

유비쿼터스 기반의 건강진단 자동화 시스템의 설계

공상환^{1*}

The Design of Automated System for Ubiquitous Healthcare Examination

Sang-Hwan Kung^{1*}

요 약 비즈니스 오퍼레이션의 핵심기능인 워크플로우는 비즈니스 프로세스(process)에서 공통적으로 나타나는 업무 절차의 핵심적인 요소와 이 요소들 간의 관계를 모델화한 것이다. 본 연구에서는 최근 우리 주변에서 쉽게 접하게 되는 건강진단 프로세스에서, 어떻게 워크플로우를 활용할 수 있을 가하는 방법에 초점을 맞추고 있다. 특히, RFID(Radio Frequency Identification) 태그가 부착된 PDA(Personal Digital Assistance)를 피검사자에게 제공하여 건강진단 프로세스의 완벽한 자동화를 실현하도록 하고 있다.

이를 위하여, 논문에서는 유비쿼터스 환경에서의 건강진단 프로세스를 실현하기 위한 소프트웨어 아키텍처의 설계와 아울러, 클래스의 구조 설계, 주요 메시지의 정의, 그리고 검사간 라우팅 알고리즘 등 구체적인 설계내용을 제안하고 있다. 다시말해, 건강진단 응용의 환경 및 서비스 시나리오에 대한 분석을 토대로, 자바와 같은 객체지향언어 환경에서 워크플로우 엔진과 건강진단 응용에 대한 소프트웨어 설계를 제안하고 있다. 본 연구의 결과는 도우미 중심적인 건강진단 프로세스의 무인화를 가능하게 하고, 나아가 많은 건강진단 센터들이 유비쿼터스 기반의 건강진단 프로세스를 현실화할 수 있을 것으로 사료된다.

Abstract The workflow, one of core features for modern business operation, models common elements and their relationships in business processes. The research is focused on how to adapt the workflow model to the healthcare examination which becomes popular these days. Especially, we intend to have fully automated process for healthcare examination by providing customers with PDA with RFID chip.

In order to realize this goal, this paper proposes the design of software architectures the class structures, and the definition of messages used for the system interactions. In other word, the paper shows the analysis of healthcare application in terms of service scenario and the design of workflow engine and ubiquitous healthcare application under the Object-Oriented programming environment. The result of the study may also enable the unmanned healthcare examination as well as ubiquitous based healthcare examination sooner or later.

Key Words : 워크플로우, RFID, 건강진단

1. 서론

어떤 사람들은 워크플로우를 제조공정의 조립과정에 비유한다. 각 공정에서 작업자는 구성품을 검사하고 기능을 시험하면서 작업이 마쳐질 때까지 각 단계의 작업을 수행한다. 워크플로우는 이와 같이 여러 단계의 작업과정을 요구하는 환경에서 효율성을 추구하는 중요한 성능달성의 도구기 된다. 따라서 워크플로우를 이용하면, 비즈니스에서 발생하는 다양한 작업은 각각의 공정에서 막힘

없이 처리되고 자동적으로 순환되어, 높은 작업성취도를 얻게 해 준다[10]. 더우기 불필요하거나 중복된 비즈니스 프로세스들은 프로세스 개선과정(BPR)을 통해 점진적으로 효율적인 프로세스로 발전하게 된다[1].

본 논문에서는 이러한 워크플로우의 개념을 여러 가지 검사과정으로 이루어지는 건강진단 프로세스에 적용해 보고자 한다. 통상, 건강진단 환경에서는 동시에 여러 환자들이 검사를 받아야 하고, 따라서 특정한 검사를 위해 대기하여야 하는 경우가 많이 발생한다. 또한, 하나의 검사를 마친 뒤 다음 검사로 향하기 위해서는 도우미의 안내를 받아야만 한다.

¹백석대학교 정보통신학부

*교신저자: 공상환(kung@bu.ac.kr)

이 문제를 해결하기 위해 시도된 본 연구는 두 가지 관점에서 연구의 성과와 특징을 찾아 볼 수가 있다. 하나는 워크플로우를 이용하여 유비쿼터스 환경에서의 건강진단 프로세스를 실현하기 위한 소프트웨어 설계를 제안한 것이다. 구체적으로는, 건강진단을 위한 서비스 환경과 비즈니스 시나리오를 정의하고, 이를 토대로 소프트웨어 아키텍처와 세부적인 클래스 및 주요한 메시지를 정의하였다는 점이다.

또 하나는 실시간적으로 변화하는 동적 상황에 대해 워크플로우 시스템에 적용한 것을 들 수 있다. 예를 들어, 실시간의 필요성은 한 사람이 검사를 마친 후 다음 검사를 결정할 때, 동적인 의사결정이 이루어 지기 때문에 중요하다 할 수 있다. 논문에서는 이를 위해 각각의 검사를 위해 대기하는 사람의 수와 각각의 검사에서의 소요 시간을 토대로 다음 장소로 이동하는 알고리즘을 동시에 제안하고 있다.

2. 관련 연구

2.1 워크플로우의 개념과 표준

프로세스의 자동화를 위한 모델이라고 할 수 있는 워크플로우는 프로세스나, 액티비티(activity), 전이조건(transition condition), 역할(role)들을 중심으로 자동화가 가능한 공통적인 요소와 기능을 정의한다[13]. 비즈니스 프로세스는 워크플로우에 의해서 표현된다. 이 워크플로우는 업무나 처리에 포함된 많은 액티비티들로 구성되는데, 이 액티비티는 워크플로우 자체나 작업(task)에 대한 추상적인 용어이다. 이 개념은 워크플로우는 복수의 작업이나 또는 복수의 부속 워크플로우로 구성된다는 것을 의미한다. 또한 각 작업은 역할(role)에 의해 처리된다. 즉, 역할은 작업항목을 처리하는 에이전트라고 할 수 있다[6].

기본적으로 워크플로우의 핵심요소는 3R(Routes, Rules, Roles)과 3P(Processes, Policies, Practices)로 표현되는 데, 이러한 요소는 어떠한 응용에서 활용되느냐에 따라 달라진다[7]. 또한, 워크플로우의 응용은 정보관리 행위의 구조를 분류하기 위한 관점에서 프로세스, 케이스(case), 폴더(folder), 규칙/응용(role/application), 그리고 문서(document)로 구성되는 5개의 계층적 구조로 분류하기도 한다[12].

이러한 워크플로우는 응용의 특성에 따라 3가지 모델에 의해 운영된다. 즉, 전자우편을 이용하는 방식과 문서기반의 워크플로우, 그리고 프로세스 중심적 워크플로우

이다[12].

워크플로우의 경우 워크플로우 업체들을 중심으로 결성된 WfMC(Workflow Management Coalition)에서 발표한 표준이 통용되고 있다[13]. 또한, OMG(Object Management Group)에서는 객체기술기반의 워크플로우를 위한 JFlow라는 표준을 제정 중에 있다.

2.2 RFID 기술 및 응용

유비쿼터스 응용은 사람 또는 임의의 객체에 대한 위치나 상태를 감시할 수 있도록 해 주는 RFID나 USN(Ubiquitous Sensor Network) 등의 기술로 실현되고 있다. 센서 네트워크 기술은 무선통신을 이용하여 원격의 상황을 측정하고 감시하는 데 유용한 반면, RFID 기술은 제품에 태그의 형태로 부착되어, 제품의 위치 파악이나 추적, 계수에 용이하게 활용된다[5]. 이러한 기술과 그리고 관련된 상품들은 물류나 환경감시 등 다양한 분야에서 활용되고 있으며, 유비쿼터스 환경은 다른 분야에서도 급격한 확산이 진행되고 있다[11]. 무엇보다 유비쿼터스 환경의 실현은 국가나 범세계적인 표준에 의해 뒷받침되어야만 실질적인 환경에서의 활용이 가능하므로, 관련 기술 및 표준의 개발활동이 EPCglobal이나 Auto-ID 센터, 그리고 전 세계 선진국들을 중심으로 활발히 진행되고 있다[2][3][8][9].

3. 응용 환경 및 서비스 시나리오

3.1 응용 환경

최근 들어, 건강에 대한 관심이 높아지면서 건강의 적신호를 조기에 파악하기 위해 많은 사람들이 건강진단을 정기적으로 받고 있는 추세이다. 건강진단은 통상 각종 의료용 시험장비가 잘 구비된 전용의 건강진단센터에서 시행된다. 이러한 센터에서 수행하는 검사는 우리가 설계하고자 하는 워크플로우에서 작업단위(activity)로 인식된다. 건강진단에서의 작업단위 유형에는 상담 및 접수외에 신체계측, 청력검사, 심전도검사, 시력검사, 혈압검사, 폐기능검사, 혈액검사, 종합진단 등이 있다.

다음의 그림 1의 건강진단 환경을 보면, 각각의 검사장소에서 검사자는 검사를 받기 위해 대기하는 사람들을 확인하거나 검사결과를 입력하기 위해 PC를 사용한다. 특히, 검사를 받는 사람들은 도우미 없이도 다음에 받아야 할 검사장소를 확인하거나 이제까지 받은 검사내역을 확인하기 위해 PDA를 활용한다. 또한, 이 PDA에는 그림 2에서 보여 주는 바와 같이, 뒷면에 RFID 칩이 부착되어

있다. 이것은 피검사자가 검사장소에 도착하면 별도의 동작없이도 자동적으로 검사장소에 설치된 리더기가 피검사자 PDA의 RFID 태그를 인식하도록 고려한 것이다. 이 RFID에 부착된 피검사자의 정보는 각각의 검사장소에 설치된 RFID 리더기에 의해 읽혀지며, 이 정보는 즉시 서버의 데이터베이스로 저장됨과 동시에, 검사자의 PC에도 나타난다.

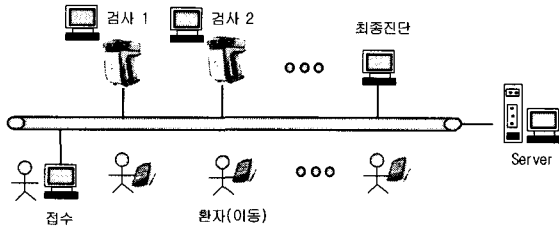


그림 1. 건강진단 환경

하나의 검사가 종료되면, 검사결과는 검사자에 의해 데이터베이스로 등록된다. 검사에 따라서는 검사결과를 얻는데 며칠이 소요되는 경우도 있다. 그러나 본 연구에서 다루는 범위는 건강진단센터에서의 실시간적인 상황이므로, 검사의 결과를 얻는 과정을 시스템에서 다루지는 않는다.

일단 검사결과가 입력되면, 시스템은 피검사자가 이동해야 할 다음의 검사장소를 결정하게 된다. 이때는 사전에 정의된 검사순서나 특정한 검사를 위한 소요시간, 그리고 각 검사에서 대기 중인 피검사자의 인원을 고려한다.

이러한 고려사항을 토대로 하여 이동해야 할 검사장소가 결정되면 피검사자의 PDA 화면에는 이 검사장소의 위치가 표시된다. 이러한 정보는 피검사자가 별도의 도우미가 없이도 스스로 검사장소로 이동할 수 있도록 해주는 것이다. 필요하다면, 피검사자는 이제까지 자신이 검사한 항목이나 미검사항목을 PDA를 이용하여 확인해볼 수도 있다.

4. 프로세스와 액티비티의 분석

하나의 워크플로우는 동시에 수행되는 복수의 프로세스를 통해 실행된다. 또한, 각 프로세스는 다시 병렬적으로 또는 순서적으로 수행되는 복수의 액티비티를 포함한다. 설계 시스템에서 하나의 액티비티는 건강진단의 가장 기본적인 작업인 검사가 될 것이다.

다음의 그림 2는 본 예에서 액티비티를 프로세스의 관점과 검사자 즉, 역할(Role)의 관점에서 살펴본 것이다.

즉, 하나의 프로세스에는 복수의 액티비티가 있으며, 프로세스에 포함된 각각의 액티비티는 서로 다른 액티비티가 된다.

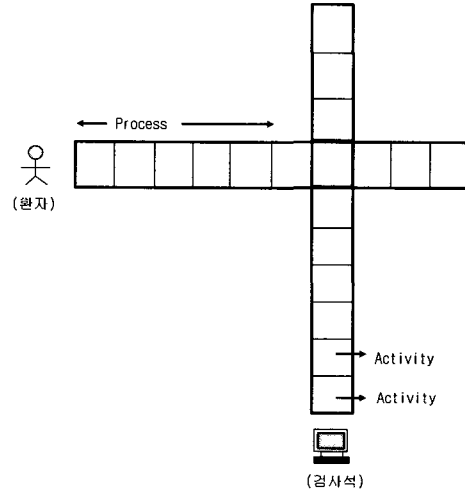


그림 2. 프로세스와 액티비티의 관계

액티비티를 수행하는 수행자는 검사자가 되며, 특정한 환자는 이러한 액티비티들을 순서에 따라 하나씩 거쳐 가게 된다. 한 환자가 일련의 과정을 거치게 되면 하나의 액티비티 집합을 완료하게 되는 데, 이것이 곧 프로세스가 된다. 다음의 그림 3은 시작과 종료를 포함한 A - F의 액티비티로 구성되는 프로세스의 한 예를 보여 준다. 프로세스를 정의한다는 의미는 건강진단이라고 할지라도 병원마다 차이가 있을 수 있기 때문에, 프로세스의 정의 내용에 따라 워크플로우 엔진이 수행된다면 하나의 워크플로우 엔진으로 여러 가지 각기 다른 검사절차를 갖는 병원의 검사업무에 쉽게 활용이 가능하게 된다.

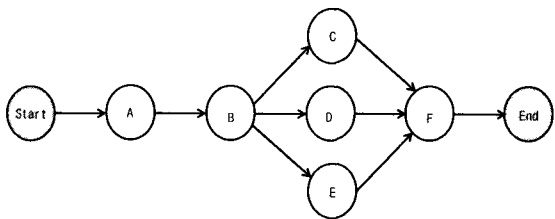


그림 3. 프로세스 정의사례

표 1은 프로세스 정의내용을 표의 형태로 표현한 것이다. 테이블에는 두 개의 인접한 액티비티 간의 관계를 설명하고 있으며, 아울러 시작 액티비티와 종료 액티비티 사이의 라우팅 방법도 표시한다. 예를 들어, 'F'는 지정 라우팅(Fixed Routing)을 의미하며, 'R'은 임의 라우팅

(*Random Routing*)을 의미한다. 지정 라우팅이란 검사순서가 미리 XML 스크립트에 정의되어, 하나의 검사에서 다음검사로 전이될 때 정의된 순서를 기반으로 다음검사장소를 결정하는 방식이다. 한편, 임의 라우팅이란 검사장소의 결정을 검사시간이나 대기자를 고려하여 동적으로 결정하는 방식이다. 테이블에서 역할이란 의미는 해당 검사를 전담하는 검사자의 역할을 의미한다. 즉, 특정한 역할의 검사자는 담당하는 검사를 전담해서 수행한다는 것을 의미한다.

표 1. 액티비티 테이블

| From-Activity | To-Activity | Routing Type | Role |
|---------------|-------------|--------------|------|
| String | String | char | int |
| Start | A | F | 0 |
| A | B | F | 1 |
| B | C | R | 2 |
| B | D | R | 3 |
| B | E | R | 4 |
| C | F | F | 5 |
| D | F | F | 6 |
| E | F | F | 7 |
| F | End | F | 8 |

Activity Type
f : Fixed Routing
r : Random Routing

Role
0: Register
1: Test 1
2: Test 2
:
:

서 다음에 실시할 검사를 찾아야만 하는 때, 만약 다음에 실시한 검사가 이전에 실시한 검사와 선후관계가 있으면, 미실시한 검사 큐의 다음 순서에 있는 검사를 새로운 검사로 결정하면 된다. 그렇지만, 임의 라우팅의 경우에는 다음에 있을 수 있는 여러 검사 중 하나를 선정하게 된다. 각각의 검사에서 소요되는 평균지체시간과 현재 대기 중인 피검사자들의 수를 곱하여 총 지체시간을 산출하고, 모든 가능한 검사들의 총 지체시간을 비교하여 가장 적은 지체시간의 검사장소를 다음에 이동할 검사장소로 리턴시켜 준다.

4. 건강진단 시스템의 설계

4.1 시스템의 구조

시스템은 크게 클라이언트 측과 서버 측의 두 가지 부분으로 구성된다. 서버 측에는 워크플로우 서비스를 제공하는 워크플로우 엔진 부분과 데이터베이스 처리와 관련된 건강진단 응용을 포함되어 있다. 한편, 클라이언트 측에는 사용자의 입력과 출력을 도와주는 건강진단 응용이 위치하게 된다.

이러한 구조를 더 세분화 시킨 블록 다이어그램은 그림 5와 같다. 여기에는 중요한 구성모듈과 이들 간의 상호작용을 표현한다. 구체화된 블록은 크게 4가지 부분에 포함되는 데, 클라이언트 부분과 워크플로우 서버 부분, 워크플로우 응용 부분, 그리고 워크플로우 서비스 부분이다. 클라이언트 부분은 다시 장치 핸들러 블록과 워크플로우 응용블록으로 구성된다. 클라이언트가 만약 서비스를 요청하게 되면, 우선 서버 측의 워크플로우 서버에 접

한편, 검사를 위한 라우팅의 경우에 특정한 검사장소에서 다음 장소로 이전할 때 사용되는 알고리즘을 요약하면 다음의 그림 4와 같다. 우선, 특정한 피검사자의 입장에서 볼 때, 모든 검사는 두 가지 중의 하나에 포함된다. 즉, 실시한 검사와 미실시한 검사이다. 이들은 각각의 큐에 저장된다. 특정한 피검사자가 하나의 검사를 마치게 되면, 이것은 그 검사를 미실시한 검사 큐에서 실시한 검사 큐로 이전시키는 이벤트가 된다. 이 시점은 또한 그 피검사자의 다음의 검사장소를 결정하게 하는 순간이 되기도 한다. 다음의 검사장소는 물론 미실시한 검사 중에

```

if(routing type of the test == Random) {
    search from the undone test queue the undone tests which have not not finished yet.
    if(number of undone test which is linked to this test is one)
        return the undone test
    else { // multiple undone tests are linked to this test
        compute the totaol waiting time of each of undone test by multipling the number of
            waiting testees by average waiting time of the test.
        choose the undone test with least total waiting time.
        return the undone test.
    }
}
else { // routing type == Fixed
    find the next test from the pre-defined path
    return next test
}
    
```

그림 4. route() 함수의 처리 알고리즘

표 2. 주요 클래스의 역할

| 클래스 명칭 | 클래스의 역할 |
|----------------|---|
| WfServer | <ul style="list-style-type: none"> - Workflow를 사용하는 클라이언트의 응용에 workflow 서비스를 제공하기 위한 서버측의 프로세스이다. - Workflow를 구성하는 각각의 프로세스를 관리할 ProcessManager를 기동시킨다. - 응용에 따른 프로세스의 구성은 사전에 XML 형태로 파일에 정의되는 데, 이 XML 파일을 읽어 들여 Process를 구성하는 액티비티의 연관관계를 테이블에 수록한다. - 클라이언트가 각 PC로부터 workflow 서비스를 요청하면 WfServer는 서버측에서 이 클라이언트에 대응하여 서비스를 대행할 ClientService 객체를 독립적인 쓰레드로 기동시켜 준다. |
| Client | <ul style="list-style-type: none"> - Workflow 서비스를 이용하는 PC 측의 응용 프로그램이며, 응용에 종속된 클라이언트가 작성된다. - 서버측의 ClientService와 1:1로 대응하면서 메시지를 주고 받게 된다. |
| ClientService | <ul style="list-style-type: none"> - Client의 Workflow 엔진을 이용하기 위한 정의된 서비스(함수)의 호출을 대행하여 엔진을 구동하는 클래스이다. - 아울러, Client 응용이 필요로 하는 서버측의 데이터베이스 저장이나 검색 등도 대행하여 주는 역할을 수행한다. - 프로세스가 진행되면서 특정한 클라이언트에게 전달되어야 할 액티비티가 발생하면 이것을 통보해 달라고 요청하기 위한 EventSubscriber 인터페이스를 구현한다. |
| Connection | <ul style="list-style-type: none"> - Client가 ClientService 클래스와 정보를 교환하는 데 사용되는 공통된 클래스이다. |
| WfMessage | <ul style="list-style-type: none"> - Client와 ClientService 간 교환되어야 할 다양한 정보를 나타내는 클래스이다. |
| ProcessManager | <ul style="list-style-type: none"> - 복수의 프로세스들을 관리하는 역할을 수행한다. - ClientService의 요청에 의하여 프로세스를 생성하며, 프로세스에 포함된 각각의 액티비티는 미리 Process 모델을 토대로 프로세스가 진행되면서 자동 또는 수동으로 만들어 진다. - 프로세스가 진행되면서 새로운 액티비티로 전환된 경우 ClientService에게 통보해 주기 위하여, 이 이벤트를 미리 등록하고, 이벤트 발생시 중재를 하기 위한 함수를 정의하는 EventBroker 인터페이스를 구현한다. |
| Process | <ul style="list-style-type: none"> - 개별 프로세스를 대표하는 클래스이다. 예를 들어, 건강검진에서 하나의 환자가 검진을 시작하면 하나의 프로세스가 시작된다고 할 수 있다. - 하나의 프로세스는 복수의 액티비티로 구성되며, 하나의 액티비티에서 다음 액티비티로의 전환은 사전에 정의된 XML을 기초로 하지만, 처리 상황에 따라 다음에 수행될 액티비티가 달라질 수 있다. - 하나의 프로세스 내에서의 액티비티가 자동전환될 때, 이를 ProcessManager를 통해 Client에게 알려 주기 위하여 RouteHandler를 구현한다. |
| Activity | <ul style="list-style-type: none"> - 한 환자의 하나의 검진이 하나의 액티비티가 되며, 이를 위한 클래스이다. - 상태를 가지며, 하나의 액티비티가 종료되면 다음 액티비티가 자동 또는 수동으로 전환되며, 이를 위해 RouteHandler를 구현한 Process를 호출하여 route를 수행한다. |
| RoleModel | <ul style="list-style-type: none"> - 특정한 검사자는 ID에 대응하는 역할(role)을 가지는 데, 이 역할은 하나 이상의 액티비티를 포함한다. - 역할과 액티비티의 대응관계는 XML 파일에 정의되는 데, RoleModel은 이 파일로부터 참조된 역할-액티비티 관계를 관리한다. |
| ActivityList | <ul style="list-style-type: none"> - 특정 Role에 대응된 액티비티를 담고 있는 테이블이다. |
| Worklist | <ul style="list-style-type: none"> - 특정 역할 즉, 검사자가 담당해야 할 작업을 포함하는 테이블이다. |
| WorkItem | <ul style="list-style-type: none"> - 검사자의 하나의 작업을 나타내는 요소이며, 프로세스 ID와 액티비티 ID로 구성된다. |

설계된 건강진단 워크플로우 시스템에서의 주요 클래스들의 역할을 설명하면 다음 표2와 같다.

4.4 메시지와 이벤트

클라이언트와 워크플로우 서버 사이에서 교환되는 정보는 다음 표 3과 같다. 메시지는 크게 클라이언트에서 서버로 가는 메시지와 서버에서 클라이언트로 향하는 메

시지로 구분된다. 검사자 로그인이나 접수석에서의 환자 접수, 그리고 검사자의 작업목록을 확인하기 위한 메시지는 클라이언트의 요청 메시지가 된다. 한편, 새로운 피검사자의 도착을 알려주는 작업항목 메시지는 서버로부터 클라이언트에게 전달되는 동적 메시지가 된다. 또한, 검사결과를 입력하는 검진내역이나 건강진단의 마지막 절차로 수행되는 의사의 최종진단 메시지는 클라이언트로부터 서버로 전달되는 메시지가 된다.

표 3. 워크플로우 메시지의 유형

| 메시지 | 메시지 방향 | 메시지 내용 | 비고 |
|----------------------------|-------------|--------------|-----------------------------|
| LOGIN (검사자 로그인) | 클라이언트 -> 서버 | ID, Password | ID는 역할도 변환되고, 역할은 액티비티와 매핑됨 |
| PATIENT_RECEIPT (환자 접수) | 클라이언트 -> 서버 | 환자정보 | 환자정보 DB에 수록 |
| GET_WORK_LIST (작업목록 요청) | 클라이언트 -> 서버 | - | 검사자가 처리해야 할 모든 검진대상 내역 요청 |
| WORK_ITEM (작업항목) | 서버 -> 클라이언트 | 새로운 검진대상 정보 | 처리해야 될 하나의 작업항목 |
| TEST (검진내역) | 클라이언트 -> 서버 | 검진결과 정보 | 개별 검사별로 별도의 메시지가 있음 |
| : | | | |
| TEST_END (최종 진단) | 클라이언트 -> 서버 | 진단 정보 | 검진의 최종 단계에 해당 |

5. 결론 및 향후 연구계획

본 논문은 건강진단을 위한 응용과 미들웨어인 워크플로우 서버의 설계에 대하여 제안하고 있다. 건강진단센터에서의 검사과정은 비즈니스 프로세스의 자동화를 위한 워크플로우의 응용으로 적합하며, 특히 실시간 라우팅은 워크플로우를 DB 중심적으로 구현하기 보다는 소프트웨어 엔진으로 구현하도록 해야 하는 당위성을 제공한다.

논문의 초점은 이러한 서비스를 가능하게 하는 워크플로우 모델의 설계와 소프트웨어의 구조의 설계에 맞추어져 있다. 이를 위해, 비즈니스 시나리오와 핵심적인 요구사항의 분석과 아울러 블록구조의 설계 및 상세 클래스 구조를 제시하고 있으며, 검사장소 이동시 임의 라우팅을 위한 간단한 알고리즘도 제안하고 있다.

본 연구를 통한 성과라고 하면, 최근 많이 활용되고 있는 건강진단센터에서의 건강진단 절차가 많은 피검사자들이 폭주하게 될 때 효율적인 검사의 분배가 어려운 환경에서 PDA와 RFID 기능을 이용함으로써, 전체 프로세스의 자동화의 실현은 물론 효율적인 동적 라우팅을 통한 서비스의 개선에도 큰 기여를 한다는 점을 들 수 있다. 아울러, 보다 유비쿼터스 기술을 보다 현실적인 응용에 활용함으로써 유비쿼터스 기술의 발전과 실현을 앞당기는 데도 기여가 될 수 있다고 하겠다.

참고문헌

[1] 도근우, BPR에 의한 업무혁신, 산업훈련연구소, 1995.

[2] Auto-ID Center, Auto-ID Servant Specification 1.0, 2003.
 [3] Auto-ID Center, PML Core Specification 1.0, 2003.
 [4] Alec Sharp & Patrick McDermott, Workflow Modeling: Tools for Process Improvement and Application Development, Artech House, 2001.
 [5] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, "Wireless Sensor Networks for Habit Monitoring," ACM Sensor Networks and Applications, 2002.
 [6] Andrezej Cichocki, Abdelsalam Helal, and Marek Rusinkiewicz, Darrell Woelk, Workflow Process Automation, Kluwer Academic Publishers, 1998.
 [7] David Coleman and Raman Khanna, Groupware: Technologies and Applications, Prentice Hall PTR, 1995.
 [8] EPCglobal, EPC Information Services(EPCIS) Version 1.0 Specification, Candidate Specification, 2004.
 [9] EPCglobal, The Application Level Events(ALE) Specification, Version 1.0, Candidate Specification, 2004.
 [10] Gary Poysick and Steve Hannaford, Workflow Reengineering, Adobe Press, 1996.
 [11] Roy Want, "Enabling Ubiquitous Sensing with RFID," Computer, Vol. 37, 2004.
 [12] Thomas M. Koulopoulos, The Workflow Imperative, John Wiley & Sons, Inc., 1995.
 [13] WfMC, The Workflow Reference Model, DN TC00-1003, 1995.

궁 상 환(Sang-Hwan Kung)

[정회원]



- 1977년 숭실대학교 전자계산학과 졸업(이학사)
- 1983년 고려대학교 대학원 전자정보차리학과 졸업(경영학석사)
- 1998년 충북대학교 대학원 전자계산학과 졸업(이학박사)
- 20001년 ~ 현재 백석대학교 정보통신학과 부교수

<관심분야>

소프트웨어 구조, 분산시스템