분산 네트워크 시스템에서 TMO를 이용한 실시간 통신 시뮬레이션 구현

김광준* · 서종주* · 강기웅* · 윤찬호**

The Implementation of Real Time Communication Simulation using TMO in Distributed Network systems

Gwang-Jun Kim* · Jong-Joo Seo* · Ki-Woong Kang* · Chan-Ho Yoon**

요 약

본 논문에서는 TMO 실시간 객체 모델을 이용하여 실시간 통신 메시지 서비스를 효과적으로 지원하기 위해 새로운 프레임워크 및 동기화 메카니즘을 나타내었다. 또한 분산된 네트워크 시스템에서 TMO 구조를 이용하여 DHS(Distributed High-Precision Simulation) 응용 환경에 적용함으로서 실시간 메시지 서비스를 보장하였다. 분산된 다중 노드 시스템에서 TMO의 시간 구동 및 메시지 구동 구조는 실시간 통신 서비스 능력을 적시에 보장하기 위한 설계자의 노력을 충분히 줄일 수 있었으며, 제안된 프레임워크는 분산된 객체 구성요소들 사이의 데드라인 시간을 보다 쉽게 보장하기 위해 일관된 구조 및 구성을 제공하였다. 프로그래머의 데드라인 설계 시간을 처음 객체부터 적용하여 보장함으로서 형성될 수 있다. 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 new framework and synchronization mechanism to effectively support developing real-time communication service by using a real-time object model named TMO (Time-Triggered Message-Triggered Object). Also, we describes the application environment as the DHS(distributed high-precision simulation) to guarantee real-time service message with TMO structure in distributed network systems. The TMO scheme is aimed for enabling a great reduction of the designer's effort in guaranteeing timely real-time communication service capabilities of among distributed multi-nodes systems. Our real-time framework provide the consistent construction and configuration of time-triggered processing components across heterogeneous distributed object environment more easily. It has been formulated from the beginning with the objective of enabling design-time guaranteeing of timely action. 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

Real-time service, TMO(Time-triggered and Message-triggered Object), DHS(Distributed high-precision simulation)

* 전남대학교 컴퓨터공학과
** 조선대학교 컴퓨터공학과

접수일자 : 2007. 3. 2

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 있어 서 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 수행 시간에 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 통신 시스템에서는 계산 결과의 논리적 정확성과 결과가 산출되는 시간의 정확성을 요구한다[7]. 만일 시스템이 시간 제약 조건을 만족하지 못하면, 시스템 실패가 발생한 것으로 간주하며, 또한 실시간 통신시스템은 어떠한 기능을 정확히 수행할 뿐 아니라 외부의 비동기적인 사건에 대하여 주어진 시간 안에 응답할 수 있는 시스템이어야 한다[3].

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[1,3,4,5]. TMO 모델에서 제공되는 시간 구동 메소드와 메시지 구동 메소드에 의해 실시간 시스템이 갖는 시간적인 특성과 이벤트를 쉽게 추상화 할 수 있는 구조를 가지고 있을 뿐 아니라, 적시 서비스 능력(timely service capability)을 설계 단계에서부터 보장할 수 있다.

본 논문에서는 이러한 실시간 객체 모델의 TMO 구조를 이용하여 실시간 통신 메시지 서비스 보장을 위해서 DHS(Distributed High-Precision Simulation)라는 응용환경을 만들어 비행체 추적 알고리즘을 사용함으로서 TMO 구조의 실시간성을 입증하고자 한다.

Ⅱ. TMO의 구조 및 특성

TMO는 Time-triggered Message-triggered Object의 약자로써 Kane Kim 등에 의해서 개발된 Object Structuring Scheme이다[4,5,6] 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 뿐 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 Mehtod)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 2-1은 TMO 객체 모델의 기본적인 구조를 나타내고 있으며, 기존의 객체 모델과는 다른 다음과 같은 특징을 가지고 있다.

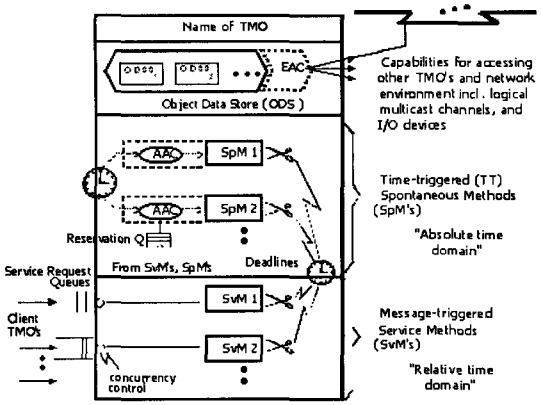

그림 2-1. TMO 기본 구조
Fig. 2-1. Basic Structure of TMO

2.1. 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 멀티 노드의 객체들은 non-blocking 형태의 원격 메소드 호출을 통하여 분산 처리를 수행한다.

2.2. 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 설계시에 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 되어야되며 t+10분 안에 완료된다는 의미를 가지고 있다.

2.3. Basic concurrency constraint (BCC)

TMO들의 시간적인 서비스 능력을 보장하기 위한 제약 조건으로써, SpM과 SvM이 공유데이터 ODS를 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.

2.4. 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 호출함으로서 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 비블록킹 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.Svl 1-name(parameter-1,parameter-2 ,···,by dl adline)

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 나타낸 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 메소드를 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 함으로서 서비스 메시지 메소드의 실행과 시간 구동 메소드의 실행사이에 발생하는 충돌을 방지할 수 있다. 하나의 서비스 메소드가 복잡하게 실행되는 경우에 시간 윈도우의 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 시간 윈도우 내에서 실행이 가능하게 할 수 있으며, 또한 분할된 서비스 메소드의 전송 오버헤드(overhead)를 줄임으로서 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 실행함으로서 실행 엔진에서 수행되고 있는 서비스 메소드 호출을 종료할 수 있다. 호출한 서비스 메소드에 요구된 서비스의 결과를 큐에 저장함으로써 서비스 메소드에 대한 호출을 완료할 수 있다. 호출한 서비스 메소드에 결과를 나타내는 상태가 되돌아올 때까지 계속해서 결과를 되돌려 준다.

실시간 통신 프로그래밍 설계자는 외부의 클라이언트가 첫 번째 서비스 메소드를 호출함으로서 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 변수

와 객체 저장 영역의 데이터를 공유함으로서 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 시스템에서 임의의 서비스 메소드로부터 돌아오는 결과를 수용하는 것은 특별한 경우이다.

Ⅳ. 분산 네트워크 시스템에서 실시간 통신 메시지 서비스 보장

4.1 TMO로 구현한 DHS

실시간 통신 시뮬레이션 프로그래밍을 하기 위해 TMO 구조를 이용하여 DHS(Distributed High-Precision Simulation)라는 응용 환경에 적용하였다. 그림 4-1은 TMO 구조를 이용하여 DHS를 나타낸 것으로서 객체 데이터 맴버들은 각 괄호없이 이루어지며, DHS에서 유지되고 있는 정보는 객체 데이터 저장 영역내에 객체 멤버에 대한 정보를 나타낸다. 객체 데이터 저장 영역은 다음 세 가지 환경 구성 요소에 대한 상태를 나타내는 객체 멤버들로 구성되어 있다.

- **Flying Airplane Group Container Information (Environment)**
- **Flying Object Tracking Information (Reporter)**
- ***DHS Space(Sky and Land)***

DHS를 구성하고 있는 객체 데이터 멤버는 SpM을 이용하여 주기적으로 정보를 갱신한다. DHS TMO내의 SpM은 계속해서 동작하고 각각 그들의 실행을 순간적으로 완료한다. 그러므로 SpM 영역은 객체 멤버의 상태 변화를 계속적으로 나타내고, 이는 DHS 환경 구성 요소인 ODSS 세그먼트 부분의 객체 멤버 데이터를 계속적으로 SpM 영역을 참조하여 주기적으로 갱신된다. 또한 다중 SpM 영역은 응용 환경 구성 요소들 사이에 존재하는 병렬적인 특성을 지속적으로 나타내기 위해 사용되는 것으로서 동시에 동작될 수 있다.

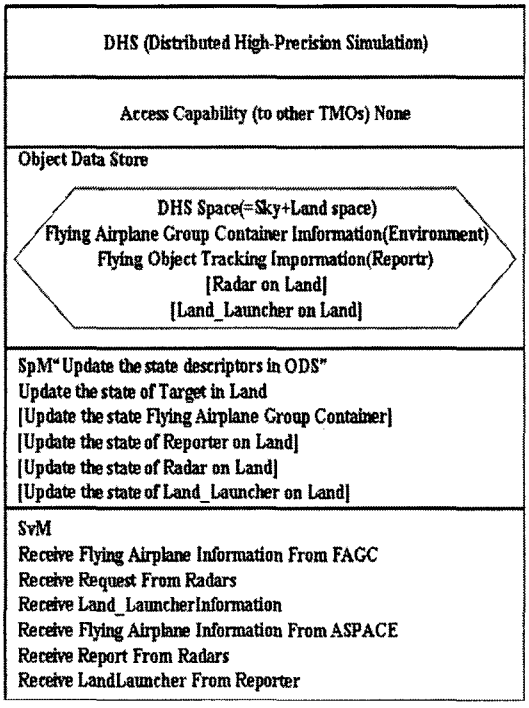

그림 4-1. DHS에 대한 TMO 명세서
Fig. 4-1. High-Level Specification of the DHS TMO

DHS에서 공간을 나타내는 DHS Space 상태 정보는 DHS에 대한 지리학적인 정보를 제공할 뿐만 아니라 DHS 공간에서 움직이는 모든 구성 요소의 정보를 유지한다. 이러한 정보는 구성 요소들 간의 이벤트 발생을 결정하기 위해 사용되며, DHS로부터 임의의 이벤트 구성 요소의 시작을 인식하기 위해 사용된다.

응용 환경을 이루고 있는 DHS 객체 멤버에 대해 TMO 구조를 적용하여 ODS 영역, SpM 영역, SvM 영역 간의 상태를 나타냄과 동시에 실시간 통신 시뮬레이션 객체 메소드들 간의 동작 상태를 나타낸다. 실시간 통신 시뮬레이션을 지원하기 위해 객체 실행 엔진에 의해 제공되는 시간 구동 메소드의 동작 주기를 선택한다. DHS를 이루고 있는 구성 요소들 간의 실제적인 시뮬레이션 과정은 시간 구동 메소드의 연속적인 동작을 토대로 실행되지 않는 객체 멤버들의 상태에 대한 설명이 구체적으로 표현되어 있지 않다. TMO의 구조를 이용한 시뮬레이션의 정확성은 자동적인 시간 동작 측정을 할 수 있는 시간 구동 메소드의 동작 주기를 정확히 예측하여 선택함으로서 수행될 수 있게 한다.

4.2 시간 구동 메소드와 메시지 구동 메소드 처리

그림 4-2는 TMO 구조를 이용한 비행체 추적 프로그램을 나타내는 것으로서 그림 4-1에서 나타낸 DHS를 이루고 있는 각각의 Environment, Reporter, Radar, Land_Launcher TMO의 SpM 영역과 SvM영역간의 메시지 처리과정을 나타낸다. 먼저, Radar TMO는 Environment TMO에 속해 있는 FAGC에 접근해 비행체의 위치 정보 메시지를 요구함과 동시에 기존의 Reporter TMO는 Radar로부터 넘겨받은 비행체 추적 위치 정보 메시지를 가지고 있다. FAGC TMO ODS영역의 비행체 추적 객체는 SpM 영역에서 일정한 시간이 경과함에 따라 자동적으로 이동체의 움직임이 갱신된다.

비행체 추적 움직임의 갱신에 따라 Radar TMO의 ODS 영역의 Radar의 객체는 SpM영역에서 Spot_Check의 방법을 통해 비행체 추적의 위치 정보를 검출한다. 검출된 비행체 추적의 위치 정보는 Radar TMO의 ODS영역의 Radar 객체에 넘겨준다. 또한 갱신된 Radar의 SpM 영역은 클라이언트 전송 호출을 통해 Reporter의 SvM영역으로 메시지를 넘겨준다. Radar로부터 넘겨받은 메시지는 SvM영역의 수행 알고리즘에 따라 처리됨과 동시에 SpM 영역에서 비행체에 추적 위치가 위험지역에 도달했는지의 여부를 일정한 시간의 경과에 따라 계속해서 주기적으로 추적 검사한다.

비행체 추적의 위치 정보가 DHS Space의 위험지역에 도달하게 되면 Reporter TMO는 위험 경고를 제공받게 되며, Reporter로부터 Land_Launcher SvM 영역에 비행체의 위치 정보를 제공함으로서 Land_Launcher로부터 미사일이 발사되어 요격하게 된다. Reporter TMO의 ODS영역의 객체는 SpM 영역의 수행결과에 따라 갱신되며, 또한 Land_Launcher TMO의 ODS 영역의 객체도 SpM 영역의 수행결과에 따라 계속해서 갱신된다.

Radar의 SpM 영역은 클라이언트 전송 호출을 통해 Environment TMO내의 ASPACE TMO SvM영역에 갱신된 Radar의 메시지를 넘겨준다. Radar로부터 넘겨 받은 메시지의 결과에 따라 ASPACE의 ODS영역의 비행체 추적 객체가 갱신되는 것으로서 DHS를 이루고 있는 구성 요소들 간의 시간 구동 메소드와 서비스 메시지 메소드 처리 과정에 대해 구체적으로 나타내고 있다.

Environment TMO 구조에서 실선으로 나타낸 것은 실제적인 TMO 구조를 나타내고 점선으로 표현된 것은 가상적인 TMO구조를 나타내고 있다. 이들 실선과 이중 점선의 차이점은 TMO구조가 복잡하게 되면 가상적인 TMO를 분리하여 또 하나의 노드에서 동작 가능한 실제적인 TMO구조에서 비행체 객체가 추적된다.

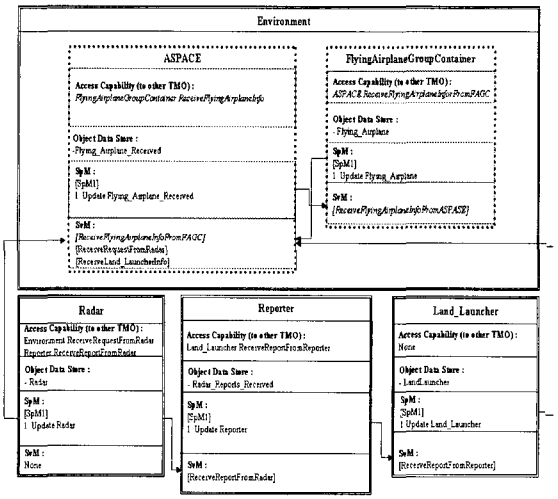

그림 4-2. DHS 각 구성 요소간의 시간 구동 메소드와 서비스 메시지 메소드의 처리과정
Fig. 4-2. Processing procedure of SpM and SvM among DHS components

4.3. 실시간 컴퓨팅 시스템에서 실시간 명세

DHS는 네 개의 실시간 통신 객체 네트워크로 이루어졌다. 네 개의 TMO 형태의 구조로 된 각각의 절차를 작성하고 난 후 Reporter에 하나의 제어 시스템, Radar와 Land_Launcher와 같은 제어 시스템을 추가하였다.

TMO의 상세 명세 설계 개요도를 나타내기 위해 Radar 제어 컴퓨터 시스템만을 고려하였다. 먼저 이러한 시스템을 구성하기 위해 실시간 통신 프로그래밍 설계자는 두 개의 중요한 데이터 멤버로 구성된 객체 데이터 저장 영역을 이용하여 단일 TMO의 구조로 된 명세서를 작성한다.

. Radar Data Queue(RDQ): 수신된 Radar 데이터를 포함하고 있는 큐

. Flying Airplane Group Container Tracking (FAGCT) Inforamtion: 날아다니는 비행체 객체의 위치 추적에 필요한 정보를 포함

그림 4-3은 Radar 데이터 큐 TMO에 대한 단일 설계 명세서를 나타내고 있으며, 이는 Reporter에서 나타난 바와 같은 동일한 방법을 이용한다. RDQ TMO내에 들어오는 데이터는 FAGCT TMO에서 사용된 Spot_Check 알고리즘을 통해 발생하는 결과가 입력된다. 데이터를 전송하기 위해 RDQ는 FAGCT에 의해서 발생되는 최근의 정보를 자주 참조하며, 이러한 것을 지원하기 위해 FAGC는 각각의 Radar 데이터를 RDQ로 보낸다.

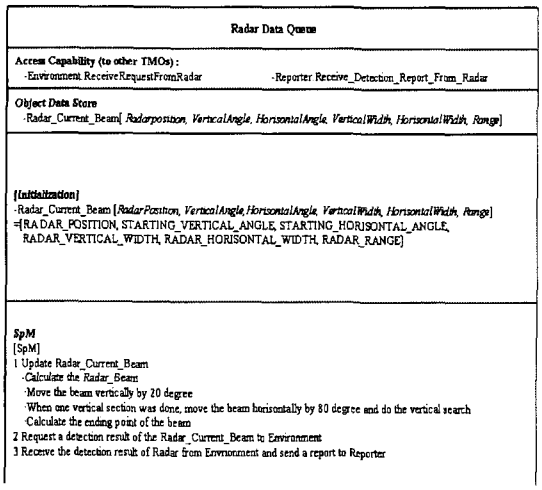

그림 4-3. Radar 제어 시스템의 단순 명세서
Fig. 4-3. Simple Specification of Radar Control System

그림 4-4는 RDQ의 상세 설계 명세서를 나타내는 것으로서 DHS에서 Radar로부터 비행체 객체의 정보를 SMM1이 수신하고 SMM2는 FAGCT로부터 요구된 Spot-Check Radar 정보를 수신한다. SpM1은 주기적으로 FAGC에 요구된 ID 수에 따라 Radar 데이터를 송신한다.

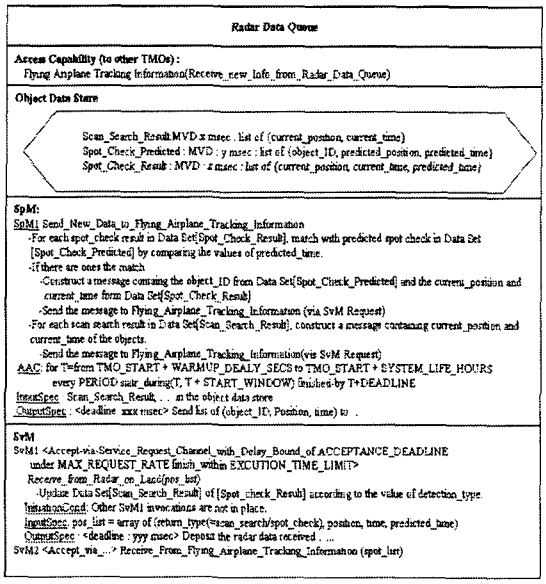

그림 4-4. Radar 데이터 큐 TMO 상세 설계 명세서
Fig. 4-4. Detailed Design Specification for Radar Data Queue TMO

실시간 통신을 필요로 하는 분산된 객체 노드에 TMO 구조를 이용하여 객체 노드간의 실시간 통신 메소드를 SpM 영역과 SvM 영역으로 명확하게 분리하여 적용할 수 있다. 분산된 객체 지향 실시간 통신시스템에서 TMO 구조를 이용한 다단계 프로그래밍 설계는 객체에 요구된 단순 절차, 상세 절차에 따라 실시간 프로그래밍을 유연하고도 쉽게 작성할 수 있다.

V. 결 론

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 부

족한 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 시뮬레이션 환경을 구축하였다. 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 확보하였다. 또한 분산된 객체 노드의 시뮬레이터 클럭에 시간 구동 메소드를 적용함으로서 각각의 객체 노드에 부여된 과중한 부하를 줄일 수 있으며, 시뮬레이터 클럭의 시간 간격을 충분히 크게 함으로서 시뮬레이션 결과 메시지의 교환을 보장할 수 있다.

분산된 객체 지향 실시간 통신시스템에서 TMO 구조에서 추적하는 다단계 프로그래밍 설계는 객체에 요구된 단순 절차, 상세 추적 절차를 작성함으로서 실질적인 실시간 통신 시뮬레이션 디자인과 코딩 과정사이에 강력한 연관성 흐름에 따라 실시간 통신 프로그래밍을 유연하게 작성 하였다.

참고문헌

[1] Kim, K.H., *"Real-Time Object-Oriented Distributed Software and the TMO Scheme"*, Int'1 Jour. of Software Engineering & Knowledge Engineering, Vol. No2, pp.251-276, April 1999.

[2] Kopetz, H., *"Real-Time Systems: Design Priciples for Distributed Embedded Applications"*, Kluwer Academic Pub., ISBN: 0-7923-9894-7, Boston, 1997

[3] Kane Kim, *"Toward New-Generation Object- Oriented Real-Time Software and System Engineering"*, SERI Journal, Vol. 1, No 1, pp.1-13, January 1997.

[4] K. H. Kim, *"Object Structures for Real-Time Systems and Simulators"*, *IEEE Computer 30*, pp.62-70, 1997.

[5] K. H. Kim and J. Liu, *"Deadline Handling in Real-Time Distributed Objects"*, *Proc. ISORC 2000*, Newport Beach, CA, pp.7-15, March 2000.

[6] 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"1 Symp. on Object-oriented Real-Time Distributed Computing), pp.54-63, May 19999.

[7] M. Champlain, *"Synapse: A Small and Expressive Object-based Real-time Programming Language"*, ACM SIGPLAN Notices 25 pp.124-134, 1990

[8] Y. Ishikawa, H. Tokuda, and C. W. Mercer, *"An object-oriented real-time programming language"*, *IEEE Computer*, pp.66-73, 1992.

[9] H. Kopetz, et al., *"Fault-Tolerant Membership Service in a Synchronous Distributed Real- time System"*, *Proc. IFIP WG 10.4 Conf. on Dependable Computing for Critical Appl.*, Santa Barbara, pp.167-174, Aug. 1989.

저자소개

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 Elec trical & Computer Eng. Univ. of California Irvine Postdoc.
2003년~2006년 2월 여수대학교 컴퓨터공학과 조교수
2006년 3월~현재 전남대학교 컴퓨터공학과 조교수
※주관심분야 : ATM망, 인터넷 통신, 컴퓨터 네트워크, 실시간 통신 프로그래밍, 영상 처리 및 통신, 프로그래밍 언어(Visual C++, Java), 의료영상 통신 등

서 종 주(Jong-Joo Seo)

2004년 여수대학교 컴퓨터공학과 졸업 (공학사)
현재 : 전남대학교 대학원 컴퓨터공학과
※주관심분야 : 광대역 종합통신망, 실시간 통신, 컴퓨터 네트워크, TCP/IP 혼잡제어, 이동 통신 등

강 기 웅(Ki-Ung Kang)

2005년 여수대학교 컴퓨터공학과 졸업 (공학사)
현재 : 전남대학교 대학원 컴퓨터 공학과
※주관심분야 : ATM망, 실시간 데이터 통신, 모바일 게임, TCP/ IP 혼잡제어, 이동 통신 등

윤 찬 호(Chan-Ho Yoon)

1997년 호남대학교 컴퓨터공학과 졸업 (공학사)
2000년 조선대학교 대학원 컴퓨터 공학과 졸업(공학석사)
현재 : 조선대학교 대학원 컴퓨터 공학과 박사과정
※주관심분야 : TCP/IP혼잡제어, ATM망, 데이터 통신, 네트워크 시뮬레이션, 이동 통신 등