

# PDA용 워크플로우 관리 시스템 클라이언트 구성 방안

조원선, 장우혁<sup>0</sup>, 한동수, 김중배  
한국정보통신대학원대학교, 한국전자통신연구원  
{wsjo22,torajim, dshan}@icu.ac.kr, jjkim@etri.re.kr

## Supporting mobile client for WFMS using PDA

Wonsun Jo, Woohyuk Jang<sup>0</sup>, Dongsu Han, Jungbai Kim  
Information and Communications University, ETRI

### 요 약

모바일 디바이스들의 강력한 이동성 및 즉시성은 전통적 워크플로우 관리 시스템과의 통합을 통하여 더욱 큰 생산성 향상을 가져 올 수 있다. 본 연구에서는 PDA를 실제 런타임 클라이언트로 가정하여, 디스커넥티드 오퍼레이션과 몇가지 모듈의 추가 및 프로세스 정의어의 변경을 통하여 모바일 디바이스의 통합을 시도하였다. 이를 위해서 기존의 워크플로우 시스템 서버와 클라이언트에 모바일 환경을 지원하기 위해 필요한 추가적인 모듈이 새롭게 고안되고 설계되었다.

### 1. 서 론

워크플로우 관리 시스템은 다양한 단계에 걸쳐 있는 인적 혹은 IT적 자원을 적절하게 배치 혹은 관리함으로써 비즈니스 프로세스의 절차적 자동화를 지원하고, 이로 인한 제품 생산 능력의 향상을 지원 하는 소프트웨어 시스템이다[1]. 전통적인 워크플로우 관리 시스템의 클라이언트는 데스크탑 컴퓨터나 터미널을 이용하여 왔다. 하지만, PDA와 같은 모바일 디바이스가 대중화 됨에 따라, 모바일 오피스라는 개념이 알려지고 있는 지금, 워크플로우 시스템에 기반을 두고 있는 어플리케이션에 이를 통합함으로써, 좀더 단축된 프로세스 주기, 높은 사용자 이동성, 사무실 밖의 작업과 같은 장점을 기대 할 수 있게 되었다. 이에, 모바일 디바이스 자체의 여러 약점들을 파악하고, 그에 따른 올바른 클라이언트 구성 방안에 대한 연구가 절실한 상황이다.

워크플로우 시스템에 이동성을 주기 위한 시도는 주로 디스커넥티드 오퍼레이션의 지원에 관한 연구로 대표되어 왔다. Bussler[2]는 사용자에게 이동성을 주기 위한 일반적인 요구사항들을 테드라인, 작업 배정, 어플리케이션과 데이터 관점에서 정의하였다. 여기서는 단절된 상태의 클라이언트의 작업 테드라인이 파기되면, 그 작업을 다른 클라이언트에게 재할당 하거나 테드라인을 늘려야 한다. 작업 배정에서는 중복된 할당이 금지되며, 어플리케이션과 그 어플리케이션 수행에 필요한 데이터들은 클라이언트에 존재하여야 한다.

Exotica/FMDC[3] 역시 디스커넥티드 오퍼레이션을 지원하는 워크플로우 시스템이다. 이 시스템에서는 세가

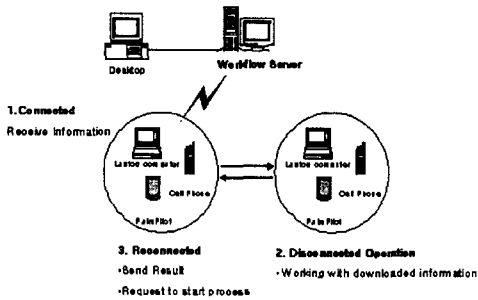
지 단계를 제시하였는데, 단절 전, 단절, 재연결 단계이다.

단절 전 사용자는 워크리스트에서 작업을 선택하고, 선택한 작업의 잠금과 관련 자료의 다운로드가 일어난다. 재 연결시에는 완료된 작업 결과가 서버쪽으로 보내지고, 중앙 데이터 베이스에 저장되게 된다.

디스커넥티드 오퍼레이션의 특징에 관한 최근의 연구에서는 작업의 타입과 어플리케이션의 타입, 수행자 타입, 데이터에 따라 그 특성을 규명해 보고자 하는 시도가 있었다[4]. 여기서는 단절이 가능하기 위해서는 작업의 타입이 반드시 수동이어야 하고, 어플리케이션이 네트워크를 통하여 분산되면 안된다. 또한 수행자는 사람이나 역할, 또는 조직이어야 하며, 데이터는 고립된 클라이언트에서만 작성되어야 한다.

그간의 연구에서 사용자의 이동성(mobility)을 제공하였던 시스템들은 크게 두 가지로 분류될 수 있는데 첫번째는 클라이언트의 실제적 이동성(디바이스의 공간적 이동성 제공)을 제공하는 방식이고 다른 경우는 HTTP를 사용하여 어디에서든 브라우저(browser)로 접근할 수 있게 한다는 제한적 이동성(웹브라우저로 접근 가능)을 제공하는 방식이다. 전자의 경우 클라이언트 디바이스를 랩탑과 같이 충분한 리소스를 보유한 기기로 가정하였기 때문에 PDA나 핸드폰과 같은 제한적 리소스를 가진 디바이스들에까지 확장하기는 힘든 상황이고, 후자의 경우는 어플리케이션 방식 대신 애플릿 혹은 웹프로그래밍을 사용한 것 이외에는 실질적인 사용자의 이동성과는 거리가 있다고 할 수 있다.

본 연구에서는 PDA나 휴대폰과 같은 Thin client를 워크플로우 시스템의 클라이언트로 사용해 보고자 하였다. 현재 모바일 디바이스의 성능은 계속 향상되고 있기 때문에, 워크플로우 시스템과의 연동은 랩탑에 비해 더욱 좋은 이동성과 인기로 인하여 생산력의 향상을 가져오기 때문이다. 그러나, 이런 모바일 디바이스들은 열악한 네트워크 환경이나 리소스의 제약 등으로 인하여 연결이 때때로 끊어질 수 있다는 약점이 있다. 따라서 단절된 상황에서도 주어진 작업을 수행할 수 있는 디스커넥티드 오퍼레이션이 필요하게 된다. 본 연구에서 단절은 사용자에 의한 자발적인 것이며 워크플로우 시스템의 작업이 사용자의 뜻에 따라 시작되고 끝낼 수 있다고 가정하고 있다.



[그림 1] 디스커넥티드 오퍼레이션의 세가지 단계

2. 시스템 설계

2.1 디스커넥티드 오퍼레이션

디스커넥티드 오퍼레이션의 제공의 목적은 사용자에게 디바이스의 연결 상태에 상관없이 일할 수 있는 환경을 제공하자는 데에 있다. 만약 사용자가 지하실이나 비행기와 같은 연결이 힘든 지역으로 이동할 경우 사용자는 단절 상태에 대비하여야 한다. 이것은 곧 사용자가 시스템에게 곧 단절이 있을 것이라는 것을 알려 주어 시스템에서 단절 상태를 준비하도록 한다는 것을 의미한다. 시스템은 디스커넥티드 오퍼레이션에 필요한 정보를 모두 제공하고 사용자는 서버의 도움 없이 작업을 수행할 수 있다. 이후 연결이 가능해 지면 사용자는 작업 결과를 서버에 업로드 하여 워크플로우 프로세스가 계속 진행되도록 한다. 이것은 사용자가 연결이 단절되었다더라도 워크플로우 프로세스가 계속 진행될 수 있도록 지원해 준다.

2.2 요구 분석

앞서 살펴본, 디스커넥티드 오퍼레이션의 정의에 따라 사용자 관점에서 시스템의 요구사항을 살펴보면 다음과 같다.

- 사용자는 연결을 단절하려는 의도를 서버에게 알린다.
- 디스커넥티드 오퍼레이션을 위한 정보는 단절 전에 서버로부터 클라이언트에게 다운로드 된다.
- 단절 상태에서 사용자는 클라이언트만을 이용하여 작업을 수행한다.

- 작업 결과는 서버로 되돌려 진다.
- 프로그램(워크플로우 파트와 어플리케이션)은 클라이언트 측에서 실행되며, 모바일 장치의 리소스 안에서 수행이 가능하도록 충분히 작은 크기로 유지되어야 한다.
- 위의 분석된 요구사항을 기반으로 단계별 시나리오와 요구사항을 살펴보면 다음과 같다.

[단절 전(Prior to disconnection)]

사용자 시나리오	시스템의 요구사항	위치
로그인	사용자 인증	S/C
작업 리스트의 검색	작업 리스트 제공	S
다운로드 요청	요청 받은 정보의 제공 작업 상태를 실행 상태로 전환	S
저장	다운 받은 정보를 저장	C

[단절(Disconnection)]

사용자 시나리오	시스템의 요구사항	위치
로그인	사용자 인증 작업(필요할 경우)	C
작업 리스트 검색	로컬 작업 리스트 제공	C
선택/시작/재시작	어플리케이션 호출	C
어플리케이션 실행	로컬 데이터를 이용한 어플리케이션 실행	C
대기	어플리케이션 실행 대기	C
실행 완료	작업 결과 저장 작업의 상태 변경	C
제거	중단된 작업 기록 제거	C

[재연결(Reconnection)]

사용자 시나리오	시스템의 요구사항	위치
로그인	사용자 인증	S/C
업로드 요청	작업 결과 업로드	S
	작업 상태 변화 또는 기록 제거	S
	수정/동기화 작업 상태를 중단 상태로 변경	S

2.3 구성

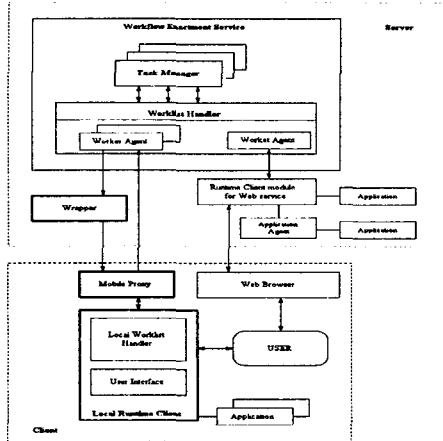
요구사항에 따라 몇 개의 모듈이 기존의 연구실 워크플로우 시스템 서버와 클라이언트 모듈에 추가되었다. 이들의 종류와 역할은 다음과 같다.

■ 클라이언트 모듈

- 로컬 런타임 클라이언트: Local Worklist Handler의 기능을 가지고 있음. 본래의 Worklist Handler의 기능과는 달리 특정 클라이언트의 작업 리스트만을 관리한다.
- 모바일 프록시: 이 모듈을 통하여 로컬 런타임 클라이언트로부터 사용자 요구를 wkAgent에게 전달하게 된다. 서버와 클라이언트 통신이 있을 경우 이를 중개하는 역할을 한다. 모바일 디바이스의 열악한 구조를 보조하여 실제 클라이언트 객체의 인스턴스가 위치한다. 클라이언트 위치에 존재하지만, 실제 PDA위에 올라가 있는 것은 아니며, 사용자 개개의 작업용 데스크탑에 존재한다.

■ 서버 모듈

□ Wrapper: 어플리케이션에 사용될 데이터는 Wrapper에 의해 변경된다. 어플리케이션에 사용될 데이터 혹은 작업 아이템에 필요한 데이터는 모바일 디바이스에 알맞도록 변형(추출) 되어야 하며, 이는 서버쪽에 존재하면서, 클라이언트와의 통신이 최소한의 사이즈로 이루어지도록 한다.



[그림 2] ICU/COWS와 새롭게 추가된 모듈

2.4 빌드 타임의 고려

프로세스 정의 시, 프로세스 디자이너에 의하여 프로세스 템플릿이나 정의가 분석된다. 이 때, 단절 가능한 작업과 수행자가 표현 되어야 한다. 앞선 연구를 수용하여 [4][5], 이것은 빌드타임에 외부적인 방법으로 표시하기로 한다. 단절 가능한 작업은 시작이나 끝을 사용자가 결정하여야 하는 수동 작업이다. 이러한 정보는 빌드 타임에 WPDL(Workflow Process Definition Language)의 확장 어트리뷰트를 사용하여 표기한다. 이를 위하여 프로세스 정의 틀을 수정하여 다음과 같은 실제 프로세스 정의가 생성되도록 변경하였다.

```
//Workflow Application List
APPLICATION 'Delivery Pickup'
NAME "Delivery_Pickup"
TOOLNAME "Del_P"
IN_PARAMETERS "ADDRESS"
EXTENDED_ATTRIBUTE "app_type" STRING "MOBILE"
EXTENDED_ATTRIBUTE 'app_data1' 'address'
```

서버에서는 위와 같이 확장된 WPDL의 내용을 인식하고, 모바일 클라이언트의 단절 요청이 있을 경우, 해당 필드를 통하여, 단절 작업 가능여부를 판단한다.

2.5 단절 상태에서 클라이언트의 오퍼레이션

사용자는 로컬에 존재하는 작업 리스트를 통하여 작업을 선택하고 모바일 프록시를 통하여 wkAgent에게 작업에 관련된 정보를 다운받고 단절을 하게 된다. 여기서 소개되는 내용은 관련 정보의 다운을 통하여 작업을 실행시키는데 필요한 정보가 로컬에 존재할 경우이다. 따

라서 모든 작업이 로컬 클라이언트에서 이루어지게 된다. □ 사용자는 살펴볼 작업 리스트를 선택한다. 작업 리스트 메뉴는 작업 아이템의 상태에 따라 구분된다.

□ 이 작업 리스트는 저장된 작업 아이템 정보에서 만들어진다.

□ 사용자는 'ready' 상태의 작업 리스트로부터 작업 아이템을 골라 시작시킬 수 있다. 로컬 작업 리스트 핸들러는 작업 아이템 정보를 읽고 해당하는 어플리케이션을 호출한다.

□ 사용자는 작업 결과를 완료 명령과 함께 서버로 보낼 수 있으며, 지연 업로드를 위하여 로컬에 저장할 수도 있다. 작업 아이템의 상태는 'complete'로 변경된다. 결과가 서버쪽으로 확실하게 전달되면, 상태는 'terminated'로 변경된다

3. 결론 및 향후 과제

우리는 본 연구에서 OMG의 j-flow모델에 부합하는 ICU/COWS에 PDA를 접목함으로써 모바일 디바이스와 기존 워크플로우 관리 시스템과의 통합을 시도하였다. 현재 본연구의 내용은 Wireless LAN을 기반으로 구현되었으며 앞으로는 CDMA2000망 환경에서 동작이 가능하도록 발전시킬 예정이다. CDMA2000망은 현재와 같이 클라이언트의 request없는 push방법 자체가 불가능하며, pop과 같이 주기적으로 서버로부터 데이터를 가져오는 방법이 필요할 것으로 예상된다.

4. 참고 문헌

[1] Workflow Management Coalition, "The Workflow Reference Model," WfMC TC-00-1003, January 19, 1995  
 [2] Christoph Bussler, "User Mobility in Workflow-Management-Systems," Proceedings of the Telecommunications Information Networking Conference, vol. 37, no. 1 pp.81-89, February 1995.  
 [3] G. Alonso, R. Gunthor, M. Kamath, D. Agrawan, A. El Abbadi, and C. Mohan, "Exotica/FMDC: Handling Disconnected Clients in a Workflow Management System," Proc. 3rd Int'l Conf on Cooperative Information System, Vienna, May 1995.  
 [4] Seung Il Lee, Dongsoo Han, and Dongman Lee, "Supporting voluntary disconnection in WfMSs," Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications, pp. 132-139, 2001.  
 [5] Mijin Hwang, "Handling Disconnected Clients in Workflow Systems," Master Thesis, School of Engineering, Information and Communication University, 2002.