

장애 감내형 CORBA

충남대학교 김기영·최 훈*

1. 서 론

서로 다른 종류의 시스템들이 네트워크를 통하여 연결된 분산 시스템 환경 하에서 각 시스템들은 서로 독립적으로 동작하면서 다른 시스템들과 서비스를 주고받는 클라이언트/서버 관계를 가진다. 사용자는 분산 시스템을 통해 보다 향상된 서비스를 보장받기를 요구하는데, 이때 가장 기본이 되는 것이 서버에 장애가 발생해도 사용자가 지속적으로 서비스를 제공받을 수 있는 장애 감내성(fault tolerance)이라고 할 수 있다.

90년대에 들어 객체 지향 분산 처리 플랫폼인 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture)[1]가 제안되면서 CORBA 하부 구조를 이용하는 어플리케이션 개발이 보편화되고 있다. 그러나, 현재 CORBA는 점 대 점(point to point) 통신 방식을 기본으로 하고 있기 때문에 장애 감내성 있는 분산 어플리케이션의 구현에 적합하지 않다. 따라서 새로운 분산 처리 플랫폼에 관한 연구가 활발히 진행되고 있다.

본 논문에서는 장애 감내형 서비스를 제공하는데 필요한 요소 기술을 설명하고 CORBA 환경에서의 장애 감내형 서비스 제공에 대한 최근 연구 동향들을 소개한다. 먼저 2절에서는 장애 감내성 향상을 위해 필요한 시스템 요소 기술들을 간단히 살펴본다. 3절에서는 통합(Integration) 방식의 연구들을 소개하고 4절

에서는 서비스(Service) 방식의 연구들을 알아본다. 5절에서는 이벤트(Event) 방식과 인터셉트(Intercept) 방식을 소개한다. 6절에서는 [2]연구에 의해 CORBA 환경에서 개발한 디지털 도서관 시스템에 적용된 장애 감내형 다중화 서버 구조와 방식을 설명하고, 7절에서 OMG의 표준화 동향을 소개한다.

2. 장애 감내성을 위한 요소 기술

장애 감내성을 위해서는 서비스를 제공하는 객체를 네트워크의 여러 장소에 중복(replication)시켜 놓음으로써, 한 서비스 객체에 장애가 발생해도 다른 객체가 계속해서 서비스를 제공하게 해야 한다. 중복을 이루고 있는 서비스 객체들은 그룹 내에서 능동적 중복과 수동적 중복 형태로 서비스를 제공할 수 있다. 객체 그룹 멤버들이 능동적 중복을 이루고 있는 경우, 클라이언트 객체의 서비스 요청은 그룹 레퍼런스(객체 주소 및 식별자)를 통해 그룹 내의 모든 멤버 객체들에게 전달되어야 하며, 중복된 객체들로부터의 응답들 중 하나만이 클라이언트 객체에게 전달되어지도록 해야 한다. 수동적 중복을 이루고 있는 경우에는 그룹 레퍼런스를 통해 전달되는 클라이언트 객체의 서비스 요청이 그룹 내의 멤버들 중 주(primary) 멤버 객체에게만 전달되어지며, 주 멤버 객체로부터의 응답을 클라이언트 객체에게 전달해야 한다.

서비스 객체가 중복되어 있는지 여부나 어떠한 형태의 중복을 이루고 있는가에 관계없이,

*중선회원

객체 그룹은 클라이언트 객체에게 하나의 단일 서비스 객체처럼 투명하게 보여져야만 한다. 그룹 내의 멤버 객체들에 대한 모든 레퍼런스(reference)는 네임 서비스(name service)를 통해 그룹 레퍼런스 하나로부터 얻어질 수 있어야 한다.

서비스 객체들이 하나의 그룹을 구성하고 있으므로 객체 그룹을 관리하는 기능들이 필요하다. 첫째, 그룹 멤버십에 변화가 생겼을 경우, 예를 들어 새로운 서비스 객체가 그룹에 참여하거나 기존의 멤버 객체가 그룹을 탈퇴하는 경우 이를 모니터링 하는 기능이 필요하다. 둘째, 그룹 멤버십의 변화로 인하여 멤버 객체들 간의 동일한 내부 상태와 뷰(view, 현재 그룹에 참여하고 있는 멤버 객체들의 리스트)를 유지하기 위한 상태 전달 기능이 필요하다. 셋째, 멤버 객체에게 장애가 발생하여 더 이상 서비스를 제공하지 못하는 경우 이를 발견하고 장애 발생 사실을 알려 장애를 복구하는 기능이 필요하다.

능동적 중복을 이루고 있는 경우 클라이언트의 서비스 요청이 그룹 내의 모든 멤버들에게 동일한 순서로 모두 전달되어야 객체들간 서비스 정보의 일치성(consistency)을 유지할 수 있다. 따라서 객체 그룹에 대한 신뢰성(reliable) 있는 멀티캐스트 기능, 즉 순서성(ordered) 있고 원자성(atomic) 있는 멀티캐스트 기능이 요구된다.

장애 감내성의 향상을 위해 위의 사항들을 고려한 분산 시스템의 구조와 구현 기술에 관한 여러 연구들이 진행되어 왔으며, 이들 연구들은 모두 중복된 서비스 객체들로의 그룹 통신에 기반하고 있다.

3. 통합(Integration) 방식의 장애 감내형 CORBA

통합 방식은 서비스 객체를 중복시키고, 기존 OMG의 ORB에 그룹 툴킷(group toolkit)을 통합시킨 새로운 형태의 ORB가 중복된 객체의 그룹 관리 및 멀티캐스트와 같이 장애 감내성 제공에 필요한 기본 기능들을 제공하는 방식이다. 대표적인 사례로는 Orbix+Isis[3]

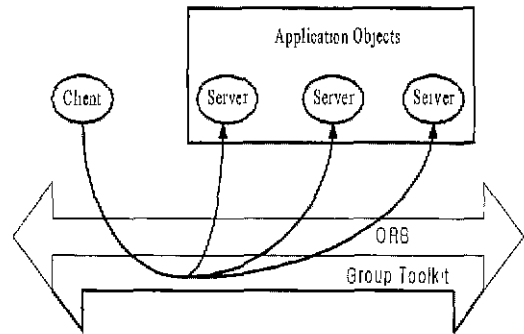


그림 1 통합 방식

와 Electra[4]가 있다.

3.1 Orbix+Isis

Orbix+Isis는 동적인 객체의 중복 관리, 장애 발견 기능들을 제공하기 위하여 기존 ORB에 Isis[5]를 통합한 것으로서, 클라이언트의 코드를 변경하지 않고 서비스 객체들을 그룹으로 관리함으로써 장애 감내성과 신뢰성을 투명하게 제공한다[6]. 서비스 객체들은 Active Replica와 Event Stream 두 개의 클래스(class)를 상속받고 멀티캐스트, 클라이언트의 선택, 코디네이터/그룹 등 세 가지 형태의 방

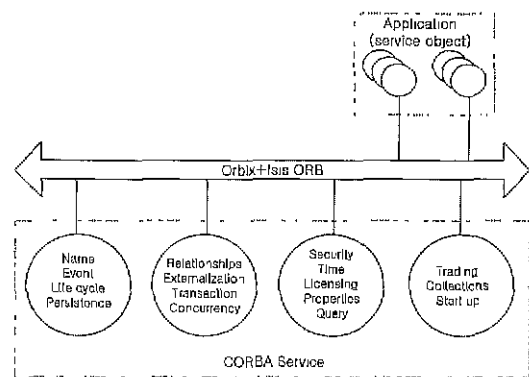


그림 2 Orbix+Isis 구조

식으로 클라이언트와 통신한다.

Active Replica 클래스에는 객체 그룹을 동적으로 관리하기 위하여 그룹 참여, 탈퇴, 모니터링 기능, 상태 전달 서비스가 정의되어 있다. 객체들은 서비스 제공을 원할 경우 언제든지 그룹에 참여할 수 있으며, 장애가 발생하면 그

룹에서 삭제되고 시스템 재부팅 혹은 장애복구 후에 그룹에 참여할 수 있다. 새로운 객체의 그룹 참여나 멤버 객체의 탈퇴 등으로 그룹 멤버집에 변화가 생겼을 경우 멤버 객체들이 동일한 뷰를 유지할 수 있도록 `_newMember` 함수와 `_memberLeft` 함수 같은 모니터링 기능을 제공한다. 또한 멤버 객체들 간 내부 상태 정보 교환을 위하여, Orbix+Isis는 그룹 멤버들의 `_sendState` 함수와 그룹에 참여를 원하는 새로운 서비스 객체의 `_receiveState` 함수를 호출하여 Orbix Request 객체에 있는 스트림 상태 정보(객체의 내부 상태 정보)를 주고받는다.

Event Stream 클래스는 이벤트(클라이언트가 보낸 메시지)라는 단방향 오퍼레이션, 즉 응답을 요구하지 않는 오퍼레이션을 제공한다. 서비스 객체들은 이벤트를 받기 위해 Event Stream 객체 그룹의 멤버(이벤트 리시버)로 참여해야 하고, 자신이 받을 수 있는 이벤트의 수를 초기화 시에 결정해야만 한다. 이벤트 리시버가 됨으로써 서비스 객체들이 얻을 수 있는 장점은 Event Stream 자체가 중복되어 있어 장애 감내성을 보장받을 수 있으며, 이벤트 리시버에 장애가 발생하여 이벤트를 받지 못하더라도 복구된 후 Event Stream에서 제공하는 이벤트 로그를 이용하여 다른 서비스 객체들과 동일한 내부 상태를 유지할 수 있다는 점이다.

3.2 Electra

Electra는 장애 감내성과 그룹 통신 기능을 위해 두 개의 새로운 인터페이스 즉, 객체 그룹을 관리하는 오퍼레이션들을 추가한 BOA 클래스 그리고 호출 형식(invocation style)을 선택하기 위한 Environment 클래스를 정의하였고[7], 투명 멀티캐스트와 불투명 멀티캐스트를 제공한다.

`BOA::create_group` 함수는 새로운 객체 그룹을 하나 생성한다. 객체는 네임 서버를 통해 그룹 레퍼런스를 얻고, 이를 통하여 객체 그룹에 참여(`BOA::join` 함수)하고 떠날(`BOA::leave` 함수) 수 있다. `BOA::destroy_group` 함수는 객체 그룹을 삭제하는 기능을 가지고 있는데, 객체 그룹을 삭제하더라도 객

체 그룹의 멤버들까지 삭제되는 것은 아니다. 새로운 객체가 그룹에 참여하는 경우 모든 멤버들의 동일한 내부 상태 유지를 위하여 Electra는 그룹 멤버들의 `BOA::get_state` 함수를 호출하여 멤버들의 내부 상태를 얻어오며, 참여를 원하는 객체에게 상태를 전달하기 위하여 새로운 객체의 `BOA::set_state` 함수를 호출한다. 또한 그룹 멤버집에 변화가 생길 경우, `BOA::view_change` 함수를 호출하여 항상 최신의 뷰를 갖도록 유지한다.

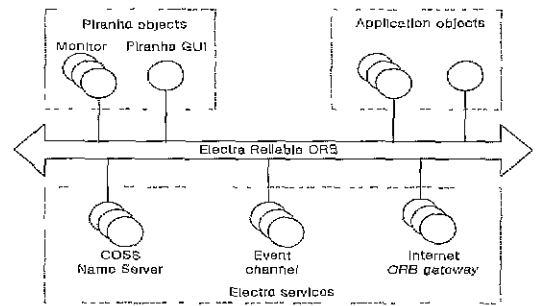


그림 3 Electra 구조

클라이언트 객체가 서비스 객체의 오퍼레이션을 호출하는 경우 CORBA Environment 클래스가 인자로 전달되는데, 일반적으로 Environment 클래스는 서비스 객체에서 클라이언트 객체로 예외 상황을 전달하는데 사용된다. Electra에서는 Environment 클래스에 새로운 기능들을 추가시켜 호출 형식을 선택하도록 하였다. `Environment::call_type` 함수는 클라이언트 객체가 오퍼레이션을 호출하는데 있어서 어떠한 형태로 오퍼레이션을 호출할지를 결정할 수 있게 해준다. 따라서 클라이언트 객체는 오퍼레이션을 동기식(synchronous 또는 blocking), 비동기식(asynchronous 또는 non-blocking), 지연 동기식(deferred synchronous) 형태로 호출할 수 있다. `Environment::num_replies` 함수는 클라이언트 객체가 오퍼레이션을 호출하여 해당 결과를 받을 때 얼마나 많은 수의 결과를 받을 것인가를 결정한다. 상수 ALL은 그룹 내에서 현재 동작중인 모든 멤버들에게 응답을 요구하는 반면, MAJORITY는 그룹 내의 멤버 중 과반수 이상으로부터의 응답을 필요로 한다. COMPARE

는 그룹의 각 멤버들에게 응답을 요구한 후, 각각의 결과를 비교하여 그 중 일치하는 빈도가 가장 높은 결과를 선택한다. 만약 그룹의 각 멤버들에게 얻은 응답이 모두 일치하지 않으면 예외 상황(exception condition)이 발생한다.

4. 서비스(Service) 방식

서비스 방식은 ORB를 수정하는 대신, 장애 감내성 향상을 위해 필요한 기능들을 제공하는 새로운 COS(Common Object Service) 객체를 정의한 것인데, 대표적인 연구로는 GCS(Group Communication Service)[8]와 DOORS(Distributed Object Oriented Reliable System)[9]가 있다.

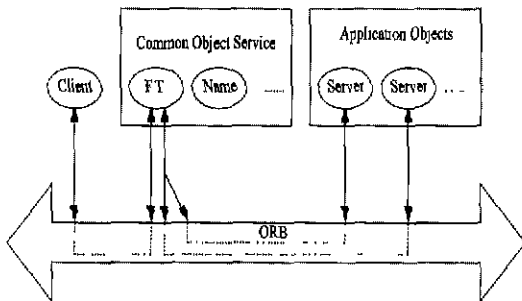


그림 4 서비스 방식

4.1 GCS(Group Communication Service)

GCS는 그림 4의 F.T.에 해당하며 네임(Naming) 서비스나 트랜잭션(Transaction) 서비스처럼 OMG ORB 위에서 CORBA 객체 서비스 형태로 동작하면서 장애 감내성 서비스를 제공한다. 클라이언트는 서비스 객체와 연결을 설정하기 전에 네임 서비스를 통해 GCS와 연결을 설정한다. 설정된 연결을 통하여 클라이언트의 서비스 요청이 GCS를 통하여 객체 그룹에 중계될 수도 있지만, GCS를 이용하여 그룹 멤버에 대한 객체 레퍼런스를 얻어와 클라이언트 객체가 직접 객체 그룹 멤버에게 서비스 요청을 전달할 수도 있다.

GCS의 서비스 구조는 Member 객체,

Service 객체, Factory 객체라는 세 가지 종류의 객체들로 구성되어 있으며, 장애 감내성을 위해 Client 뷰, Member 뷰, Service 뷰라는 서비스 인터페이스를 제공한다[10]. Member 객체는 메시지 전달, 상태 전달, 뷰의 변화 등과 같은 이벤트가 발생할 경우 대응하며, Service 객체는 그룹과의 상호작용을 담당하고, Factory 객체는 그룹의 생성, 즉 Service 객체들의 생성 기능을 담당한다.

Client 뷰는 클라이언트가 그룹에 대한 정보를 얻거나(get-view 함수), 그룹과의 멀티캐스트(multicast 함수), 그룹 멤버와 일대일 통신(deliver 함수)을 할 수 있도록 서비스를 제공한다. Member 뷰는 Client 뷰의 상위 집합으로서 그룹 멤버 객체가 그룹을 탈퇴(leave-group 함수)하거나, 새로운 서비스 객체가 그룹에 참여(join-group 함수)할 수 있도록 서비스를 제공한다. Service 뷰는 그룹 멤버십에 변화가 생겼을 경우 뷰의 변화를 모니터링(view-change 함수)하거나, 새로운 서비스 객체가 그룹에 참여를 원하는 경우 그룹 멤버들간의 동일한 내부 상태 유지를 위하여 상태 전달(get_state 함수, set_state 함수) 서비스 기능을 제공한다.

4.2 DOORS(Distributed Object Oriented Reliable System)

DOORS는 장애 감내성 향상을 위해 DOORS와 DoorMan이라는 두 가지 구성 요소로 이루어져 있으며, 서비스 객체 및 호스트의 장애를 모니터링하고 복구하는 기능을 제공한다[9]. DOORS는 CORBA 기반 환경에서 구현된 시스템으로서 WD(Watch Dog), SWD(Super Watch Dog), RM(Replica Manager)으로 이루어져 있다. WD는 DOORS 시스템 내의 모든 호스트 상에 존재하는 모듈인데, 폴링(polling) 메시지를 통하여 서비스 객체를 모니터링하고 진단(heartbeat) 메시지를 주기적으로 SWD에게 보낸다. 또한 WD는 서비스 객체의 장애 시에 복구를 담당한다. SWD는 WD과는 달리 한 호스트에만 존재하는데 DOORS 시스템 내의 모든 WD와 주기적으로 진단 메시지를 주고받음으로써 각 호

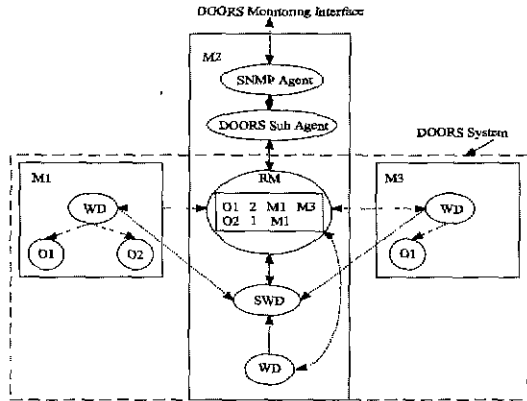


그림 5 DOORS 시스템 구조

스트의 장애 발생 사실을 모니터링 한다. RM 또한 DOORS 시스템 내의 한 호스트 상에만 존재하면서 모든 중복된 객체들을 관리하는 역할을 한다. 서비스 제공을 위해 객체들은 자신과 자신의 요구 사항(중복 정도, 중복 스타일, 위치하고자 하는 호스트 등)을 RM에 테이블 형태로 저장한다. 또한 RM은 DOORS 시스템 전반에 관한 정보(호스트의 상태, 장애 발생 회수, 중복된 각 객체의 상태 등)를 테이블 형태로 관리하고 이를 주기적으로 체크하여 시스템을 관리한다.

DoorMan은 SNMP(Simple Network Management Protocol) 기반의 모듈로서 DOORS를 모니터링하고, RM으로부터 전달받은 정보에 실제 하부 구조에서 동작하고 있는 플랫폼에 대한 정보 등을 추가하여 RM에게 전달함으로써 DOORS 시스템이 효과적인 의사 결정을 가능하게 한다. CORBA 환경 내에 있는 RM과 SNMP 환경에 있는 DoorMan 사이의 정보 전달을 가능하게 하기 위해서는 게이트웨이가 필요한데 그 역할을 하는 것이 DOORS SubAgent이다.

5. 인터셉트 방식 및 이벤트 방식

인터셉트(Intercept) 방식은 ORB를 통해 전달되는 클라이언트 객체의 서비스 요청이 시스템 커널에 의해 전송되기 전에 그룹 툴킷에 의해 가로채어진 후, 이를 다시 그룹 툴킷이

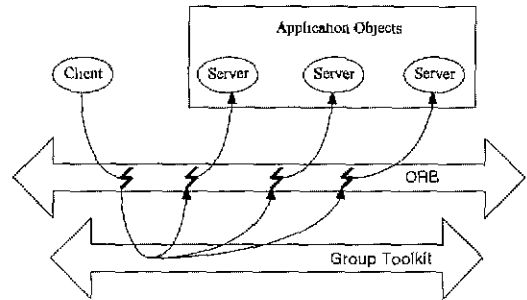


그림 6 인터셉트 방식

ORB에 멀티캐스팅 해주는 방식이다. 인터셉트 방식에서는 통합 방식에서 제공하고 있는 그룹 툴킷을 통한 효율적인 멀티캐스트 및 그룹 관리 기능과 서비스 방식에서 제공하고 있는 ORB간 상호 운용성의 보장이 장점이 된다. Eternal[11]에서는 중복된 서비스 객체들을 관리하기 위하여 인터셉트 방식을 사용하고 있다. Eternal Interceptor는 IIOP(Internet Inter ORB Protocol) 메시지를 가로채고, Eternal Replication Manager는 이 메시지를 Totem[12] 그룹 툴킷에 매핑 시켜주는 역할을 한다.

CORBA 이벤트(Event) 서비스에서 공급자는 이벤트 데이터를 생산하고 소비자는 생성된 이벤트 데이터를 처리하며, 이벤트 채널은 공급자와 소비자 사이에 존재하는 중간 객체로 다수의 공급자와 다수의 소비자간에 비동기 통신을 가능하게 한다. 이벤트 서비스는 push 모드와 pull 모드라는 두 가지 형태의 통신 모드를 제공한다. Push 모드에서는 공급자가 소비자에게 이벤트 데이터를 전달하고 초기화를 담당하며, pull 모드에서는 소비자가 공급자에게

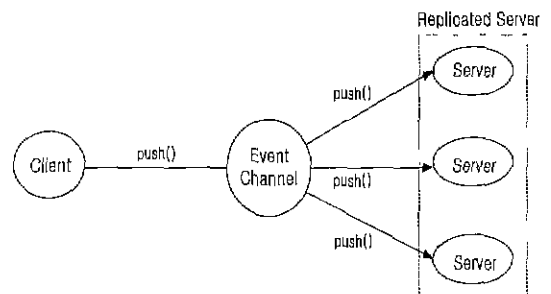


그림 7 중복된 서비스 객체를 위한 이벤트 서비스

이벤트 데이터를 요청하고 초기화를 담당한다. 중복된 서비스 객체들과 클라이언트 객체 사이의 이벤트 서비스를 이용하여 멀티캐스트를 지원하며, 역 방향 이벤트 채널을 이용하여 양방향 통신을 가능하게 한다.

6. 장애 감내형 다중화 서버

이 절에서는 충남대학교에서 개발한 디지털 도서관[2]에 적용된 FTM(Fault Tolerance Manager)[13]을 소개한다. FTM은 디지털 도서관 기능 중 핸들(handle) 서버의 장애 감내성을 위해 구현되었다. 핸들 서버란 디지털 도서관의 전자 문서 식별자와 문서의 위치 정보를 관리하면서 문서 식별자를 저장 위치로 변환시키는 네임 서버이다. 핸들 서버 객체에 장애가 발생할 경우 모든 문서에 액세스하지 못하게 되므로 장애 감내성은 대단히 중요하다.

FTM은 중복된 핸들 서버들을 그룹으로 묶어 관리하면서 모든 멤버들이 동일한 내부 상태를 유지하게 하고, 클라이언트 객체에게 투명하게 객체 그룹을 단일 서비스 객체처럼 보여지게 한다. 클라이언트 객체는 FTM의 존재 여부를 모르고 CORBA 네임 서비스를 통해 핸들 서버의 레퍼런스를 얻으려고 시도한다. 네임 서비스에는 핸들 서버용 FTM의 객체 레퍼런스가 등록되어 있어서 결과적으로 클라이언트 객체는 FTM과 연결이 설정된다. 클라이언트 객체는 어떠한 코드의 수정도 하지 않

므로 투명성을 보장받는다.

그림 8은 장애 감내형 핸들 서비스를 제공하기 위한 FTM의 구조를 나타내고 있다. FTM은 핸들 서버 객체 그룹을 관리하며 핸들 서버들에게 그룹을 생성, 삭제, 참여, 탈퇴할 수 있는 기능을 제공한다. 그룹을 관리하기 위하여 자신의 로컬 저장 장치에 사용자 서비스 요청 로그와 객체 그룹 레퍼런스 테이블을 가지고 있다. 또한 FTM은 클라이언트 객체의 서비스 요구를 핸들 서버에게 중계하고 장애 발생 여부를 모니터링 한다. 예비 FTM은 FTM의 대기(stand-by) 객체로서 주기적으로 FTM이 가지고 있는 사용자 서비스 요청 로그와 객체 그룹 레퍼런스 테이블을 받아 항상 최신의 정보를 유지하며, 모니터링 기능을 수행하여 FTM에 장애가 발생할 경우 FTM의 역할을 수행한다. FTM은 각각의 핸들 서버 객체로부터 전달받은 결과들 중 하나를 택하여 클라이언트 객체에게 전달하는 기능을 수행하는데 결과의 신뢰성을 높이기 위해 일치하는 빈도가 가장 높은 결과를 클라이언트 객체에게 전달한다(majority voting).

7. OMG의 표준화

장애 감내형 분산 어플리케이션 구현 환경을 제공하는 앞의 여러 방식들은 각기 장단점을 가지고 있다.

Orbix + Isis, Electra와 같은 통합 방식은 툴킷에 종속적인 단점을 가지고 있다. 따라서 새로운 툴킷에 동작시키기 위해서는 툴킷에 종속적인 Adaptor 객체를 개발해야만 한다. 그리고 ORB를 변형하므로 표준 CORBA 환경과의 상호 운용성(interoperability)과 이식성(portability) 문제가 지적되고 있다. DOORS, GCS, FTM과 같은 서비스 방식의 연구들은 ORB에 수정을 가하는 대신 새로운 형태의 객체 서비스를 정의하고 있기 때문에 통합 방식의 단점인 상호 운용성이 해결되며, OMG의 기본 원칙, 즉 ORB는 구현 언어나 물리적인 주소에 상관없이 상이하게 분산되어 있는 객체들의 상호 연동을 위해 최소한의 메커니즘만을 제공하며 기타 기능은 COS 형태로 정의한다는

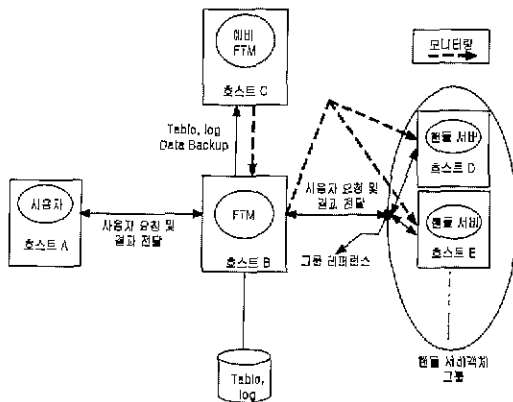


그림 8 FTM 구조 및 동작 방식

제안기관	제출일	문서	방식
Ericsson, IONA, Nortel	1998.10.20	ftp://ftp.omg.org/pub/docs/orbos/98-10-10	Enhancement IIOF (optionally : enhancement IIOF+Service)
Oracle	1998.10.20	ftp://ftp.omg.org/pub/docs/orbos/98-10-13	Integration
Objective Interface Systems	1998.10.20	ftp://ftp.omg.org/pub/docs/orbos/98-10-03	Intercept
Highlander Comm., TIBCO, Inprise Corporation, Lockheed Martin, Lucent Technologies	1998.10.20	ftp://ftp.omg.org/pub/docs/orbos/98-10-09	Service+Integration
Eternal Systems, Sun Microsystems	1998.10.19	ftp://ftp.omg.org/pub/docs/orbos/98-10-08	Service or Service+Integration

것에 부합된다. 그러나 그룹 툴킷을 사용하지 않아 통합 방식보다 고성능 멀티캐스트를 지원하지 못한다. 인터셉트 방식은 통합 방식과 서비스 방식의 장점들만을 제공하지만, 그룹 툴킷과 시스템 운영체제에 종속적인 단점을 가지고 있다. 이벤트 방식은 새로운 COS 서비스의 정의나 그룹 툴킷의 사용 등을 지양하고 대신 이미 정의된 CORBA 이벤트 서비스를 이용하려 했지만, 중복 객체의 동적 관리에 한계가 있어서 크게 주목받고 있지 못하다.

현재 OMG에서는 장애 감내성에 대한 표준을 제정하기 위해 지금까지 제출된 여러 RFP (Request For Proposal)들을 검토하여 그 중 다섯 개의 안을 위의 표와 같이 선별하였다. OMG의 ORB/Object Services Platform Task Force에서는 투표를 통해 이들 중 하나를 선정하여 OMG의 장애 감내성에 대한 표준으로 정할 계획이다.

참고 문헌

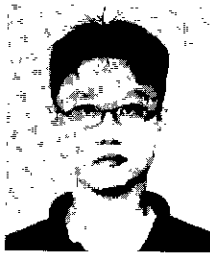
[1] OMG, "The Common Object Request Broker : Architecture and Specification," Inc. Publications, July 1995.
 [2] 신동욱, 이철훈, 최 훈, 디지털 도서관을 위한 분산 플랫폼 구축에 관한 연구, 충남대학교 소프트웨어연구센터, 최종연구 보고서, 1999.

[3] S. Landis, S. Maffeis, "Building reliable distributed system with CORBA," ftp://ftp.softwired.ch/people/maffeis/electra/reliable-corba.ps.gz. April 1997.
 [4] S. Maffeis, C. Schmidt, "Constructing reliable distributed communication systems with CORBA," IEEE Communications, Vol.14, No.2, February 1997.
 [5] K. P. Birman, T. A. Joseph, "Reliable communication in the presence of failures," ACM Trans. Computer Systems, Vol.5, No.1, pp.47-76, 1987.
 [6] IONA Technologies Ltd., ISIS Distributed Systems Inc, "An introduction to Orbix+ISIS," July 1994.
 [7] S. Maffeis, "Piranha-A CORBA tool for high availability," IEEE Computer, Vol.30, No.4, pp.59-66, April 1997.
 [8] P. Felber, B. Gabinato, R. Guerraoui, "Towards Reliable CORBA Integration vs. Service Approach," Special Issues in Object-Oriented Programming, Springer-Verlag, pp.199-205, 1997.
 [9] J. Schonwalder, S. Garg, Y. Huang, P.A. van Moorsel, S. Yajnik, "A Management Interface for Distributed

Fault Tolerance CORBA Services”,
 Proceedings of the Third IEEE International Workshop on Systems Management, Newport, RI, USA, April 1998.

- [10] P. Felber, B. Garbinato, R. Guerraoui, “The Design of a CORBA Group Communication Service,” Proceedings of the 15th Symposium on Reliable Distributed Systems, 1996.
- [11] P. Narasimhan, L. E. Moser, P. M. Melliar-Smith, “Exploiting the internet inter ORB protocol to provide CORBA with fault tolerance,” USENIX COOTS, Portland, pp.81-90, 1997.
- [12] L. E. Moser, P. M. Melliar-Smith, D.A. Agarwal, R. K. Budhia, C. A. Lingley-Papadopoulos, “Totem : A fault tolerant multicast group communication system,” Communications of the ACM, Vol.39, No.4, pp.54-63, 1996.
- [13] 김기영, 최훈, “장애 감내형 디지털 도서관 서비스 구현,” 한국정보과학회 봄 학


술 발표 논문집(A), Vol.26, No.1, pp. 423-425, 1999.



김 기 영

1998 충남대학교 컴퓨터공학교육과
 학사
 1998~현재 충남대학교 컴퓨터공
 학과 석사과정 중
 관심분야 : 분산시스템, 객체 지향
 시스템, CORBA, Fault
 Tolerant CORBA, 자
 바, 네트워크 프로그래
 밅, UNIX 시스템 프로그래
 밅

E-mail : kimky@comeng.chungnam.ac.kr



최 훈

1983 서울대학교 컴퓨터공학과
 학사
 1983~1996 한국전자통신연구원
 광대역통신망연구부
 근무
 1990 Duke University 전산학과
 석사
 1993 Duke University 전산학과
 박사
 1996~현재 충남대학교 컴퓨터공
 학과 조교수
 관심분야 : 분산 시스템, 이동 통신,
 Fault-tolerant 시스템,
 Stochastic Petri Nets

E-mail : hchoi@comeng.chungnam.ac.kr

● 제26회 정기총회 및 추계학술발표회 ●

- 일 자 : 1999년 10월 22일(금)~23일(토)
- 장 소 : 광운대학교
- 논문 접수마감 : 1999년 8월 21일(토)
- 문의 및 접수처 : 한국정보과학회 사무국

Tel. 02-588-9246, Fax. 02-521-1352

http://kiss.or.kr, E-mail:kiss@kiss.or.kr

서울시 서초구 방배3동 984-1(머리재빌딩) ☎ 137-063