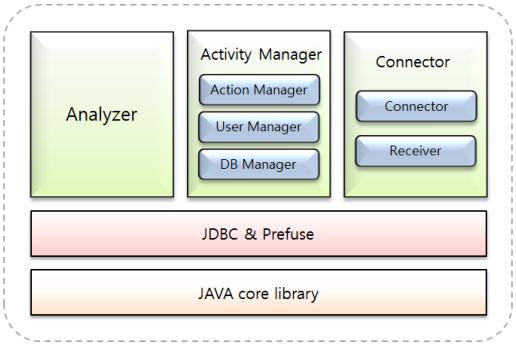


(그림 1) 상황 인지 플랫폼 시스템 구조

컨텍스트 매니저는 다시 액티비티 매니저, 커넥터, 컨텍스트 콜렉터 모듈로 구성된다. 각 모듈은 각자의 핵심 스레드를 가지고 있으며 독립적이고 유기적으로 동작한다. 이 때 각 모듈은 동기화가 되어야 하며 그 관리 기능의 우선권은 액티비티 매니저 안에 액션 매니저가 갖는다. 또한 어플리케이션 플레임 워크 레이어에 위치해 있기 때문에 다른 프레임워크에 쉽게 접근하고 관련 라이브러리 함수를 효율적으로 호출할 수 있다. 컨텍스트 매니저의 메인 달빅 프로세스는 UI 없이 백그라운드 서비스로 실행되며 모바일 기기의 퍼포먼스에 영향을 작게 주도록 설계 하였다. 전체적인 생명 주기는 모바일 기기가 구동할 때 실행하여 전원이 꺼질 때 리소스를 해제하고 종료 된다. 또한 컨텍스트 매니저가 각 컴포넌트에 접근하기 위해서는 플랫폼의 관리 권한으로 각 컴포넌트를 제어하는 시스템 서비스 프로세스에 해당 프레임워크와 라이브러리를 호출함으로써 접근 가능하며 필요에 따라 상황 정보 리시버를 시스템 서비스에 등록한다. 또한 기본적인 데이터베이스 테이블을 확장하여 다양한 상황 정보를 저장하고 접속할 수 있는 구조를 가지고 있다.

3.2 컨텍스트 서버

서버는 상황 인지 플랫폼간 상황 정보를 공유하고 전달해주는 매체이다. 또한 방대한 양의 상황 정보를 DB에 저장하며 플랫폼의 요청에 의해 그 정보를 제공해 주는 역할도 담당한다. 이를 서버에서 수행하는 이유는 다양한 종류의 정보들을 관리해야하고 고수준 상황 정보 추론을 위해 많은 연산을 해야 하기 때문이다. 결국 서버는 상황 인지 플랫폼간 상황 공유와 고수준 상황 정보의 추론을 하



(그림 2) 컨텍스트 서버 시스템 구조

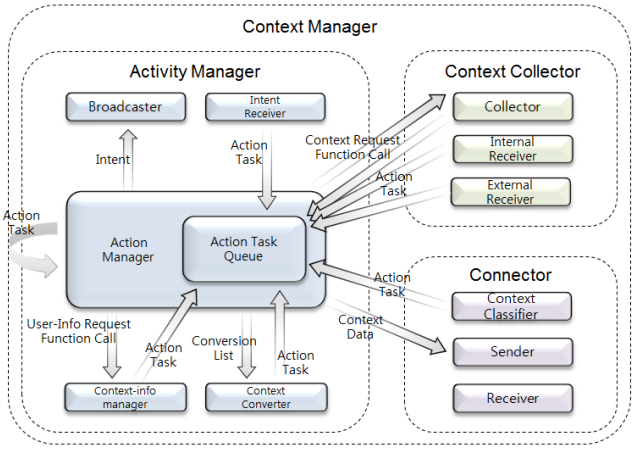
기 위해 필요하다. 따라서 전체 통신 시간이 단축되고 사용자 모바일 기기의 성능에도 영향을 주지 않는다. 서버를 구성하는 컴포넌트들은 그림 6과 같은 구조를 가지고 JAVA와 Prefuse 라이브러리를 바탕으로 하며 JDBC 커넥터로 MySql DB에 접근하도록 디자인 되었다. 그 윗 계층에 서버의 핵심 모듈들이 위치하며 상황 정보의 분석 및 전달 기능을 수행한다.

4. 친밀한 관계 분석을 위한 액티비티 매니저

모든 상황 수집 이벤트는 액티비티 매니저에 의해 발생되며 컨텍스트 콜렉터는 관련된 상황 정보를 수집한 후 액티비티 매니저에게 전달한다. 컨텍스트 서버는 이런 사용자들의 상황 정보를 바탕으로 친밀한 관계 모델에 의해 친밀도를 계산한다.

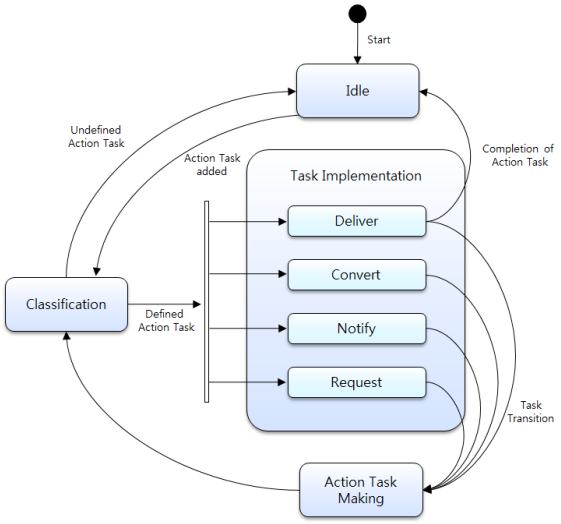
4.1 액티비티 매니저

이를 위해 액티비티 매니저는 액션 매니저, 컨텍스트 컨버터, 켄텍스트 인포 매니저, 브로드캐스터, 인텐트 리시버로 이루어진 서브 모듈을 포함하고 고유의 데이터를 주고받으며 다른 컴포넌트와 상호 운용된다. 액티비티 매니저의 입력과 출력은 그림 2와 같으며 입력의 경우 작업 순서의 복잡도를 없애고 동시에 빠른 처리가 가능한 액션 태스크 클래스로 통일되어 있다. 모든 액션 태스크는 액션 태스크 큐에 의해 관리되어지고 액션 매니저에 의해 순차적으로 실행된다. 액션 매니저는 구조적인 코드를 재사용함으로써 수행 효율을 높이는 매커니즘을 갖는다. 이를 위해 처리되



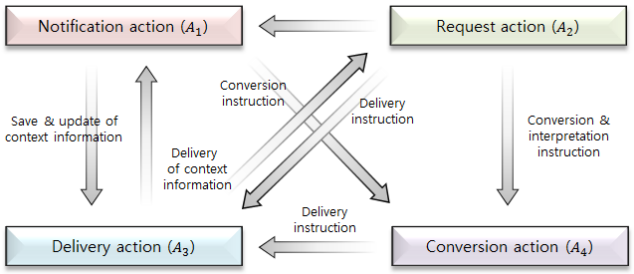
(그림 3) 액티비티 매니저의 I/O 흐름

는 액션의 성격에 따라 여러 가지 액션 그룹으로 나누어 처리하게 디자인 하였다. 따라서 액션 태스크는 반드시 여러 가지 액션 그룹 중 한 가지 액션 그룹에 속하게 된다. 이 액션 그룹은 처리 속성에 따라 컨트롤 액션과 수행 액션으로 나뉜다. 컨트롤 액션 그룹은 다시 노티피케이션 액션 그룹과 리퀘스트 액션 그룹으로 나뉘고 수행 액션 그룹은 딜리버리 액션 그룹과 컨버전 액션 그룹으로 나뉜다. 노티피케이션 액션 그룹은 어떤 상황을 수집했을 때 어플리케이션이나 컨텍스트 서버에 알려주는 작업을 수행한다. 리퀘스트 액션 그룹은 서버나 어플리케이션 레이어에서 상황 정보를 요청하는 작업이다. 딜리버리 액션 그룹은 실질적인 데이터 이동을 담당하고 컨버전 액션 그룹은 XML을 상황 정보로 또는 상황 정보를 XML로 변환하는 작업을 수행 한다. 따라서 그림 4와 같이 무슨 작업이든지 시작은 컨트롤 액션으로 시작하여 수행 액션을 거쳐 끝마치게 된다. 위 작업 그룹의 실질적인 수행은 세부 서브 모듈들이 수행 한다. 어플리케이션 레이어와의 인터랙션은 브로드캐스터와 인텐트 리시버가 담당하며 컨텍스트 인포 매니저는 수집한 상황 정보들을 저장하고 항상 최신 정보로 유지하며 어플리케이션 계층의 어플리케이션들과 DB



(그림 4) 액션 매니저의 상태 전이도

를 공유한다. 컨텍스트 컨버터는 정보를 다른 계층에 전달하기 위해 상황 정보를 XML로 변환하거나 XML을 상황 정보로 치환시켜 준다.



(그림 5) 액션 그룹간 상태 변화

액션 태스크는 작업의 타입과 모드의 조합으로 총 14개의 세부 작업으로 나뉘며 각 작업 그룹 간에 실행 순서는 그림 5와 같다. 각 태스크는 고정된 작업 내용을 가지고 그 결과는 다음 작업의 인풋으로 입력된다. 따라서 노티피케이션 작업과 리퀘스트 작업은 항상 딜리버리 액션 그룹을 거쳐 태스크간 데이터가 이동하고 컨버전 액션 그룹을 거쳐 상황 정보가 XML로 변환되거나 XML이 상황 정보로 변환된다. 이는 상황 정보 데이터의 내용과 관계없이 데이터를 처리하기 때문에 플랫폼에 고수준 상황 정보를 추가로 삽입하는 것 같은 플랫폼의 확장성을 고려한 것이다. 어떤 작업이 액션 매니저 스레드에 의해 실행되면 그 작업은 해당 데이터를 처리한 후 화살표 방향으로 전이되며 항상 딜리버리 액션 그룹에 의해 모든 작업이 종료된다. 이 작업 처리 구조는 한 개의 스레드를 위한 것으로 모든 작업 입력은 액션 매니저 큐에 저장되고 메인 스레드가 큐에서 작업을 순서대로 가져와서 처리한다. 메인 스레드는 큐의 엘리먼트가 없을 때까지 반복 처리하며 작업이 없을 때에는 작업이 들어 올 때까지 대기한다.

4.2 친밀한 관계 모델 및 분석

액티비티에서 수집되는 상황 정보를 바탕으로 친밀한 관계를 분석할 수 있으며 관계 분석을 위해 친밀한 관계 모델을 제안한다. 수식 (1)은 각 속성 간의 관계와 결합으로 친밀한 관계 cls_{AB}를 표현한 것으로 인터랙션 횟수와 지속시간이 높고 연속성과 균등성이 1에 가까울수록 사용자 A와 B의 친밀도는 높다.

$$cls_{AB} \propto (fre_{AB} + dur_{AB}) \times bet_{AB} \times con_{AB} \quad (1)$$

여기에서 fre_{AB}는 인터랙션 횟수를 dur_{AB}는 지속 시간, bet_{AB}는 연속성, con_{AB}는 연속성이다. 각 속성들은 사용자마다 모바일 기기를 많이 사용하는 사용자와 적게 사용하는 사용자 모두 친한 지인과는 인터랙션이 비교적 많고 중요하게 생각하는 매체가 서로 다르기 때문에 그에 따른 가중치 값을 계산해야 한다.

$$fre_{AB} = \sum_{k=0}^{n} \delta_k \times W_k,\ dur_{AB} = \sum_{k=0}^{n} \delta_k \times W_k, \quad (2)$$

일반적으로 인터랙션의 횟수와 지속 시간이 높을수록 가까운 관계이다. 따라서 한 사용자 전체 소셜 인터랙션 횟수에서 특정 지인의 비중을 계산함으로써 빈도수와 지속 시간 수식 (2)를 일반화 할 수 있다. 지속성 con_{AB}는 사용자 A에 대해 사용자 B가 얼마나 끊임없이 인터랙션을 이어왔는지를 측정한다. 이는 주어진 시간에서 인터랙션이 발생한 횟수를 나타내는 포아송 누적 분포로 식 (3)과 같다.

$$con_{AB} = \frac{\lambda^x \times e^{-\lambda}}{x!} \tag{3}$$

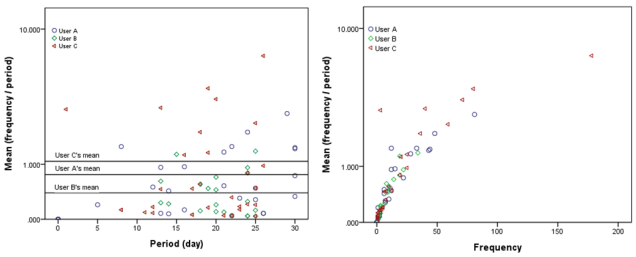
친밀한 관계에서 균등성은 상호 간의 참여를 기반으로 하기 때문에 중요하다. 예를 들어 사용자 B가 스팸 문자나 전화같이 일방적으로 많은 인터랙션을 사용자 A에게 취해 왔다고 해서 친밀한 관계라고 할 수 없다. 따라서 균등성 수식 (4) bet_{AB}는 상호 인터랙션 횟수를 비교함으로써 일방적인 관계를 필터링한다. 이는 fre_{AB}와 dur_{AB}가 높더라도 균등성이 낮으면 상대적으로 친밀도가 낮게 나오게 하기 위함이다.

$$bet_{AB} = \sum_{k=0}^{n} \left(\frac{fact_{sm}(k)+1}{fact_{bg}(k)} \times W_k \right) \tag{4}$$

$$\begin{cases} in > out, \; fact_{bg} = in, fact_{sm} = out \\ in \le out, \; fact_{bg} = out, fact_{sm} = in \end{cases}$$

5. 평가

상황 정보의 수집과 공유의 원활한 처리가 이루어지는 것을 보기 위해 실제 안드로이드 스마트폰에서 구현한 액티비티 매니저를 내장하여 평가하였다. 사용자간 상황이 공유되면 상호 간에 정보가 일치하는지를 판단하기 위해 스마트폰의 통신 이력을 수집하여 분석하였으며 그 결과는 그림 6과 같다.

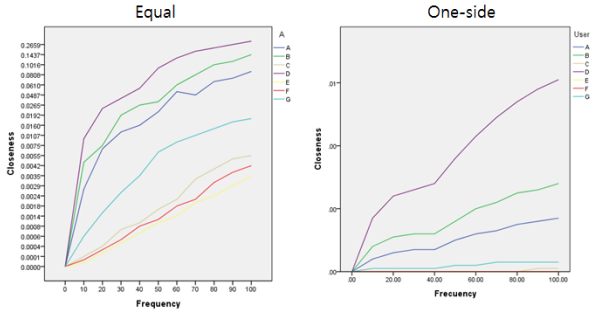


(그림 6) 사용자간 통신 이력 비교

각 사용자의 통신 이력을 분석한 결과 스마트폰에 동일한 액티비티 매니저가 실시간으로 동작하기 때문에 사용자간 친밀한 관계를 그림 7과 같이 분석할 수 있다. 또한 사용자마다 사용 매체의 비중이 달라도 사용자간 통신 이력 공유는 상호 효율적으로 처리된다.

6. 결론

모바일 사용자의 주변 상황을 인지하기 위해서는 플랫폼 차원에서 상황 정보 수집과 공유를 지원하는 상황 인지 플랫폼과 플랫폼을 이어주는 서버가 필요하다. 특히 동



(그림 7) 소셜 인터랙션의 균등성 비교

시다발적으로 무수히 발생하는 상황 정보들을 전체 시스템 성능에 거의 영향을 주지 않고 처리하기 위해서는 효율적인 데이터 처리 메커니즘이 필요하다. 따라서 본 논문에서는 저수준 상황 정보와 고수준 상황 정보까지 포함하며 그 데이터들을 효율적으로 처리하는 액티비티 매니저 모듈을 제안한다. 제안하는 모듈은 안드로이드 기반 컨텍스트 매니저 프레임워크 안에서 동작하며 다른 프레임워크들에 접근이 쉬워 다양한 용도로 확장이 용이하다. 이를 이용하여 사용자간 친밀도 모델을 제안하였으며 액티비티 매니저에 의한 사용자 친밀도를 평가하였다. 따라서 상황 인지 플랫폼을 이용한 서비스 제공자나 개발자에게 다양한 서비스 개발 환경을 제공해 줄 수 있다.

Acknowledgement

본 연구는 BK21 2단계 사업의 지원을 받아 연구되었음.

참고문헌

[1] J. Hakkila, et al., "Context-Aware Mobile Media and Social Networks," Proceedings of the International Conference on HCI, Article No.108, 2009.

[2] M. Raento, et al., "ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications," Proceedings of IEEE Pervasive Computing, Vol. 4, No. 2, pp. 51-59, 2005.

[3] A. Beach et al, "WhozThat? Evolving an Ecosystem for Context-Aware Mobile Social Networks," Proceedings of IEEE Network, Vol.22, Issue 4, pp. 50-55, 2008.

[4] B. L. Andersen, et al., "iSocialize: Investigating Awareness Cues for a Mobile Social Awareness Application," Proceedings of OZCHI, Vol. 206, pp. 7-14, 2006.

[5] A. Ankolekar, et al., "Friendlee: A Mobile Application for Your Social Life," Proceedings of International Conference on HCI with Mobile Devices and Services, Sep. 15-18, 2009.

[6] K. Sorathia, and A. Joshi, "My World - Social Networking through Mobile Computing and Context Aware Application," Communications in Computer and Information Science, Vol. 53, pp. 179-188, 2009.

[7] 김바울, 김경덕, 김상욱, "안드로이드 기반 상황 인지 플랫폼," 정보처리학회 춘계학술발표대회 Vol. 17, No.1, pp. 222-225, 2010.