

TMO를 이용한 실시간 생체정보 전송 모니터링 시스템 구현

임세정*, 김광준**

The Implementation of Real-Time Vital Sign Information Transmission Monitoring System using TMO

Se-Jung Lim*, Gwang-Jun Kim*

요 약

TMO 구조는 시간 구동 메소드와 메시지 구동 메소드의 두 가지 형태의 메소드로 이루어져 있으며, 일정한 시간이 지남에 따라 자동적으로 수행되는 시간 구동 메소드와 객체 노드 사이의 실행 결과를 주고받을 수 있는 서비스 메시지 구동 메소드로 명확하게 구분된다. 시간 구동 메소드의 실행은 실제 시간을 설정하는 과정에 결정된 특정한 값에 의해 실시간 통신 클럭의 도착에 따라 이루어지고 반면에 서비스 구동 메소드 실행은 클라이언트로부터 메시지를 요구하는 서비스에 의해서만 실행된다.

본 논문에서는 이러한 TMO 구조를 이용하여 실시간 통신 시뮬레이션 프로그래밍을 하기 위해 환자 모니터 원격진료 시스템 응용 환경에 적용하였다. Central Monitor로부터 전송되어진 환자의 생체정보 Raw Data가 HIS의 데이터 수신 모듈을 통해 사용가능한 데이터로 재구성될 수 있도록 설계가 이루어져 있다. 환자 생체정보에 대한 실시간성과 생체정보에 대한 생체정보 데이터의 연속성을 부여함으로써 베드 사이트의 환자에게서 발생된 모든 생체정보에 의해 환자 관리가 이루어진다.

ABSTRACT

The TMO may contain two types of methods, time-triggered methods(also called the spontaneous methods of SpMs) which are clearly separated from the conventional service methods (SvMs). The SpM executions are triggered upon design time whereas the SvM executions are triggered by service request message from clients.

In this paper, we describes the application environment as the patient monitor telemedicine system with TMO structure. Vital sign information web viewer systems is also the standard protocol for medical image and transfer. In order to embrace new technologies as telemedicine service, it is important to develop the standard protocol between different systems in the hospital, as well as the communication with external hospital systems. It is able to apply to remote medical examination and treatment. Through the proper data exchange and management of patient vital sign information, real time vital sign information management will offer better workflow to all hospital employee.

키워드 : TMO(Time and Message-Triggered Method), Patient Monitor Telemedicine System, Vital Sign Information

1. 서 론

최근에 급성장하고 있는 실시간 통신 분산 컴퓨팅은 최근에 컴퓨터 응용분야의 하나로서 컴퓨터 과학과 공학 분야에서 급성장하고 있는 한 분야이다. 실시간 객체 지향 분산 컴퓨팅은 분산된 컴퓨터 시스템에서 객체 네트워크의 형태로 구성된

실시간 분산 컴퓨팅이다. 실시간 통신 객체 모델에서 서비스 메시지를 분산된 노드사이의 객체들에게 송수신할 수 있는 시간동작이 존재해야 하며, 안정적인 상태에서 동작될 수 있도록 시뮬레이터 클럭을 필요로 한다[1].

객체 지향 실시간 통신 분산 프로그래밍 구조를 구체적으로 나타내기 위해 시간 구동과 메시지

* 전남대학교 컴퓨터공학과 석사과정(kgj@chonnam.ac.kr)

** 전남대학교 컴퓨터공학과 교수

구동(TMO : Time-triggered Message-triggered Object Model)에 대한 구조를 일반적인 형태의 구조로 모델링 하였으며, 이러한 모델링은 분산된 실시간 통신 객체와 비실시간 객체를 포함하여야 하고, 메시지 구동과 시간구동을 모든 객체 구조 형태에 적용함으로써 실시간 통신 프로그래밍 설계가 이루어질 수 있다.

TMO 구조는 시간 구동 메소드와 메시지 구동 메소드의 두 가지 형태의 메소드로 이루어져 있으며, 일정한 시간이 지남에 따라 자동적으로 수행되는 시간 구동 메소드와 객체 노드 사이의 실행 결과를 주고받을 수 있는 서비스 메시지 구동 메소드로 명확하게 구분된다. 시간 구동 메소드의 실행은 설계 시간을 설정하는 과정에 결정된 특정한 값에 의해 실시간 통신 클럭의 도착에 따라 이루어지고 반면에 서비스 구동 메소드 실행은 클라이언트로부터 메시지를 요구하는 서비스에 의해서만 실행된다.

본 논문에서는 이러한 TMO 구조를 이용하여 실시간 통신 시뮬레이션 프로그래밍을 하기 위해 환자 모니터 원격진료 시스템 응용 환경에 적용하였다. Central Monitor로부터 전송되어진 환자의 생체정보 Raw Data가 HIS의 데이터 수신 모듈을 통해 사용가능한 데이터로 재구성될 수 있도록 설계가 이루어져 있다. 환자 생체정보에 대한 실시간성과 생체정보에 대한 생체정보 데이터의 연속성을 부여함으로써 베드 사이트의 환자에게서 발생된 모든 생체정보에 의해 환자 관리가 이루어진다. 또한 보다 많은 검사장비의 데이터를 데이터베이스를 이용하여 전산화함으로써 해당 환자의 광범위한 자료 검색이 이루어지므로 획기적인 원격진료가 가능하다. 또한 적절한 생체정보 데이터의 교환과 정보 관리의 수정을 통해 병원 정보 시스템은 모든 병원 관계자의 보다 나은 업무처리를 향상시킬 수 있다.

II. TMO 구조와 엔진

2.1 TMO 구조

시간 구동 개념을 토대로 객체 지향 실시간 통

신 분산 프로그래밍 구조를 구체적으로 나타내기 위해 시간 구동과 메시지 구동에 대한 구조를 일반적인 형태의 구조로 모델링 할 수 있어야 하며, 이러한 모델링은 분산된 실시간 통신 객체와 비실시간 객체를 포함하여야 하고, 메시지 구동과 시간구동을 모든 객체 구조 형태에 적용함으로써 실시간 통신 프로그래밍 설계가 이루어질 수 있다. 그림 1은 Kane Kim에 의해 제안된 TMO(Time-triggered and Message-triggered Object) 구조를 나타내고 있으며, 시간 구동과 메시지 구동 방식을 실시간 통신 객체 지향 방식에 적용한 것으로서 다음과 같이 4개의 영역으로 구성되어 있다[2][3].

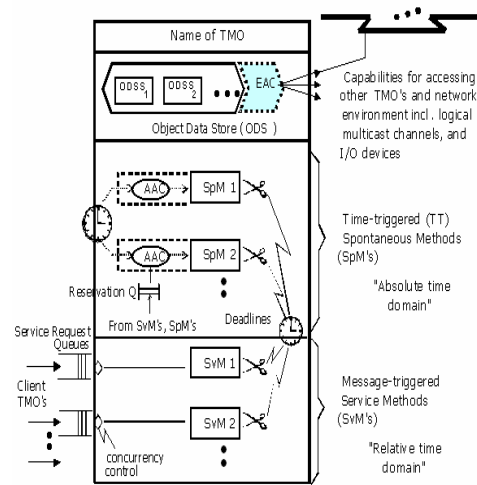


그림 1 실시간 통신 TMO 구조
Fig. 1 Structure of Real-time communication TMO

먼저, ODSS(Object-Data-Store section)영역은 객체 데이터의 저장 영역을 의미하는 것으로서 실시간 통신 시뮬레이션을 수행하기 위한 객체들의 데이터 멤버들로 이루어져 있다. ODSS영역의 데이터 멤버는 외부 클라이언트로부터 송신되거나 수신된 메시지 결과를 통해 갱신되며, 또한 외부 클라이언트로부터 서비스 요구를 호출한 후 호출된 객체의 수행결과로부터 생성된 메시지는 다른 노드 객체의 SpM(Service Spontaneous

Method)영역 또는 SvM(Service Message Method)영역으로 통신이 가능하다.

SpM(Service Spontaneous Method section) 영역은 시간 구동 메소드로서 자동적인 시간 측정을 위한 메소드 영역을 의미하며, 외부 클라이언트와 서버 객체 노드의 통신에 의해 자동적으로 실행되는 시간을 측정하는 영역이다.

SvM(Service-Message Mothod section) 영역은 서비스 구동 메소드를 의미하는 것으로서 ODSS 영역에 속해있는 객체 멤버들의 갱신 상태에 관련된 정보를 다른 객체 노드 SpM 메소드 영역이나 SvM 메소드 영역으로 송수신할 수 있다. 분산된 객체 노드간의 시간 구동 메소드와 메시지 구동 메소드가 동시에 실행되는 동시성을 만족시키면서 설정된 데드라인 시간 이내에 서비스 메시지 송수신의 완료가 이루어져야 한다.

EAC(Environment Access-Capability section) 영역은 다른 객체와의 효율적인 호출 경로를 설정해주거나 또는 입출력 장치와의 인터페이스 설정 영역을 나타낸다.

2.2 TMO 엔진과 프로그래밍

TMOSM/Linux는 TMOSM나 WTMO스와는 달리 리눅스 및 임베디드 리눅스 플랫폼에서 TMO를 지원할 수 있는 미들웨어엔진[5,6]으로 개발중이며, 실시간 지원을 최적화시킨 런타임 라이브러리를 포함하고 있다.

그림 2-5의 TMO 프로그래밍 환경은 TMO 프로그램을 정확히 실행시켜줄 수 있는 하드웨어 및 기능을 지원하는 소프트웨어로 구성된다. TMO응용프로그램은 TMO 처리엔진에서 제공하는 기능들을 사용하기 위해서 처리엔진과 직접적으로 통신하지 않는다. 대신에 TMO 응용프로그램은 TMOSL (TMO Support Library)라는 API를 사용하게 된다. 응용프로그램은 미들웨어 서비스 인터페이스(Middleware Service Interface)인 TMOSL을 사용하여 TMO 처리엔진 기능을 사용할 수 있으며, TMO 기반의 응용프로그램을 구현할 수 있는 라이브러리를 C++ 클래스 형태로 제공받는다[4].

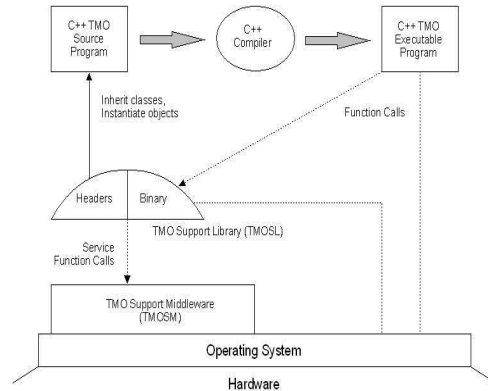


그림 2 TMO 프로그래밍 환경
Fig. 2 Programming Environment of TMO

TMO 응용프로그램은 TMOSL이 제공하는 기본 클래스를 상속 받아서 TMO 구성 요소인 ODSS, SpM 그리고 SvM를 TMO 클래스의 멤버로 정의하면서 시작된다.

다음 프로그래밍은 프로그램 예제 코드 중 TMO 클래스 정의를 보여주고 있다.

```
// Example of a TMO
#include <TMOSL.h>

// Definition of TMO1
class TMOClass1: public TMOBaseClass
{
private:
    ODSSClass1 m_ODSS1;
    SpMClass1 m_SpM1;
    SpVClass1 m_SvM1;
public:
    TMOClass1(const char* TMO_Name,
              TMOGateClass& Gate1, const tms&
              TMO_start_time)
        : m_SpM1("SpM1", gate1, m_ODSS1,
              RO),
          m_SvM1("SvM1", m_ODSS1, RW)
    {
        //Register this TMO to TMOSM
        activate(TMO_Name, TMO_start_time);
    }
}
```

III. 생체정보 수신모듈 TMO 프로세서

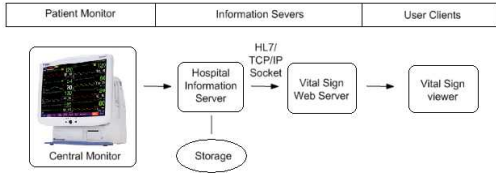


그림 3 환자 감시 시스템의 웹 뷰어 시스템의 구성

Fig. 3 Web viewer system configuration of patient monitoring system

그림 3에서 나타낸바와 같이 표준화된 HL7 프로토콜을 이용한 Vital Sign 생체정보 웹 뷰어 시스템 구성에서 Patient Monitor로부터 실시간 vital sign 생체정보 트리거 이벤트(Trigger Event)를 TCP/IP 소켓을 통해 병원정보 서버 시스템으로 전송한다. 병원정보 서버 시스템으로부터 수신된 환자 vital sign 생체정보를 사용자 클라이언트가 이용 가능하게끔 웹 서버를 구축 함으로서 수행될 수 있다.

그림 4는 TMO 구조를 이용하여 환자 모니터 원격진료 시스템의 Parsing-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 수치데이터를 나타내는 Numeric Measure Data Buffer, 생체정보의 파형을 나타내는 Wave Data Buffer로 객체 데이터 멤버로 이루어짐을 알 수 있다. Parsing_TMO ODSS 영역의 Patient Info, Numeric Measure Data buffer, Wave Buffer의 객체 데이터 멤버는 Main-TMO의 SpM을 통해 Parsing-TMO의 SvM을 호출함으로써 DS-7600 센트럴 모니터의 생체정보 row 데이터를 전송하여 ODSS 객체 데이터 멤버를 주기적으로 갱신한다. 수신된 생체정보의 이

진 정보 row 데이터를 Parsign 수신 모듈을 통해 베드 사이트의 환자 모니터 생체정보를 DS-7600 센트럴 모니터로 송신한 정보를 응용환경에 맞게 설정하기 위해 파싱 수신 모듈 알고리즘을 적용한다.

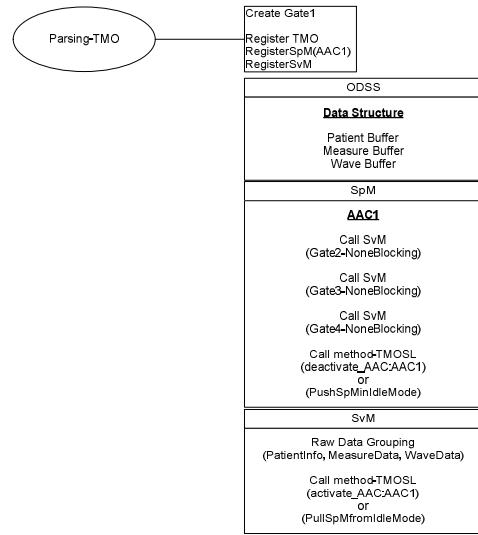


그림 4 환자 모니터 원격진료 시스템의 Parsing-TMO

Fig. 4. Parsing-TMO Patient Monitor Telemedicine System

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 Measure Data TMO 및 생체정보 파형을 나타내는 Wave Data TMO로 정의하였다. 이러한 TMO에 접근하기 위해서 Parsign-TMO내의 SpM을 통해 Patient Info, Measure, Wave의 SvM을 호출함으로써 각각의 분산된 노드에서 TMO를 수행함으로써 센트럴 모니터로부터 송신된 환자의 생체정보에 대한 객체에 대해 주기적으로 계속해서 갱신된 데이터를 전송한다. Main-TMO로부터 송신된 생체정보의 row 데이터는 Parsing-TMO의 SvM을

통해 ODSS 영역의 객체 데이터 맴버 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);
    TMOSLprintf(_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);
    TMOSLprintf(_T("<ParsingTMO_Regist Successfully>\n"));
}

BOOL CParsingPro::ParsingPatientSpM()
{
    TMOSLprintf(_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)
            TMOSLprintf(_T("<PatientGate NonBlockingSR Successfully>\n"));
        else
            TMOSLprintf(_T("<PatientGate NonBlockingSR Failure>\n"));
        MicroSec system_age = GetCurrentDCSage();
        deactivate_AAC (_T("aac_patient"), tm4_DCS_age(system_age));
    }
    return TRUE;
}

```

IV. 생체정보 뷰어 시스템

그림 5는 환자의 생체정보를 웹을 통해 실시간 검색할 수 있도록 구성된 웹 뷰어 시스템에서 의료진 개인에게 부여된 아이디로 로그인 과정에

성공하면 나타나는 “Trend Graph” 라는 기능의 선택으로 제공되는 화면으로 환자의 계측 수치가 본 화면에서 선 그래프와 그에 해당하는 수치에 대한 검색으로 가능하다.

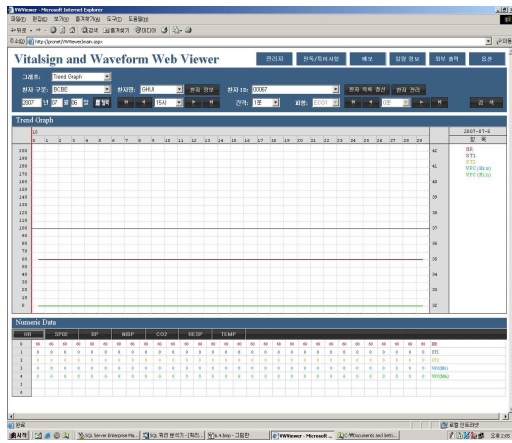


그림 5 생체정보 웹 뷰어 시스템 Trend Graph 인터페이스 화면

Fig. 5 Trend graph interface display of vital sign information web viewer system

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 계측 수치를 그래프와 수치로 된다.

그림 6은 환자의 생체정보를 웹을 통해 실시간 검색할 수 있도록 구성된 웹 뷰어 시스템에서 “Wave Graph” 라는 기능의 선택으로 제공되는 화면으로 환자의 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.

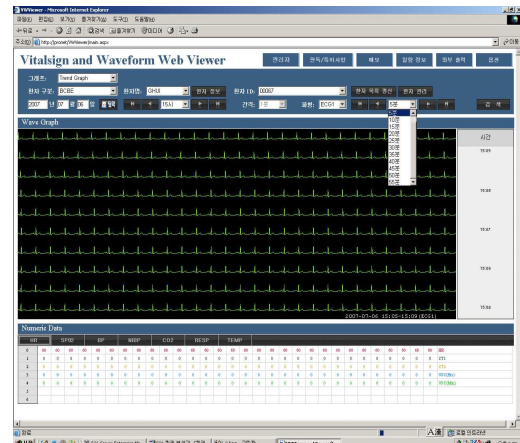


그림 6 생체정보 웹뷰어 시스템 Wave Graph(ECG1) 인터페이스 화면

Fig. 6 Wave graph(ECG1) interface display of vital sign information web viewer system

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

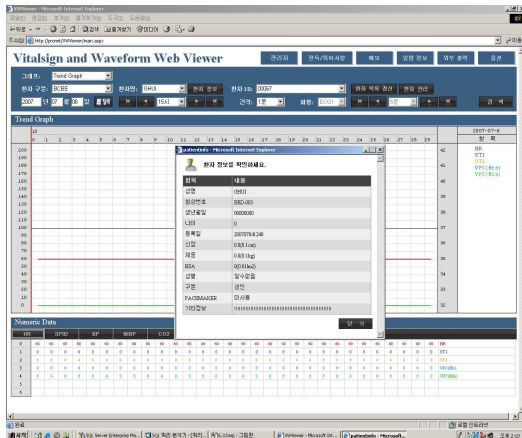


그림 7 생체정보 웹뷰어 시스템 환자 정보 인터페이스 화면

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

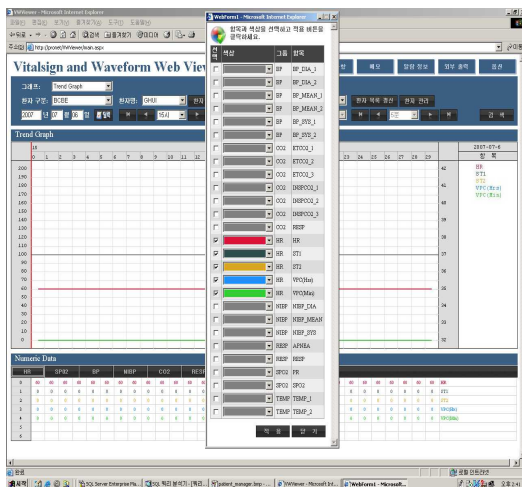


그림 8 생체정보 웹뷰어 시스템 옵션 인터페이스 화면

Fig. 8 Option interface display of vital sign information web viewer system

그림 8은 “옵션” 메뉴의 선택으로 나타나는 화면으로 Trend Graph의 항목과 색상에 적용되는 기능이다. 이 기능에서 선택된 항목과 색상은 이 화면의 적용 버튼을 누름과 동시에 웹 뷰어 시스템의 메인 화면의 Trend Graph에 선택된 항목과 그 항목의 색상으로 적용된다.

옵션에서 선택할 수 있는 항목은 총 27개의

항목이다. 하지만, Central Monitor로부터 획득되어 의미 있는 데이터로써 웹 뷰어 시스템에서 제공되는 항목은 HR, VPC, ST1, ST2에 제한적이다. 이 부분은 향후 추가적인 개발과 확장이 가능하리라 기대하고 있다.

V. 결론

Central Monitor에 나타난 생체정보를 TCP/IP 인터넷 프로토콜을 통해 HIS에 구성된 스토리지에 저장하게 된다. 이렇게 저장된 각 환자들의 생체 정보는 사용자의 필요에 따라 웹을 통한 Viewer로 실시간 검색할 수 있도록 구현하였다. Central Monitor로부터 전송되어진 환자의 생체정보 Raw Data가 HIS의 데이터 수신 모듈을 통해 사용가능한 데이터로 재구성될 수 있도록 설계가 이루어져 있다. 환자 생체정보에 대한 실시간성과 생체정보에 대한 생체정보 데이터의 연속성을 부여함으로써 베드 사이트의 환자에게서 발생된 모든 생체정보에 의해 환자 관리가 이루어짐을 알 수 있다.

HL7 프로토콜을 이용한 생체정보 파싱 모듈을 통해 Patient Monitoring 웹 뷰어 시스템 구현은 환자 생체정보의 실시간적 모니터링을 통해 환자에 대한 결과를 시간과 장소를 구분하지 않고 조회할 수 있으므로 해당 환자에 대한 빠른 조치로 응급 상황에도 신속, 정확하게 대응할 수 있다. 또한 보다 많은 검사장비의 데이터를 데이터베이스를 이용하여 전산화함으로써 해당 환자의 광범위한 자료 검색이 이루어지므로 획기적인 원격진료로 활용이 가능하며 이미 앞서서도 언급한것과 같이 진료 경험을 공유할 수 있는 통계의 목적으로 활용되는 물론 학술 및 연구의 목적으로 활용 가능하다.

참고 문헌

[1] K. H. Kim, C. Subbaraman, and L. Bacellar, "Support for RTO.k Object Structured

Programming in C++", Control Engineering Practice 5, pp.983-991, 1997.

[2] E. Shokri, P. Crane, and K. H. Kim, "An implementation model for time-triggered message-triggered object support mechanisms in CORBA-compliant COTS platforms", Proc. IEEE 1st Int'l Symp. on Object-oriented Real-time Dependable Computing (ISORC), Kyoto, Japan, pp.12-21, April 1998.

[3] J. G. Kim et al., "A soft real-time RTSP platform - WRTSPS and Implementation techniques", Proc. IEEE 1st Int'l Symp. on Object-oriented Real-time Dependable Computing (ISORC), Kyoto, Japan, pp.256-264, April 1998.

[4] E. Shokri, P. Crane, and K. H. Kim, and C. Subbaraman, "Architecture of ROAFTS/Solaris: A Solaris-based Middleware for Real-Time Object-Oriented Adaptive Fault Tolerance Support", Proc. COMPSAC '98, IEEE CS 22nd Int'l Computer Software & Applications Conf., Vienna, Austria, pp.90-98, August 1998.

[5] D. C. Schmidt, D.L. Levine, and S. Mungee, "The Design and Performance of Real-Time Object Request Brokers", Computer Communications, vol.21, pp.294-324, Apr. 1998.

[6] J. G. Webster "Encyclopedia of Medical Devices and Instrumentation", New York, John Wiley & Sons, 1988.

[7] L. A. Geddes and L. E. Baker, "Principles of Applied Biomedical Instrumentation," New York, John Wiley & Sons, 1989.

[8] Fukuda Denshi Co "Service Manual for DS-3300 Hardware Bedside Monitor", Tokyo, Fukuda Denshi Co., 1992

[9] Park JH. HL7 RIM based object oriented analysis & design for nursing information system. [Dissertation]. Seoul:Yonsei University;2002

[10] HL7 Modeling & Methodology Committee. HDF preliminary draft. La Verne.: Health Level Seven, Inc.;2004. pp.67-90.

저자약력



임 세 정(Se-Jung Lim)
 2008년 전남대학교
 컴퓨터공학과 졸업(공학사)
 현재 전남대학교 대학원
 컴퓨터공학과 재학

<관심분야> TCP/IP혼잡제어, 생체정보 및 의료정보, 무선 인터넷, 이동 통신 등



김 광 준(Gwang-Jun Kim)
 1993년 조선대학교
 컴퓨터공학과 졸업(공학사)
 1995년 조선대학교 대학원
 컴퓨터공학과 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, 실시간 통신 프로그래밍, 영상 처리 및 통신, 프로그래밍 언어(Visual C++, Java), 의료정보 통신 등