

분산 환경을 위한 이벤트 채널 서비스의 설계 및 구현

안형근 한영태 민덕기
건국대학교 컴퓨터·정보통신 공학과
{h96ahg, ythan, dkmin}@konkuk.ac.kr

Design and Implementation of Event Channel Service for Distributed Environment

Hyunggeun An Youngtae Han Dugki Min
Department of Computer Science and Engineering, Konkuk University

요 약

고전적인 Client/Server 환경에서의 통신은 Lan 환경하의 동기적 일대일 통신 기반 이었다. 그러나 인터넷이 확산되고 일반화 되면서 많은 소프트웨어들은 Cluster 또는 Grid 같은 분산시스템 환경과 핸드폰, PDA 같은 Mobile 환경에서 개발되고 있다. 이러한 환경은 WAN 기반의 비동기적 일대다 다대다 통신을 기반으로 한다. 본 논문은 이러한 환경에 적합한 애플리케이션에 독립적이고 비동기 통신을 제공하며 각 통신요소(Event Supplier, Consumer)를 그룹으로 묶어 동적으로 관리해주는 기능을 제공하는 Event Channel Service를 제안한다. 이 Event Channel Service는 Event Channel을 사용하여 Event Supplier와 Consumer를 분리하며 Push, Pull 방식으로 Event를 전달할 수 있는 구조를 제공한다. 또한 다양한 Protocol을 지원하고 Quality of Service를 제공한다.

1. 서론

현재 개발되는 많은 소프트웨어들의 개발 환경은 모바일 또는 대형 분산 시스템 환경의 미들웨어 기반이다. 이러한 환경에서 소프트웨어들의 개발을 간단히 하고 발생하는 Event들을 효율적으로 처리하기 위해서는 Supplier, Consumer 기반의 비동기 통신을 제공하는 Event Channel Service가 필요하다.

이러한 Event Channel Service는 몇 가지 요구사항을 만족해야 한다. 첫째 Event Channel Service는 Application에 독립적이고 Event 기반의 통신을 제공해야 한다. 둘째 Supplier와 Consumer를 그룹으로 묶어 동적으로 관리 할 수 있어야 한다. 셋째 Pull, Push 통신 모델을 지원해야 한다. 넷째 비동기 통신을 제공해야 한다.

본 논문은 위에서 언급한 요구사항들을 만족하고 다양한 Protocol을 지원하는 Event Channel Service를 제안하며 초기 성능테스트 결과를 제시한다. 본 논문에서 제안하는 Event Channel Service는 Quality of Service (QoS) 정책을 지원하며 Filtering을 적용할 수 있는 유연한 구조로 설계되어 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개한다. 3장에서는 본 논문이 제시하는 Event Channel Service의 구조를 설명한다. 4장에서는 구현된 Event Channel Service의 성능결과를 보여주며 마지막장에서는 결론과 향후 추가될 사항으로 마무리 한다.

2. 관련연구

많은 Event Service들은 CORBA 기반과 비 CORBA 기반의 Event Service로 개발되어 왔다[1]. 이 장에서는 그 동안 연구되었던 Event Service들에 대해서 설명한다.

2.1 CORBA Based Event Channel Service

Object Management Group(OMG)에서는 Common Object Request Broker Architecture(CORBA)에서 Event Service를 제시한다[2]. Event Service는 Event Supplier와 Consumer가 Event Channel을 통해서 Event를 주고 받는 Channel 기반의 Event Service이다. CORBA Event Service는 통신 기반으로 Object Request Brokers를 사용 한다.

TAO는 Washington 대학 Computer Science과에서 개발하였다. TAO는 CORBA Event Service를 확장한 Real Time Event Service이다[3]. TAO는 Source, Type 기반의 필터링과 Event Correlation, Real-Time Event 전송을 제공한다.

ORBacus는 Object-Orientated Concepts Inc 에서 CORBA Event Service를 구현한 상용제품이다[4]. C++로 구현되었으며 Typed, Untyped Event와 Push Pull 통신 모델을 지원한다.

jEVENTS는 Outback Resource Group Inc에서 개발하였다 [5]. jEVENTS는 Java로 구현되었으며 Untyped message를 위한 CORBA Event Service이다.

2.2 Non-CORBA Based Event Channel Service

Elvin은 Queensland 대학 Distributed Systems Technology Center에서 개발한 Publish/Subscribe Notification Service이다[6]. Reverse Subscription Mechanism을 이용해 네트워크 트래픽을 감소시키며 Consumer가 관심 있는 Event만을 제공한다.

iBus는 SoftWired사에서 개발한 상용제품이다[7]. 아키텍처는 CORBA Event Service는 유사하다. 비동기 Push 방식과 동기 Pull 방식의 통신을 모두 지원한다. 또한 IIOP bridge를 사용해서 CORBA 애플리케이션에서 사용할 수 있다.

ECO는 Georgia Tech에서 개발한 Event 전달 미들웨어 시스템이다.[8]. 이 모델은 익명으로 하는 그룹 통신 구조로 구현되어 있다. Event는 통신 규칙에 따라 전송되며 Event Sender와 Receiver는 Channel로 매칭 된다.

3. 제시하는 Event Channel Service의 구조 및 구현

3.1 Event Channel Service Model

본 논문에서 제시하는 Event Channel Service는 Event Channel을 통해 Event Supplier와 Consumer를 분리시키고 Consumer를 그룹으로 묶어 Event를 전송한다. 또한 Event Supplier와 Consumer가 서로 분리 되어 있으므로 Supplier는 Consumer가 누구인지, 어디에 있는지, Consumer가 존재하는 지를 몰라도 Event를 전송할 수 있으며 Consumer도 Supplier가 누구인지 모르고 어디에 있는지 몰라도 Event를 전송 받을

수 있다.

Supplier는 Event를 Channel로 전송하기 위해서는 Channel에 등록해야 한다. Supplier가 등록되면 Channel은 Supplier를 위한 Proxy를 생성한다. Consumer도 Event Channel로부터 Event를 받기 위해서는 Event Channel에 등록해야 하며 Channel에 등록되면 Proxy가 생성된다. Supplier는 능동적으로 Event를 생산해서 Event Channel Service로 Push하거나 수동적으로 Event Channel Service가 Pull 요청 할 때까지 기다린다. Consumer (Consumer)는 능동적으로 Event Channel Service에 Pull 요청을 하거나 수동적으로 Event Channel Service가 Event를 Push 할 때까지 기다린다. 본 논문의 Event Channel Service 모델은 그림 1과 같다.

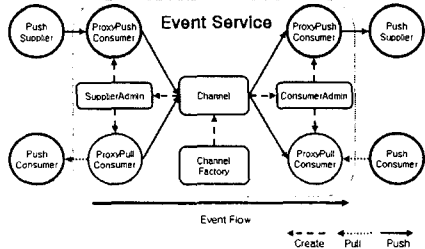


그림 1 Event Channel Service 모델

3.2 Event 전달 방법

Event Channel Service는 Event Channel을 통해 Push-Push, Pull-Pull, Push-Pull, Pull-Push 4가지 방법으로 Supplier와 Consumer간에 Event를 전달한다. 그림 2는 4가지 Event Communication 방법을 보여준다.

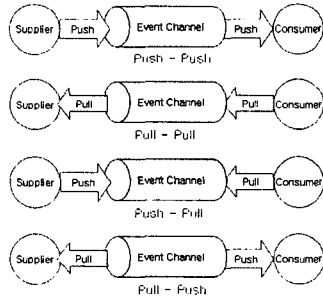


그림 2 Event Communication 방법

3.3 Quality of Service

본 논문의 Event Channel Service는 QoS(Quality of Service) Service 정책을 지원한다. 표 1은 본 논문의 Event Channel Service에서 지원되는 QoS 속성을 보여준다.

속성	Event	Proxy	Admin	Channel
Priority	X	X	X	X
OrderPolicy		X	X	X
DiscardPolicy		X	X	X
MaximumBatchSize		X	X	X
PacingInterval		X	X	X

표 1 QoS 속성

QoS는 계층적인 구조를 가진다. Supplier, Consumer Admin 객체는 Channel의 QoS를 상속받는다. Supplier, Consumer Admin이 Proxy 생성시 생성하는 Proxy에 대한 QoS를 제공받지 못한다면 Channel에서 상속받은 QoS를 생성하는 Proxy의 기본 QoS로 설정한다.

MaximumBatchSize, PacingInterval은 Sequence-Style Mode로 Event를 전송할 경우에만 사용된다. PacingInterval

은 Batch로 Event를 전송하는 간격이다. DiscardPolicy는 Proxy에 새로운 Event가 도달했을 때 Channel의 Queue나 Proxy의 Buffer가 가득찬 경우에 어떤 순서로 Event를 무시할지 결정한다. OrderPolicy는 어떠한 순서로 Event를 전송할지 결정한다. 표 2는 OrderPolicy, DiscardPolicy에 올 수 있는 속성값을 보여준다.

순서	설명
ANY	어떠한 순서이던지 상관없다.
FIFO	먼저 받은 Event 순으로 무시된다.
LIFO	나중에 받은 Event 순으로 무시된다.
PRIORITY	우선순위가 낮은 Event 순으로 무시된다.
DEADLINE	Event의 유효기간에 가까운 Event 순으로 무시된다.

표 2 OrderPolicy, DiscardPolicy 속성값

3.4 Event Channel Service Design

그림 3은 Event Channel Service의 객체 디자인이다.

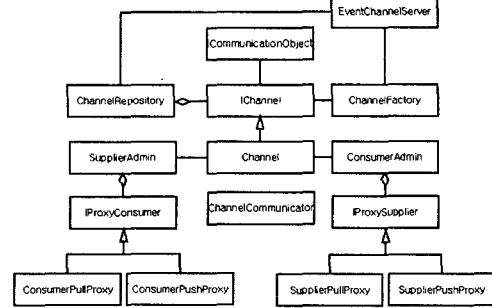


그림 3 Event Channel 객체 Design

Event Channel Service는 EventChannelServer객체를 실행함으로써 시작된다. EventChannelServer는 실행과 동시에 Configuration 정보를 읽어 이전에 만든 Channel이 존재하면 생성하고 Channel을 관리하기 위한 명령어를 기다린다.

Channel은 Channel Factory를 통해서 생성된다. Channel Factory는 Channel QoS와 Management Parameter를 가지고 Channel을 생성한다. Channel은 Management Parameter에서 제공하는 Queue Length Parameter가 정하는 크기의 Queue를 가지고 있다.

Supplier, Consumer Admin은 Channel이 생성됨과 동시에 생성된다. Admin은 오직 Proxy만을 관리하며 Event는 Admin 객체를 통과하지 않는다. Supplier, Consumer Admin은 Proxy QoS와 Management Parameter를 가지고 Proxy를 생성한다. Proxy의 QoS가 제공되지 않는다면 Admin객체의 Default QoS를 사용한다. Proxy는 QoS를 지원하기 위해 내부에 Buffer를 가지고 있다.

ICommunicationObject 인터페이스는 Event Channel Service에 Communication Infrastructure를 제공한다. Channel은 ICommunicationObject를 사용해서 Network를 통해 Supplier에게서 Event를 받고 Consumer에게 Event를 전송한다.

Event는 Event Header와 Event Body를 가지는 Structured Event 구조이다. StructuredEvent 객체는 전송 시 XML로 변환되어 전송된다. Channel은 Communication Infrastructure에서 받은 StructuredEvent의 Header에 있는 도메인과 ID를 보고 Event를 Supplier가 등록한 IProxyConsumer객체를 구현한 객체에 전달한다.

3.5 Communication Infrastructure Framework

본 논문의 Event Channel Service는 통신단에서 다양한 Protocol(TCP, UDP, RMI)을 사용할 수 있다. 그림 4는 Event Channel Service에서 사용한 Communication Infrastructure의