
분산 네트워크 환경하에서 TMO를 이용한 중환자 생체정보 원격 진료 시스템 구현

김광준*

The Implementation of Patient Vital Sign Information Telemedicine System using TMO in Distributed Network Environment

Gwang-Jun Kim*

요 약

본 논문에서는 TMO 실시간 객체 모델을 이용하여 실시간 통신 메시지 서비스를 효과적으로 지원하기 위해 중환자의 실시간 생체정보 전송 시스템을 나타내었다. 또한 분산된 네트워크 시스템에서 TMO 구조를 이용하여 ICU(intensive care unit) 응용 환경에 적용함으로써 실시간 메시지 서비스를 보장하였다. TMO를 이용한 생체정보 파싱 수신 모듈은 ICU의 Central Monitor로부터 수신할 수 있는 Raw Data 형태의 환자의 생체 정보를 생체 정보 웹 뷰어 시스템에서 사용가능한 데이터로 분석할 수 있도록 설계하였다. 실시간 뷰어 시스템은 환자 생체정보에 대한 실시간성과 생체 정보에 대한 데이터의 연속성을 부여하여 베드 사이트의 환자에게서 발생된 모든 생체정보에 의해 환자 관리를 데이터베이스를 이용하여 전산화함으로써 환자의 광범위한 자료 검색이 이루어지므로 원격진료로 활용이 가능하다. TMO 객체 모델을 기반으로 한 실시간 시뮬레이션에서 몇 가지의 TMO 구조의 장점을 가지고 있으며, TMO 객체 모델은 요구 명세서와 설계 사이의 강력한 연관성을 가지고 있다.

ABSTRACT

In this paper, we present a patient real-time vital sign information transmission system to effectively support developing real-time communication service by using a real-time object model named TMO (Time-Triggered Message-Triggered Object). Also, we describe the application environment as the ICU(Intensive Care Unit) to guarantee real-time service message with TMO structure in distributed network systems. We have to design to obtain useful vital sign information, which is generated at parsing data receiver modulator of HIS with TMO structure, that is offered by the central monitor. Vital sign informations of central monitor is composed of the raw data of several bedside patient monitors. We are willing to maintain vital sign information of real time and continuity that is generated from the bedside patient monitor. It is able to apply to remote medical examination and treatment. we proposed integration method between a vital sign database systems and hospital information systems. In the real time simulation techniques based on TMO object modeling, We have observed several advantages to the TMO structuring scheme. TMO object modeling has a strong traceability between requirement specification and design.

키워드

TMO, ICU(Intensive Care Unit), Vital Sign Information, Parsing Module, Telemedicine Viewer System

I. 서 론

실시간 시스템은 작업 수행 결과의 정확도 및 작업의 수행이 시간적인 제한을 가지고 있는 시스템으로써 경성 실시간 시스템(Hard real-time system)과 연성 실시간 시스템(Soft real-time system)으로 분류할 수 있다. 경성 실시간 시스템은 원자력 발전소 제어 시스템, 민감한 의학기기 시스템 그리고 군사 방어 체계 시스템 등과 같이 이벤트 처리의 정확성과 시간적인 제한성을 지켜주지 않으면 엄청난 불상사를 불러일으킬 수가 있어서 이벤트 처리 서비스에 대한 예측성과 보장성이 주어져야 하는 시스템이다. 연성 실시간 시스템은 경성 실시간 시스템과 같이 작업의 수행 결과 및 시간의 정확성은 필요로 하지만 시간 조건 불만족 시에도 서비스가 허용되거나 복구 절차가 주어질 수 있는 시스템을 말한다. 즉 경성 실시간 시스템은 예측성(Predictability)과 설계 시 수행시간에 대한 보장 서비스(Timeliness guaranteed service)가 최대한 고려되어야 하는 시스템이고, 연성 실시간 시스템은 비동기적 사건에 대한 빠른 병행 처리와 임계시간 초과 등의 결함 발생시의 복구 절차를 제공하는 것을 목표로 하는 시스템이다[1,2].

실시간 객체 모델의 대표적인 예로는 Kane Kim에 의해 제안된 TMO 모델(Time-triggered Message-triggered Object Model)로서 객체 모델을 실시간 시스템의 모델링에 적합하도록 확장한 것으로써 시간에 의해 구동되는 시간구동(Time-Triggered) 메소드와 메시지에 의해 구동되는 메시지 구동(Message-Triggered) 메소드를 제공한다[1-6].

TMO 모델에서 제공되는 시간 구동 메소드와 메시지 구동 메소드에 의해 실시간 시스템이 갖는 시간적인 특성과 이벤트를 쉽게 추상화 할 수 있는 구조를 가지고 있을 뿐만 아니라, 적시 서비스 능력(timely service capability)을 설계 단계에서부터 보장할 수 있다.

본 논문에서는 이러한 TMO 구조를 이용하여 실시간 통신 환경을 구축하기 위해 ICU(Intensive Care Unit)의 중환자 모니터 원격진료 시스템 응용 환경에 적용하였다. Central Monitor로부터 전송되어진 환자의 생체정보 Raw Data가 HIS의 TMO 파싱 데이터 수신 모듈을 통해 사용가능한 데이터로 재구성될 수 있도록 설계가 이루어져 있다. ICU의 환자 생체정보에 대한 실

시간성과 데이터의 연속성을 부여함으로써 베드 사이의 중환자에게서 발생된 모든 생체정보에 의해 환자 관리가 가능하다. 또한 보다 많은 검사장비의 데이터를 데이터베이스를 이용하여 전산화함으로써 해당 환자의 광범위한 자료 검색이 이루어지므로 획기적인 원격진료가 가능하다. 적절한 생체정보 데이터의 교환과 정보 관리의 수정을 통해 병원정보 시스템은 모든 병원 관계자의 보다 나은 업무처리를 향상시킬 수 있다.

II. TMO 구조

TMO는 Time-triggered Message-triggered Object의 약자로서 Kane Kim 등에 의해서 개발된 Object Structuring Scheme이다[1,2,3] TMO는 기존의 객체 모델을 경성 실시간 시스템에서 높은 효율성을 보일 수 있는 객체 모델로 확장하기 위한 연구에서 나온 결과이다. 따라서 TMO는 실시간 시스템이 가지는 시간적인 특성과 행동을 쉽게 추상화 할 수 있는 구조를 가지고 있을 뿐만 아니라 적시 서비스 능력(timely service capability)을 시스템 설계 단계에서부터 보장할 수 있다. TMO의 객체 모델의 구조는 다음 4개의 부분으로 구성된다.

- Object Data Store(ODS): 실시간 데이터를 저장하기 위한 부분으로 Object Data Store Segment (ODSS)단위로 관리된다. ODSS는 TMO 메소드인 SpM(Spontaneous Method)과 SvM(Service Method)에 의해서 상호 배타적으로 접근 가능하고, MVD (Maximum Validity Duration)이 지나면 무효한 데이터가 된다.
- Environment Access Capability(EAC) : 외부와의 논리적인 통신 채널, 그리고 I/O 디바이스 인터페이스 등에 대한 연결 통로이다.
- Spontaneous method (SpM): 주기성을 띠거나 시간성을 갖는 메소드이다.
- Service Method (SvM) : 기존의 객체 모델이 가지고 있는 메소드 그룹과 같은 형태로써 클라이언트로부터 온 메시지에 의해 분산 TMO 객체의 서비스 호출을 위해서 제공되는 메소드이다.

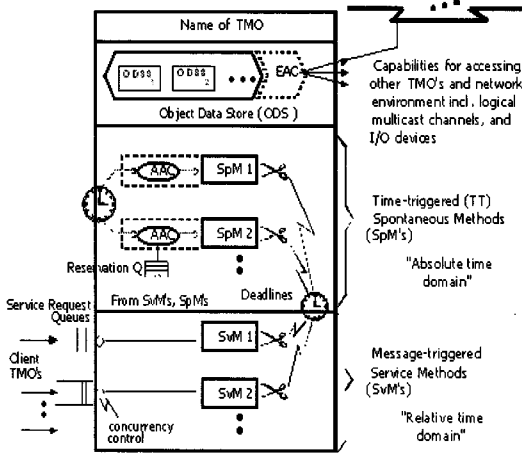


그림 1. TMO 기본 구조
Fig. 1 Basic Structure of TMO

그림 1은 TMO 객체 모델의 기본적인 구조를 나타내고 있으며, 기존의 객체 모델과는 다른 다음과 같은 특징을 가지고 있다.

(1) 분산 컴퓨팅 컴포넌트

TMO 모델의 설계 개념 중 가장 두드러진 특징은 RTCS(Real-Time Computing System)로서 항상 TMO들로 구성된 분산 실시간 네트워크의 형태를 취하는 것이다. TMO들은 서버에 있는 서비스 메소드에 대한 클라이언트 호출을 통해서 서로 상호작용을 한다. 분산된 멀티노드의 객체들은 non-blocking 형태의 원격 메소드 호출을 통하여 분산 처리를 수행한다.

(2) Spontaneous Method (SpM)

Time-triggered method인 SpM은 클라이언트의 서비스 요청에 의해서 실행되는 SvM과는 달리 TMO 설계 시에 명세한 시간이나 주기가 되면 실시간 클럭(clock)에 의해 자동적으로 실행되는 메소드이다. SpM의 시간 조건은 설계 시에 Autonomous Activation Condition(AAC)에 상수로 명세된다.

다음은 AAC 명세 방법에 대한 예시를 나타내고 있다.

“for t = from 10:00am to 10:50am
every 20min
start-during (t, t+5min)

finish-by t+10min

위 AAC 명세의 의미는 10:00am에서 10:50am 까지 20분을 주기로 활성화되어야 하고, 매 주기의 시작은 t에서 시작하여 t+5분 사이에 활성화 되어야 하며 t+10분 안에 완료된다는 의미를 가지고 있다.

(3) Basic concurrency constraint (BCC)

TMO들의 시간적인 서비스 능력을 보장하기 위한 제약 조건으로써, SpM과 SvM이 공유데이터 ODSS를 동시에 접근하려고 할 때 발생할 수 있는 충돌을 방지하기 위한 수행 규칙이다. 이때 SpM이 SvM 보다 더 높은 우선순위를 갖는다. 즉, 외부의 클라이언트로부터 온 메시지에 의해 수행되는 SvM의 실행은 기본적으로 SpM과 충돌이 없는 경우나, 충돌이 일어난 SpM의 실행이 끝난 후에만 가능하다. 정확히 말하자면 SpM과 SvM 사이에는 데이터를 공유하기 위한 ODSS(Object Data Store Segment)가 존재하는데, 이를 동시에 액세스 하려는 경우에 SpM이 SvM 보다 더 높은 우선순위를 가지는 것이다. 그러므로 객체의 수행되는 시간을 디자인 단계에서 고정시킬 수 있고, SpM의 수행은 SvM에 의해서 방해 받지 않으며, SpM의 수행 시 그 시간을 보장할 수 있는 것이다.

(4) 종료시간과 데드라인 보장

디자이너가 메소드의 시작시간, 종료시간 그리고 데드라인을 명세함으로써 시스템의 적시 서비스 능력(timely service capabilities)을 디자인 단계에서 보장할 수 있도록 지원 한다.

III. 객체 및 클라이언트 전송 호출

실시간 통신 객체 지향 분산 컴퓨팅 설계는 실시간 통신 시뮬레이터를 가지고 있는 각 객체 노드를 분산시킨 네트워크 형태로 구축한다. 실시간 통신 객체들은 서버 객체에서 서비스 메소드를 통해 클라이언트 객체를 호출함으로써 상호 동작한다. 호출자는 클라이언트의 시간 구동 메소드나 서비스 메소드가 될 수 있다. 클라이언트나 서버 객체의 동시성 극대화를 위해 호출 서비스 메소드는 블록킹(Blocking) 형태의 호출을 이용하거나 비

블록킹 형태의 호출을 이용한다. 그러므로 실시간 통신 시뮬레이션 프로그래밍 구조는 서버 TMO내의 서비스 메소드에 블록킹 호출과 비블록킹 호출의 두 가지 형태 전송 호출을 사용한다.

블록킹 호출은 클라이언트가 서비스 메소드를 호출한 후 서비스 메소드로부터 결과 메시지가 되돌아올 때까지 대기하는 것으로 표현식은 다음과 같다.

**Obj-name.SvM-name(parameter-1,parameter-2,
...,by deadline)**

클라이언트와 서버 객체는 두 개의 다른 처리 노드에서 상주할 수 있고 이러한 호출은 원격 호출 절차의 형태에서 구현된다. 서비스 메소드에 결과 메시지가 없다 하더라도 서버 메소드로부터 실행 완료 신호가 특정 데드라인 시간까지 도착하지 않을 수도 있으며, 클라이언트 객체에 대한 실행 엔진은 산술적인 오버플로우(overflow)가 발생할 때 적절한 예외 핸들링(handling) 함수를 발생한다.

비블록킹 전송호출은 클라이언트가 서비스 메소드를 호출한 후 순차적으로 처리되고 서비스 메소드로부터 결과 메시지를 기다리는 것으로서 다음과 같이 나타낸다.

**Obj-name.SvM-name(parameter-1,parameter-2,...,mode
NWFR, Timestamp TS);**
-----statements-----;
get-result Obj-name.SvM-name(TS) by deadline;

위의 표현에 나타난 NWFR(No Wait For Return)은 비블록킹 호출에서 결과가 돌아올 때까지 기다리지 않는다는 의미이다. 클라이언트가 서비스 메소드를 호출할 때 클라이언트는 TS (Time Stamp)라는 변수에 시간 스탬프를 기록한다. 이러한 클라이언트로부터 시간 스탬프는 서비스 메소드와 관련된 호출의 형태를 구분한다. 그러므로 클라이언트가 초기의 비블록킹 호출로부터 되돌아온 서비스 메소드 결과를 실행하고 변수 TS에 서비스 메소드 이름뿐만 아니라 TMO 이름을 기록한다. 시간 스탬프는 클라이언트로 되돌아온 결과 구문을 실행하기 이전에 여러 개의 비블록킹 호출 함수가 호출될 때 비블록킹 호출을 하는 실행 엔진을 구분하여 서비스 메소

드를 실행한다. 클라이언트는 서비스 요구의 결과가 실행 시간 내에 되돌아오지 않는다면 서버 실행 엔진의 결과가 도착할 때까지 기다린다. 하나의 비블록킹 호출은 클라이언트의 서비스 메소드와 서버의 서비스 메소드 사이에 동시에 실행이 가능하도록 하고, 서비스 메소드의 요구에 부합된 실행 결과가 나올 때까지 동시성이 계속해서 지속된다. 어떤 상황에서는 클라이언트가 비블록킹 호출에 대한 서비스 메소드의 결과를 필요로 하지 않을 경우에 클라이언트는 결과 구문을 사용하지 않는다.

TMO 구조를 가지고 있는 여러 개의 클라이언트 객체가 분산되어 있는 시스템에서 임의의 TMO내의 서비스 메소드가 다른 TMO내의 서비스 메소드로 서비스를 요구하기 위해 클라이언트 전송 호출 함수를 이용한다. 마지막 TMO내의 서비스 메소드가 서비스를 요구한 최초의 클라이언트에 결과 값을 되돌려줄 때까지 계속해서 클라이언트 전송 호출 함수를 이용한다. 동시성 제약으로부터 클라이언트 전송 호출 함수를 사용하는 경우에 시간윈도우 크기를 충분히 크게 함으로써 서비스 메시지 메소드의 실행과 시간 구동 메소드의 실행사이에 발생하는 충돌을 방지할 수 있다. 하나의 서비스 메소드가 복잡하게 실행되는 경우에 시간윈도우의 크기를 충분히 크게 설정하지 않게 되면 서비스 메소드의 실행 엔진이 실행되지 못하는 경우가 발생한다.

이러한 경우 발생하는 문제를 해결하기 위해서 복잡한 서비스 메소드를 여러 개의 서비스 메소드로 분할하여 실행 시간 내에 서비스 메소드를 호출함으로써 시간윈도우 내에서 실행이 가능하게 할 수 있으며, 또한 분할된 서비스 메소드의 전송 오버헤드(overhead)를 줄임으로써 요구되는 서비스의 결과를 얻을 수 있다. 각각의 순차적인 통신 전송 오버헤드의 감소로 인해 클라이언트의 첫 번째 서비스 메소드의 호출이 이루어진 후 분산된 또 다른 실시간 통신 시뮬레이터의 서비스 메소드의 순차적인 호출을 통해 계속해서 서비스 메시지를 전송할 수 있다. 마지막 서비스 메소드가 클라이언트에 돌아올 때까지 결과를 계속해서 되돌려 준다. 클라이언트 시뮬레이터 객체에 하나의 서비스 메소드에 대한 전송 호출을 실행함으로써 실행 엔진에서 수행되고 있는 서비스 메소드 호출을 종료할 수 있다. 호출한 서비스 메소드에 요구된 서비스의 결과를 큐에 저장함으로써 서비스 메

소드에 대한 호출을 완료할 수 있다. 호출한 서비스 메소드에 결과를 나타내는 상태가 되돌아올 때까지 계속해서 결과를 되돌려준다.

실시간 통신 프로그래밍 설계자는 외부의 클라이언트가 첫 번째 서비스 메소드를 호출함으로써 서비스 메소드에 대한 결과를 예측할 수 있어야 하며 어디로부터 결과 메시지가 오는지를 알 수 있게끔 구현해야 한다. 하나의 클라이언트에 의해 서비스 메소드가 호출되는 경우에 클라이언트 전송 호출은 통과된 클라이언트 변수와 객체 저장 영역의 데이터를 공유함으로써 정보를 통과시킨다. 각각의 클라이언트 내의 서비스 메시지 메소드에 대한 전송 호출 프로그래밍 표현은 아래와 같다.

ClientTransferCall(SvM-name, parameters)

서비스 메시지 메소드의 이름은 호출되어지는 서비스 메소드로 구별하며, 호출된 서비스 메소드 클라이언트 포트(port)의 ID 또는 채널은 서비스 요구에 대한 결과를 수신하고, 객체 실행 엔진에 의해서만 수행되고 있는 서비스 메소드에 메시지를 통과시킨다. 즉 호출된 서비스 메소드와 현재 클라이언트의 서비스 메소드 사이에 적절한 회귀 경로 연결 설정이 이루어져 있어야 한다. 변수들은 호출된 서비스 메소드에 의해 만들어진 최신의 변수를 포함하면서 클라이언트 전송 호출에 의해 순차적으로 변수를 넘겨준다. 이러한 클라이언트 전송 호출은 또 다른 실시간 통신 시뮬레이터를 가지고 있는 객체 노드에서 서비스 메소드를 호출하는 경우에 확장할 수 있다.

외부 객체 노드의 서비스 메소드에 대한 클라이언트 전송 호출 프로그래밍은 다음과 같다.

**ClientTransferCall
(Obj-name.SvM-name, parameters)**

클라이언트는 호출된 서비스 메소드로부터 되돌아온 결과나 또 다른 객체 노드의 서비스 메소드로부터 돌아온 결과를 구별할 필요가 없다. 실제적으로 클라이언트는 호출된 서비스 메소드로부터 돌아온 결과를 수용하는 것이 일반적이며, 실시간 통신 시뮬레이터 시스템

에서 임의의 서비스 메소드로부터 돌아오는 결과를 수용하는 것은 특별한 경우이다.

**IV. TMO를 이용한 ICU 실시간
생체정보 시스템**

4.1 병원 ICU(Intensive Care Unit) 구조

병원 ICU(Intensive Care Unit)에서는 중환자의 상태에 대한 연속적이고 집중적인 모니터링의 기능을 제공하는 환자 모니터(Bedside Monitor)가 주요한 의료 장비로서 활용되고 있다. 환자 모니터 시스템에 의한 환자 상태의 모니터링은 중환자의 모니터링에 수반하는 인력과 노력 및 부담의 절감이라는 효과를 제공하며, 보다 신속하고 정확한 환자 상태의 파악을 바탕으로 의료진에 의한 적절한 대처를 가능하게 한다. 환자 모니터 시스템의 가장 기본적인 기능은 환자에 부착된 각종 센서들로부터 여러 종류의 생체 신호들을 수집, 처리, 분석, 도시 및 전송이다. 환자 모니터 시스템은 통상 환자당 한대씩 설치되며, 이렇게 설치된 여러 개의 환자 모니터 시스템으로부터 수집된 환자의 생체 정보는 그림 1과 같이 중앙 환자 모니터 시스템으로 전송되어 여러 명의 환자들에 대한 통합적인 환자 모니터링이 이루어지기도 한다[7].

그림 2에서 나타낸 환자 모니터링 시스템은 환자의 상태를 모니터링하기 위하여 심전도, 혈압, 호흡, 체온, 동맥산소 포화도, 정맥 산소 포화도, 심박출력, 호기 시 이산화탄소, 흡기 시 산소, 기타 가스 농도와 이산화탄소 및 산소 분압 등을 측정하는데 환자의 상태와 진료의 목적에 따라서 이러한 항목에서 몇 가지가 선택적으로 사용된다. 따라서 환자 모니터 시스템은 필요에 따라 그 구성을 달리할 수 있는 모듈형의 구조를 가지는 것이 바람직하다. 또한, 각종 신호의 단순한 수집 및 도시의 기능에서 한걸음 더 나아가 기기 자체가 이러한 신호를 처리, 분석하여 위급한 상황을 검출하거나 또는 그러한 상황을 예측하여 의료진에게 전달하는 경보의 기능이 필수적이다[7][8][9].

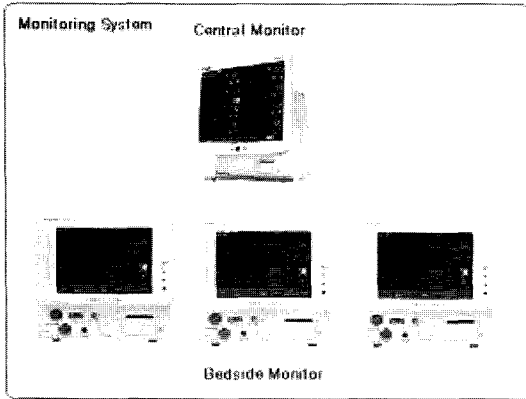


그림 2. 중환자 모니터링 시스템 구조
Fig. 2 Architecture of Patient Monitoring System

이를 위하여 환자 모니터링 시스템은 전산화가 필요하게 되며 전산화된 환자 모니터링 시스템에서는 여러 대의 환자 모니터를 전산망으로 연결하여 각종 데이터를 통합 관리하고, 여러 환자들을 중앙 집중 관리하는 등 다양한 부가 기능들을 구현하는 것이 필요하다.

4.2 분산된 네트워크 환경하에서 TMO 파싱 수신 모듈

분산된 네트워크 환경하에서 TMO를 이용한 중환자의 생체정보를 실시간적으로 전송하는 시스템을 구현하기 위해 TMO ICU 생체정보 파싱 수신모듈을 제안하였다. 또한 ICU의 Central Monitor에 나타난 생체정보를 TCP/IP 인터넷 프로토콜을 통해 HIS(Hospital Information System)에 구성된 스토리지에 저장하게 된다.

Central Monitor로부터 전송되어진 환자의 생체정보 Raw Data가 HIS(Hospital Information System)의 TMO 데이터 파싱 수신 모듈을 통해 각각의 환자 생체정보 Raw Data가 저장되며 장기간의 안정화 테스트가 필수적으로 고려되어야 한다. 이는 환자 생체정보에 대한 실시간성과 생체정보에 대한 생체정보 데이터의 연속성을 부여함으로써 베드 사이트의 환자에게서 발생된 모든 생체정보에 의해 환자 관리가 이루어져함을 의미한다.

먼저, ICU의 Central Monitor에서 전송되어지는 환자의 생체정보를 수신하기 위한 TMO 데이터 파싱 데이터 수신 모듈을 실행시킴으로서 생체 정보 데이터

의 수신을 활성화시키고, 서버의 소켓을 생성함으로써 그림3과 같은 환자의 생체정보 Raw Data의 수신이 이루어진다. 발생하는 환자의 생체정보 Raw Data의 정확한 수신과 수신된 정보를 스토리지에 저장하는 역할을 하게 되는 것이다. 수신된 환자의 생체정보 데이터는 클라이언트의 소켓을 통해 Raw Data를 획득할 수 있으며, 데이터 TMO 파싱 모듈을 이용하여 이미 정의된 메시지구조에 따른 파싱 과정을 통해 각 환자의 생체정보는 사용자의 요구와 환경에 따라 다각적인 구성을 통하여 제공이 가능하다. 실제 HIS의 스토리지에 저장되는 많은 양의 데이터를 실시간 처리하기 위하여 데이터 수신 모듈의 안정성과 신뢰성이 반드시 평가되어야 한다.

```

0030 f4 0b 41 00 00 00 f0 01 00 08 bc be 00 3e 0d 3a .ad.....>.:
0040 a0 36 20 00 00 14 20 20 20 20 20 20 20 20 20 20 .6 ...
0050 20 20 20 20 20 20 20 20 20 20 20 10 00 34 23 4b .....488
0060 21 3f 23 42 21 3f 23 53 21 21 21 21 21 21 21 21 !?88!788!!!!!!!
0070 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 !!!!!!!*
0080 2f 42 2f 53 20 20 20 20 20 20 20 20 20 20 20 20 /6/S
0090 20 20 20 20 00 28 42 45 44 24 30 30 34 20 30 30 .....(BED-004 00
00a0 30 30 30 30 30 00 00 32 30 30 37 30 35 32 32 0000..20070522
00b0 31 31 30 31 00 00 00 00 00 00 00 00 00 00 20 30 !101..... 0
00c0 00 3e 21 21 21 21 21 21 21 21 21 21 21 21 21 21 <!!!!!!
00d0 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 !!!!!!!
00e0 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 !!!!!!!
00f0 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 !!!!!!!
0100 00 00 f0 01 00 08 bc be 00 3e 0d 3a a0 36 10 00 .....>.6..
0110 00 08 0d f4 0b 41 00 00 00 3c 30 00 00 04 00 00 .....>.8.<0
0120 00 00 30 10 00 30 80 00 f0 00 00 30 00 78 00 28 .....>.0..<0..x.(
0130 00 c8 ff 38 00 c8 ff 38 03 07 00 05 00 03 00 14 .....>.8..0
0140 00 0a 00 00 00 02 07 00 a2 00 00 00 00 00 00 .....>.B.....
0150 00 00 00 00 00 30 50 00 14 c0 00 00 0e 00 1e .....>.0P.....
0160 00 05 00 0f 00 00 00 00 15 00 00 00 00 30 d0 .....>.0.....0
0170 00 28 00 00 00 00 00 00 30 00 00 b4 00 50 01 2c .....>.0...P..
0180 00 0a 01 2c 00 00 00 00 00 00 00 00 00 00 00 .....>.....
0190 00 00 ff ff 00 00 f0 01 00 08 bc be 00 3e 0d 3a .....>.:
01a0 a0 36 10 00 00 08 0d f4 0b 41 00 00 00 3c 40 00 .....>.6.....>.8.<e
01b0 00 fc 00 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
01c0 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
01d0 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
01e0 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
01f0 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0200 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0210 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0220 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0230 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0240 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0250 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....>.....
0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....>.....
0280 00 80 80 80 80 80 80 80 80 80 80 80 80 80 80 .....>.....
0290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....>.....
    
```

그림 3. ICU의 Central Monitor에 수신된 환자 생체정보 Raw Data
Fig. 3 Receiving Patient Vital Sign Information Raw Data of ICU Central Monitor

4.3 TMO 파싱 수신 모듈 라이브러리 설계

환자의 생체정보 Raw Data는 TMO의 파싱 모듈을 통해 사용 가능한 정보로써의 가치를 가진다. 환자의 생체

정보 Raw Data를 파싱하는 과정에서 가장 중요한 사항은 불규칙하게 수신되는 데이터를 메시지 단위로 구분하여야 하는 것이다. 이점을 해결하기 위하여 불규칙하게 수신되는 데이터의 처음 2byte가 메시지의 시작을 나타내는 태그 0xf001과 일치할때까지 임시 버퍼에 데이터를 추가하고 메시지의 시작 태그 0xf001과 일치하는 태그의 수신시에 임시 버퍼의 데이터를 메시지 단위로 구분하여 파싱하도록 하는 과정을 반복 처리하였다.

환자의 생체정보 수신 모듈에서 획득된 데이터는 여러 개의 메시지로 구성되어 있으며 각 메시지는 태그 2byte를 통해 구분되어진다. 결국, Raw Data의 파싱은 태그 2byte를 통해 구분된 메시지를 다시 세그먼트 단위의 의미 있는 데이터로써 사용될 수 있도록 하는 역할을 수행하는 것이다.

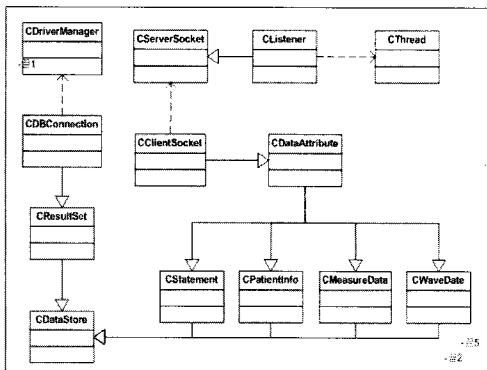


그림 4. TMO 파싱 라이브러리 모듈 클래스
Fig. 4 TMO Parsing Library Module Class

프로세스의 오류를 최소화 시켜야하는 것과 함께 메시지의 시작과 종료 시점을 정확하게 분석하는 작업은 파싱 모듈에서 가장 중요하므로 데이터 파싱에 앞서서 체크하여야 한다.

환자 정보를 파싱하는 모듈과 상태 정보를 파싱하는 모듈 그리고, 환자의 생체정보에서 계측 수치(HR, VPC, ST1, ST2) 및 파형 수치(ECG1, VP1, VP2, RESP, SpO2)를 파싱하는 모듈로 분기하여 파싱 과정을 그림4에서와 같이 수행하도록 되어있다. 분기하는 모듈은 파싱을 수행하기 위한 데이터를 담고 있는 버퍼와 길이를 인자로 받으며 Raw Data 파싱 모듈과 동일하게 태그와 데이터 길이를 분석하면서 파싱하는 방식으로 동작한다. 파싱 과

정을 완료하면 분기되었던 상태에서 복귀하며 유효한 환자일 경우에 스토리지에 저장하고 파싱 과정을 수행한 데이터의 길이를 증가시켜 Raw Data 모듈의 두 번째 인자인 데이터의 byte수와 비교하는 과정을 다시 반복을 수행한다.

그림 5는 TMO 구조를 이용하여 환자모니터 원격진료 시스템의 파싱 TMO를 나타낸 것으로서 DS-7600의 센트럴 모니터로부터 환자생체 정보를 획득한 Main-TMO의 SpM 영역에서 Parsing-TMO의 SvM 영역을 호출하여 전송한다. Parsing-TMO의 ODSS 영역은 환자의 정보를 나타내는 Patient Info Buffer, 환자 생체정보의 수치 데이터를 나타내는 수치데이터 버퍼, 생체정보의 파형을 나타내는 파형 데이터 버퍼로 객체 데이터 멤버로 이루어짐을 알 수 있다.

ICU Parsing- TMO
Access Capability (to other TMOs) None
Object Data Store Data Structure Patient Buffer Measure Buffer Wave Buffer
SpM Update the state descriptors in ODSS AAC1 Call SvM (Gate2-NonBlocking:Update Patient Buffer) Call SvM (Gate3-NonBlocking:Update Measure Buffer) Call SvM (Gate4-NonBlocking:Update Wave Buffer) Call Method-TMOSL(Deactivate_AAC:AAC1) or (PushSpMIdleMode)
SvM Raw Data Grouping Receiving PatientInfo From Main_TMO Receiving MeasureData From Main_TMO Receiving WaveData From Main_TMO Call method-TMOSL(Activate_AAC:AAC1) or (PullSpMFromIdleMode)

그림 5 환자 모니터 원격진료 시스템의 Parsing-TMO
Fig. 5. Parsing-TMO of Patient Monitor Telemedicine System

파싱 TMO ODSS 영역의 환자 정보, 수치 데이터버퍼, 파형 데이터 버퍼의 객체 데이터 멤버는 Main-TMO의 SpM을 통해 파싱 TMO의 SvM을 호출함으로써 DS-7600 센트럴 모니터의 생체정보 Raw 데이터를 전송하여

ODSS 객체 데이터 멤버를 주기적으로 갱신한다. 수신된 생체정보의 이진 정보 Raw 데이터를 파싱 수신 모듈을 통해 베드 사이트의 환자 모니터 생체정보를 DS-7600 센트럴 모니터로 송신한 정보를 응용환경에 맞게 설정하기 위해 파싱 수신 모듈 알고리즘을 적용한다.

TMO 구조를 이용한 환자 모니터 원격진료 시스템의 Parsing-TMO 실시간 프로그래밍의 주요 내용은 다음과 같이 나타낼 수 있다. 프로그래밍내에 나타낸 CParsingPro 클래스는 환자의 정보를 나타내는 Patient TMO, 환자의 생체정보를 수치데이터로 표현하는 수치 데이터 TMO 및 생체정보 파형을 나타내는 파형 데이터 TMO로 정의하였다. 이러한 TMO에 접근하기 위해서 파싱 TMO내의 SpM을 통해 환자 정보, 수치 데이터, 파형의 SvM을 호출함으로써 각각의 분산된 노드에서 TMO를 수행함으로써 센트럴 모니터로부터 송신된 환자의 생체정보에 대한 객체에 대해 주기적으로 계속해서 갱신된 데이터를 전송한다. Main-TMO로부터 송신된 생체정보의 Raw 데이터는 파싱 TMO의 SvM을 통해 ODSS 영역의 객체 데이터 멤버 스토리지에 저장한다.

다음 프로그램은 Parsing-TMO 모듈의 코딩을 나타내고 있다.

```
CParsingPro::CParsingPro
(TCHAR* TMO_external_name,
TCHAR* SvM_external_name,
AAC &aac_spec_patient,
AAC &aac_spec2_measure,
AAC &aac_spec3_wave,
TCHAR* PatientTMO_name,
TCHAR* PatientSvM_name,
TCHAR* MeasureTMO_name,
TCHAR* MeasureSvM_name,
TCHAR* WaveTMO_name,
TCHAR* WaveSvM_name,
tms gate_start_time,
tms TMO_start_time);

m_PatientGate(PatientTMO_name, PatientSvM_name,
gate_start_time),
m_MeasureGate(MeasureTMO_name, MeasureSvM_name,
gate_start_time),
```

```
m_WaveGate(WaveTMO_name, WaveSvM_name, gate_start_time)
{
SvM_RegistParam svm_spec;
svm_spec.GETB = 20 * 1000;
_tcscpy(svm_spec.name, SvM_external_name);

svm_spec.build_regist_info_ODSS(m_ParsingODSS.GetId(), RW); RegisterSvM((PFSvMBody)ParsingSvM, &svm_spec);
TMO_Sprintf(_T("<ParsingSvM_Regist Successfully>\n"));
TMO_RegistParam tmo_spec;
_tcscpy(tmo_spec.global_name, TMO_external_name);
tmo_spec.start_time = TMO_start_time;
RegisterTMO(&tmo_spec);
TMO_Sprintf(_T("<ParsingTMO_Regist Successfully>\n"));
}

BOOL CParsingPro::ParsingPatientSpM()
{
TMO_Sprintf(_T("<PatientSpM Started>\n"));
if (0 < m_ParsingODSS.m_wDataLen)
{
BOOL nResult = TRUE;
tmsp Timestamp;
ParamStruct_Parsing_SvM SvMPara;
memset(SvMPara.szBuffer, 0, 512);
memcpy(&SvMPara.szBuffer[0], m_ParsingODSS.m_szBuffer, m_ParsingODSS.m_wDataLen);
SvMPara.wDataLen = m_ParsingODSS.m_wDataLen;
SvMPara.tagType = m_ParsingODSS.m_tagType;
SvMPara.strFlag = m_ParsingODSS.m_strFlag;
SvMPara.strMacAddr = m_ParsingODSS.m_strMacAddr;
SvMPara.strSerialNo = m_ParsingODSS.m_strSerialNo;
SvMPara.strInitDate = m_ParsingODSS.m_strInitDate;
SvMPara.strInitTime = m_ParsingODSS.m_strInitTime;
SvMPara.strDataDate = m_ParsingODSS.m_strDataDate;
SvMPara.strDateTime = m_ParsingODSS.m_strDateTime;
```



```

SvMPara.strLanTime = m_ParsingODSS.m_strLanTime;
nResult = m_PatientGate.NonBlockingSR(&SvMPara,
sizeof(SvMPara), Timestamp);
if (nResult == TRUE)
TMOSSLprintf(_T("<PatientGate NonBlockingSR
Successfully>\n"));
else
TMOSSLprintf(_T("<PatientGate NonBlockingSR
Failure>\n"));
MicroSec system_age = GetCurrentDCSage();
deactivate_AAC (_T("aac_patient"),
tm4_DCS_age(system_age));
}
return TRUE;
}
    
```

V. ICU의 실시간 생체정보 뷰어 시스템

그림 6은 Patient Monitor(DS-7100)에서 생성된 환자의 생체정보를 ICU의 Central Patient Monitor(DS-7600)에서 TMO 과싱 수신 모듈을 이용해 실시간적으로 수신된 상태를 나타내고 있는 것으로서 수신된 생체정보를 HIS 서버 시스템으로 TCP/IP 인터넷 프로토콜을 통해 전송되며, 전송된 생체정보 데이터를 다시 TMO 구조를 적용한 웹 서버를 통해 Patient Monitor (DS-7100)에서 생성된 동일한 생체정보를 Client User에 디스플레이 하는 결과를 나타내고 있다.

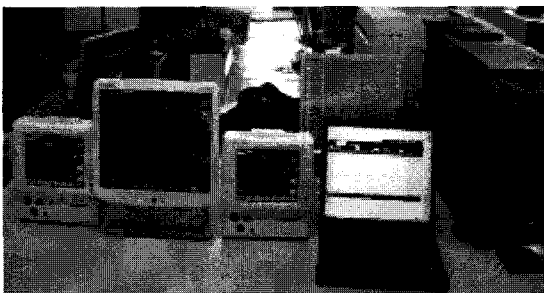


그림 6. TMO 생체정보 실시간 전송 테스트 베드 시스템
Fig. 6 TMO Vital Sign Information Real Time Transmission Test-Bed System

생체 정보 과싱 수신 모듈을 통해 사용자에게 제공할 수 있는 데이터로 구축된 데이터베이스에 저장된 환자의 생체정보는 생체 정보 뷰어 시스템을 통해 다시 확인할 수 있다.

그림 7은 Central Monitor에 등록된 환자의 정보를 검색할 수 있는 기능으로 “환자 정보” 메뉴에서 나타나는 화면으로 Central Monitor를 통해 획득된 데이터를 과싱 과정으로 구성된 화면이다. 제공되는 환자에 대한 정보는 환자의 성명, 베드 번호, 생년월일, 나이 등으로 ICU의 Central Monitor에 등록되지 않은 환자에 대한 정보는 그림7에서도 확인할 수 없다.

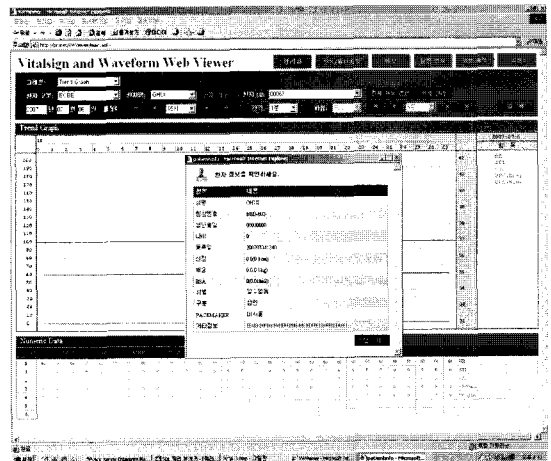


그림 7. 생체정보 뷰어 시스템 환자 정보
인터페이스 화면

Fig. 7 Patient information interface display of vital sign information viewer system

그림 8은 Central Monitor에서 환자의 위험 상태를 확인하기 위해 설정한 알람신호를 검색할 수 있는 기능으로 “알람 정보” 메뉴에서 나타나는 화면이다. 기본적으로 시시각각 변하는 환자의 상태를 체크하기 위한 알람의 ON/OFF 상태와 환자의 위험 상태를 감지하기 위한 알람의 상·하한치 설정을 확인할 수 있다. 또한, 알람 항목을 그룹으로 나누어 그림 8의 화면에서와 같이 상단에 7개의 탭으로 구성하였으며 선택된 탭에 해당하는 항목을 화면에서와 같이 확인할 수 있다.

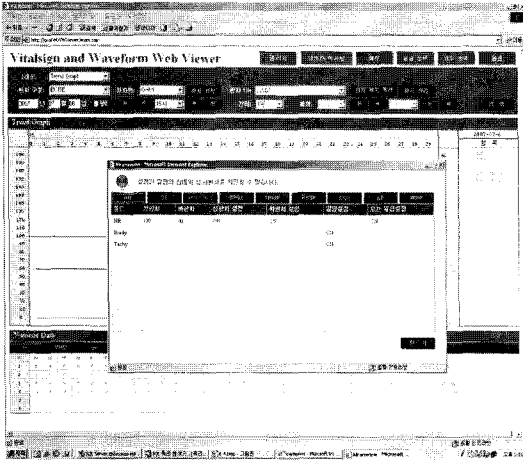


그림 8. 생체정보 뷰어 시스템 알람 정보 인터페이스 화면

Fig. 8 Alram information interface display of vital sign information viewer system

그림 9는 웹에서 의료진이 환자에 처지 및 투약사항을 간단히 입력할 수 있는 기능으로 “메모” 메뉴에서 나타나는 화면이다. 웹 뷰어 시스템은 Central Monitor에서 전송된 환자의 생체정보를 기반으로 검색이 가능한 시스템이지만, 이 기능은 환자에 투약이나 처지에 대해 입력할 수 있는 기능으로 입력된 내용과 함께 입력된 시간 까지도 확인 가능하다. 웹 뷰어 시스템의 메인화면에서 선택된 년-월-일에 해당하는 메모의 검색과 입력이 동시에 가능하다.

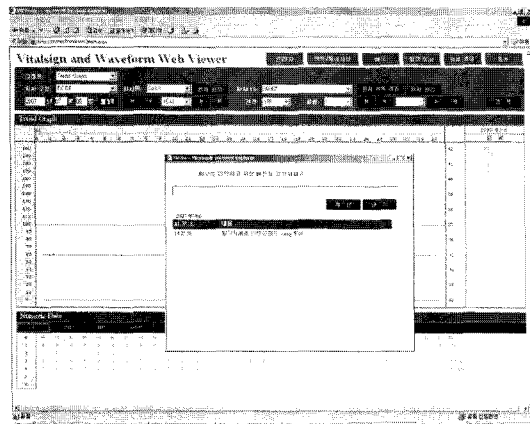


그림 9. 생체정보 웹뷰어 시스템 메모 인터페이스 화면
Fig. 9 Memo interface display of vital sign information web viewer system

그림 10은 “Trend Graph” 라는 기능의 선택으로 제공되는 화면으로 환자의 계측 수치가 본 화면에서 선 그래프와 그에 해당하는 수치에 대한 검색으로 가능하다. Central Monitor의 Mac주소로 분류되는 환자의 구분을 선택하고 검색 환자의 환자명이나 환자ID 선택으로 Trend Graph 검색이 가능하다. 추가적으로 검색을 하고자하는 날짜와 시간을 선택하거나 간격을 선택하면 사용자의 요구에 맞는 결과를 얻을 수 있도록 구성되어 있다. 특히, 간격은 1분 단위에서 5분, 10분, 30분, 60분등으로 검색을 선택할 수 있는 기능으로 60분 검색의 경우에는 한 화면에 해당 환자의 하루 동안의 계측수치를 그래프와 수치로 된다.

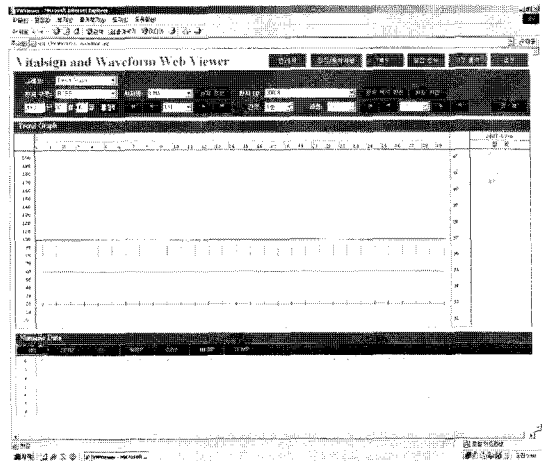


그림 10. 환자의 생체 계측 정보 데이터 GUI
Fig. 10. GUI of Patient Vital Sign Numeric Data

그림 11은 환자의 생체정보를 웹을 통해 실시간 검색할 수 있도록 구성된 웹 뷰어 시스템에서 “Wave Graph” 라는 기능의 선택으로 제공되는 화면으로 환자의 ECG1에 해당하는 파형 정보를 제공하기 위한 화면이다. Trend Graph와 마찬가지로, Central Monitor의 Mac주소로 분류되는 환자의 구분을 선택하고 검색 환자의 환자명이나 환자ID 선택으로 Wave Graph 검색이 가능하고, 검색을 하고자하는 날짜와 시간을 선택하면 사용자의 요구에 맞는 결과를 얻을 수 있도록 구성되어 있다. 특히, Wave Graph는 5분 단위로 검색하도록 구성되어 있고 화면에서 확장되어 있는 컨트롤은 검색을 시작하는 분을 선택하기 위함이다.

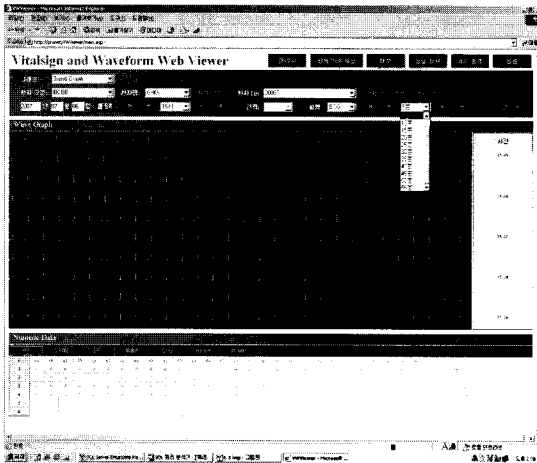


그림 11. 환자의 ECG1 생체정보 GUI
Fig. 11 GUI of Patient ECG1 Vital Sign Information

그림 12, 13, 14는 웹 뷰어 시스템에서 “Wave Graph”의 환자의 SpO2, Vp1, Vp2에 해당하는 각각의 파형 정보를 제공하기 위한 그래픽 사용자 인터페이스 결과를 나타내고 있다.

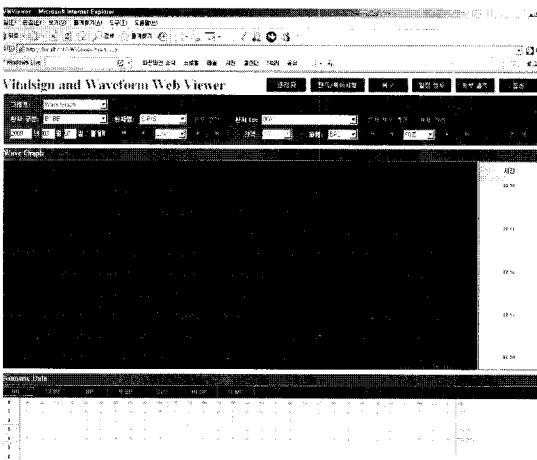


그림 12. 환자의 SpO2 생체정보 GUI
Fig. 12 GUI of Patient SpO2 Vital Sign Information

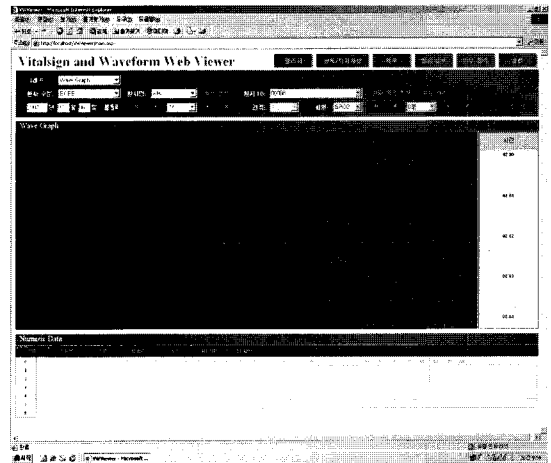


그림 13. 환자의 Vp1 생체정보 GUI
Fig. 13 GUI of Patient Vp1 Vital Sign Information

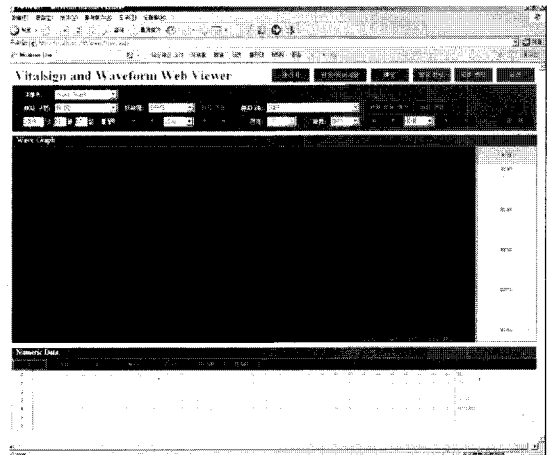


그림 14. 환자의 Vp2 생체정보 GUI
Fig. 14 GUI of Patient Vp2 Vital Sign Information

VI. 결론

기존의 실시간 객체 모델은 시스템 복잡도의 증가에 따라 결함이 발생할 가능성이 커지고, 실시간 시스템의 고 신뢰도를 보장하는 다양한 요구사항을 잘 반영할 수 있는 실시간 시스템 모델과 실시간 시스템 개발 기간을 단축시킬 수 있는 통합된 개발환경을 제공하기에는 부족한 면을 가지고 있다.

이러한 단점을 해결하기 위해 실시간 통신 객체 모델이 가지고 있는 기존의 실시간 통신 특성들을 포함하면서 시간 구동 메소드와 서비스 메시지 메소드 객체에 객체지향 프로그래밍 기법을 이용한 TMO 구조를 토대로 하나의 통합된 실시간 통신 환경을 구축할 수 있다.

TMO 구조를 이용하여 실시간 통신 환경을 구축하기 위해 ICU(Intensive Care Unit)의 중환자 모니터 원격진료 시스템 응용 환경에 응용하여 적용하였다. TMO 구조를 이용한 각각의 객체 노드는 실시간 통신 특성 요소인 시간 구동 메소드와 서비스 메시지 구동 서비스가 기본 제약 사항을 따르지 않고 동시성을 만족하면서 각각의 객체 노드에 유연하게 적용됨으로서 메시지의 실시간성을 확보하였으며, 또한 분산된 객체 노드의 시뮬레이터 클럭에 시간 구동 메소드를 적용함으로써 각각의 객체 노드에 부여된 과중한 부하를 줄임으로서 중환자 생체정보 메시지의 교환을 보장할 수 있다.

ICU의 Central Monitor로부터 전송되어진 환자의 생체정보 Raw Data가 HIS의 TMO 파싱 데이터 수신 모듈을 통해 사용가능한 데이터로 재구성될 수 있도록 설계하였으며, 중환자 생체정보에 대한 실시간성과 생체정보에 대한 생체정보 데이터의 연속성을 부여함으로써 베드 사이트의 환자에게서 발생된 모든 생체정보에 의해 환자 관리가 이루어진다.

TMO 생체정보 파싱 모듈을 통해 중환자 모니터 뷰어 시스템 구현은 환자 생체정보의 실시간적 모니터링을 통해 환자에 대한 생체정보 결과를 시간과 장소를 구분하지 않고 조회할 수 있으므로 해당 환자에 대한 빠른 조치로 응급 상황에도 신속, 정확하게 대응할 수 있다.

참고문헌

- [1] Kim, K.H., "Real-Time Object-Oriented Distributed Software and the TMO Scheme", Int'l Jour. of Software Engineering & Knowledge Engineering, Vol. No2, pp.251-276, April 1999.
- [2] K. H. Kim and J. Liu, "Deadline Handling in Real-Time Distributed Objects", Proc. ISORC 2000, Newport Beach, CA, pp.7-15, March 2000.
- [3] Kim, K.H., Ishida, M., and Liu, J., "An Efficient Middleware Architecture Supporting Time- Triggered Message-Triggered Objects and an NT-based Implementation," Proc. ISORC'99 (IEEE CS 2nd Int'l Symp. on Object-oriented Real-Time Distributed Computing), pp.54-63, May 1999.
- [4] 김광준, 서종주, 강기웅, 윤찬호, "분산 네트워크 시스템에서 TMO를 이용한 실시간 통신 시뮬레이션 구현", 한국해양정보통신학회 논문지, pp.897-905, Vol.11, No.5, 2007
- [5] 김광준, 서종주, 강기웅, 윤찬호, "다중 노드 시스템에서 TMO를 이용한 실시간 서비스 메시지 보장", 한국전자통신학회 논문지, pp.28-34, Vol.1, No.1, Oct. 2006
- [6] Kim GJ, Kim CS, Kim YG, Yoon CH and Kim MH, "Real-Time Communication using TMO in Distributed Computing Systems", International Journal of KIMICS, Vol. 1, No. 1, March 2003
- [7] L. A. Geddes and L. E. Baker, "Principles of Applied Biomedical Instrumentation", New York, John Wiley & Sons, 1989
- [8] 한국보건산업진흥원, HL7Korea, "Health Level Seven (HL7)과 개발도구", 2002
- [9] HL7 Modeling & Methodology Committee, "HDF Methodology Specification chapter1-7, HDF Reformatted Core Chapter", La Verne, Health Level Seven, Inc. pp.22-56, 2004
- [10] Fukuda Denshi Co, "Service Manual for DS-3300 Hardware Bedside Monitor", Tokyo, Fukuda Denshi Co., 1992
- [11] Choi SH, "Developing HL7-based medical information architecture", Dissertation, Deajeon KAIST, 2000
- [12] 김광준, 임세정, "HL7(Healthy Level Seven) 프로토콜을 이용한 실시간 환자 모니터링 웹 뷰어 시스템 개발", 한국해양정보통신학회 논문지, pp.546-555, Vol.13, No. 3, March 2009
- [13] George WB, "HL7 version 3-An object-orient methodology for collaborative standards development", International Journal of Medical Informatics, pp.151-161, 1998

저자소개



김 광 준(Gwang-Jun Kim)

1993년 조선대학교 컴퓨터공학과
졸업(공학사)

1995년 조선대학교 대학원 컴퓨터
공학과 졸업(공학석사)

2000년 조선대학교 대학원 컴퓨터공학과 졸업(공학박사)

2000년~2001년 Dept. of Electrical & Computer Eng.
Univ. of California Irvine Postdoc.

2003년~2006년 2월 여수대학교 컴퓨터공학과 조교수

2006년 3월~현재 전남대학교 컴퓨터공학과 조교수

※관심분야: ATM망, 인터넷 통신, 컴퓨터 네트워크, 실
시간 통신 프로그래밍, 가상화, 영상 처리 및 통신, 프
로그래밍 언어(Visual C++, Java), 의료정보 통신 등