

『 **객체지향에 대해 심도있게 공부한 개발자라면 한 번 들어 본 적이 있는 POSA 시리즈 중 하나인 『Pattern Oriented Software Architecture Volume 2(Patters for Concurrent and Networked Object)』이다.**

POSA 2의 가치는 바로 현실성에 기반한 책이라는 것이다. ACE(Adaptive Communication Environment), TAO(The ACE ORB) 실제 구현물의 내부 소스코드를 파헤쳐가며 전해주는 강렬한 메시지는 개념적으로 분산 객체의 구조를 말하는 책과는 확연히 다른 것이다. 저자는 ACE 프레임워크, OMG CORBA의 스펙을 가장 잘 준수하는 TAO의 창시자인 더글라스 슈미트이다.

이 책은 분산 객체의 주제를 분산 서비스 접근과 설정에 관한 패턴들, 다양한 디스패처에 관한 패턴들, 분산 동기화에 관련된 패턴들, 서비스의 동시성을 보장하기 위한 패턴들과 같이 크게 네 가지로 범주로 나누어 설명하고 있다. 소개된 패턴 하나하나마다 분산 객체의 중요한 이슈를 객체라는 것을 이용해 예술적으로 해결하는 방법을 보여준다. 분산 객체의 골치 아픈 문제를 어떤 패턴으로 풀어나가는지 몇 가지 알아보자.

먼저 분산 객체에서 위치 투명성(Location Transparency)을 제공하는 방법을 보자. 분산 객체를 호출하는 소비자가 쉽게 분산 서비스를 사용하기 위해서는 복잡한 부분들은 제거하고 쉽게 사용하는 인터페이스를 소비자에게 제공하는 것이 중요하다. POSA 2에서는 GoF 디자인 패턴의 Fa^made 패턴을 확장한 Wrapper Fa^made 패턴을 이용하여 깔끔하게 해결하는 방법을 설명한다.

'새로운 서비스를 추가하기 위해 잘 돌아가는 현재의 서비스에 영향을 미치지 않기

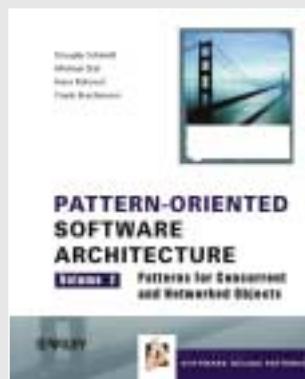
위해서는 어떻게 구조를 잡아야 할까? 서비스를 실행시키는 Dispatcher(변하지 않는 것)와 서비스인 Handler(자주 변하는 것)를 어떻게 분리시킬 것인가? 거기다 성능의 향상까지 얻을 수는 없을까?'라는 물음에 대해 POSA 2는 다음과 같이 방향을 제시한다. 우리가 흔히 사용하고 있는 멀티쓰레드 기반의 폴링 방식은 많은 과부화를 발생시킨다. 이를 해결하기 위해서는 객체지향 5원칙 중 하나인 DIP(Dependency Inversion Principle)를 활용하여 서버와 클라이언트 간의 주고받는 메시지 양을 줄이고, GoF의 템플릿 패턴을 이용하여 새로운 서비스를 쉽게 추가할 수 있는 리액터(Reactor)라는 마법 같은 패턴을 알려준다.

실제 네트워크 환경에서는 하나의 데이터를 전송하기 위해서 여러 개의 복사본을 만들어 전송한다. 그렇다면 서버에 복사본을 포함한 동일한 메시지가 여러 개 도착하면 어떻게 처리해야 할까? 비동기 메시지는 언제 도착할지 모르는데 항상 기다릴 수도 없다. POSA 2에서는 UUID(Universally Unique IDentifier)라는 이러한 문제를 간단히 해결하는 방법을 Asynchronous Completion Token 패턴으로 설명하고 있다. 이외에도 네트워크/분산 객체 프로그램 설계에 있어 골머리를 썩게 하는 많은 문제들을 명쾌하게 설명하고 있다.

시시각각 변하는 고객의 요구사항을 잘 반영하기 위해 유연한 아키텍처를 설계하는 것도 중요하지만 분산 객체이고 네트워크 기반이기 때문에 발생할 수밖에 없는 본질적인 문제의 해결책을 객체지향 원칙들과 패턴들의 조합으로 승화시킨 책이다. 필자는 주저없이 POSA 2를 객체분야에서 몇 안 되는 '살아있는 책'이라고 말하고 싶다.

화제의 해외서적

Pattern Oriented Software Architecture Volume 2



더글라스 슈미트 저/WILEY
출판사/2003년 10월 발행



손영수 only2u4u@devpia.com | 패턴과 분산 객체에 많은 관심을 가지고 있으며 진정한 아키텍트를 꿈꾸는 개발자다. 현재 데브피아 Architecture & Design 시설을 역임중이며, PLOP와 같은 세계적인 패턴워크를 국내에서도 만들기 위해 노력하고 있다.