Rational Software Architect
2004년 6월

서비스 지향 아키텍처(SOA)의 보안 관심사항 모델링

요약
여러 기업들은 웹 서비스를 사용하여 서비스 지향 아키텍처(SOA)를 구현하고, 모델 기반 아키텍처(MDA)의 원칙에 따라 이러한 서비스를 디자인하고 있습니다. MDA를 표현하는 데 사용되는 UML에 비즈니스 프로세스의 보안 필요성을 나타내는 모델 요소가 없으므로 시스템 설계자는 모델에서 보안 관심사항을 무시하거나 구현별 방식을 선택하게 됩니다. 이 문서에서는 비즈니스 이해 당사자와 함께 작업할 때 비즈니스 요구사항을 파악하기 위해 비즈니스 사용자 및 소프트웨어 설계자가 UML 요소에 적용할 수 있는 스테레오타입으로 보안 관련 목적 요소를 표시하는 UML의 프로파일 후보를 제안합니다. 제안된 프로파일을 사용하면 설계자는 상위 레벨 동작 모델의 구현별 세부사항에 대한 MDA 금지사항을 위반하지 않고 디자인 시 비즈니스 보안 목적을 지정할 수 있습니다.

Simon Johnston
설계자
IBM 소프트웨어 그룹

목차

소개

서비스 지향 아키텍처(SOA)는 공개되어 검색 가능한 인터페이스를 통해 응용프로그램 또는 다른 서비스에 서비스를 제공하도록 소프트웨어 시스템을 디자인하는 방법입니다. 각 서비스는 결합력이 약한(일반적으로 비동기) 메시지 기반 통신 모델을 통해 분리된 비즈니스 기능성 묶음을 제공합니다. SOA를 활용하는 시스템 설계자는 응용프로그램에 하나 이상의 서비스를 컴포넌트로 통합할 수 있습니다.

지금까지 대부분의 소프트웨어 산업은 웹 서비스 및 해당 상호작용을 구현하기 위한 기본 기술에 집중해 왔습니다. 웹 서비스를 사용한 기업 수준의 소프트웨어 솔루션 설계에 필요한 기법과 도구에는 거의 관심을 갖지 않았습니다. 다른 복잡한 구조와 마찬가지로, 고품질 소프트웨어 솔루션의 디자인은 잘 알려진 디자인 기법, 구조 패턴 및 스타일에서 지원하는 아키텍처를 조기에 결정한 결과입니다. 이러한 패턴은 확장성, 신뢰성 및 보안과 같은 일반 서비스 문제를 다룹니다.[1]

비즈니스 이해 당사자는 자신들의 비즈니스 요구사항에 해결 방법을 제공하는 IT 조직을 신뢰합니다. 재정적 이유와 시장 중심의 이유로 인해 이해 당사자는 IT 솔루션을 전달하는 데 걸리는 시간과 자금 투자를 축소하고자 합니다. 동시에 각 소프트웨어 프로젝트가 제공하는 요구사항 적용 범위를 최대한

확대하여 IT 솔루션에서 파생되는 가치를 늘리려고 합니다.

현재 이러한 대부분의 프로젝트에는 웹 서비스가 필요하므로 SOA를 사용하여 비즈니스 요구사항을 신속하고 효과적으로 구현하려면 우수한 도구와 기법이 꼭 필요합니다. 모델링은 관심사항을 구분하여[2] 관심사항에 대한 단일화된 관점을 제공하는 기능이 있으므로 특히 중요합니다. 많은 응용프로그램들은 조직의 경계선을 구분하지 않고 작동하므로 서비스 구현의 보안이 주요 관심사항입니다. 이 문서에서는 비즈니스 이해 당사자가 요구사항 프로세스에서 보안 목적을 지정할 수 있도록 기본 모델링 요소들을 제공합니다.

아키텍처 모델과 구현 모델

IT 전문가가 웹 서비스를 사용하여 성급하게 응용프로그램을 신속하게 전달하려다 보면 아키텍처 모델(SOA) 및 구현 모델(웹 서비스)을 동시에 빨리 진행하는 경우가 있습니다. 이러한 상황에서 예상할 수 있듯이 모델과 구현을 구별하기 어려워집니다. 이 문서에서는 응용프로그램의 아키텍처 및 동작에 대한 플랫폼 공통 모델과 모델링된 동작을 구현하는 데 사용되는 기술 및 플랫폼의 혼합을 예방하는 모델 기반 아키텍처[3]를 사용하는 것으로 가정합니다. 시스템 설계자는 UML(Unified Modeling Language)[4]의 도메인 특정 언어 또는 프로파일을 채택하여 서비스 도메인의 관심사항을 모델링합니다. 설계자는 이 모델에서 원칙적으로 플랫폼 및 언어 관심사항을 분리해야 하므로 구현별 보안 관심사항도 분리하게 됩니다. 예를 들어, 서비스 및 메시지의 추상적 개념을 포함하는 모델은 메시지에 공용 키 암호화 및 인증을 사용하여 서비스 인증 및 메시지 서명을 구현하는 방법에 대한 세부사항은 포함할 수 없습니다. 포함할 경우 플랫폼 공통 모델에 특정 기술 구현의 세부사항을 도입하므로 가장 기본적인 원칙(관심사항 구분 필요성)을 위반하게 됩니다. 반면 보안은 나중에 고려할 수 있는 사항이 아닙니다. 보안 구현은 복잡하며 성능에 심각한 영향을 미치므로, 서비스를 지원하는 IT 하부 구조에 대한 요구사항이 커지게 합니다. 따라서 보안 관심사항을 다른 관심사항 만큼 신중하게 모델링하는 데 관심이 집중되고 있습니다.

관심사항을 구분하면 IT 조직은 비즈니스 이해 당사자에게 요구사항 적용 범위를 최대한 확대해야 하는 비즈니스 필요성을 쉽게 이해시키고 설명할 수 있습니다. 여기서는 MDA의 원칙을 따르면서 활동 관심사항을 구현 및 플랫폼별 관심사항과 구분하여 상위 레벨 모델에서 보안 목적을 지정하는 방법을 보여줍니다.

보안 관심사항 재검토

B2B 표준의 RosettaNet[5] 제품군을 개발한 팀이 방금 언급된 내용과 유사한 문제를 제기했으며 문제 해결을 위한 활동을 시작했습니다. 목표는 비즈니스 설계자 즉, RosettaNet 팀에서 데이터 및 처리 요구사항을 수집한 주요 이해 당사자에게 간소화된 선택사항을 제시하는 것이었습니다. 비즈니스 설계자는 보안 관심사항의 기술 세부사항에 익숙하지는 않았지만 안전한 방식으로 전달해야 하는 데이터와 보안을 고려하지 않고 보낼 수 있는 데이터는 구별할 수 있었습니다.

그러나 이러한 접근 방식은 간소화된 용어가 치명적일 수 있다는 한 가지 문제점을 안고 있습니다. 용어가 복잡하거나 불명확해지면 비즈니스 이해 당사자는 만일에 대비하여 사용 가능한 모든 종류의 보안을 요청하므로 최적의 디자인이 불가능해집니다. 이해 당사자 측의 특성 때문에, 아키텍처 그룹은 사용자가 이해할 수 있는 설명 및 제한사항의 구체화를 위한 간단한 가이드라인을 만들었습니다. 덕분에 이해 당사자가 비용 편익 절충 결정을 내리는 데 필요한 사전 지식이 제공되었습니다. 예를 들어, RosettaNet 팀은 비즈니스 사용자가 데이터 암호화 비용이 보호할 데이터의 가치보다 훨씬 커지는 시점을 알려주는 예제를 사용했습니다.

보안 문제 일반화

일반 소프트웨어 보안 문제에는 여러 가지 주제가 있으며 특정 보안 구현 및 기술에는 이보다 많은 주제가 있습니다. 그러나 보안 관련 기술 구현을 가능하게 하는 기본 목적에 대해 설명하려고 합니다. 특히 특정 기술 구현을 식별하는 데 사용할 수 있으며 이해하기 쉽게 설명된 기본 목적들을 지정하고자 합니다.

"보안"이라는 용어가 의미하는 기본 문제 및 관심사항에 대해 알아봅니다. 일반적인 예로 ATM 기계에서 현금을 인출하는 경우가 있습니다. 먼저 ATM 기계로 가서 두 가지 사항 즉, 내 ATM 카드(공식 ID 역할을 담당함)와 개인 식별 번호(PIN)를 제공해야 합니다. 카드와 PIN은 나와 은행만 알고 다른 사람은 모르는 "공유 기밀사항"입니다. 이제 ATM은 식별 정보와 PIN을 받고 기계 앞에 있는 사람을 적합한 신뢰 수준의 계좌 보유자로 볼 수 있는지 여부를 은행에 문의합니다. 개설 은행은 제공된 세부사항을 승인하고 계좌 보유자에 대한 추가 정보를 제공하는 보안 신임 세트를 ATM에 다시 보냅니다. 이 계좌별 정보에 따라 ATM은 계좌 보유자로서 수행하도록 권한 부여된 조치 목록을 표시합니다. (예: "출금", "입금" 등과 같은 조치. 이 옵션 세트는 실제로 다음 두 세트의 공통 부분을 표시합니다.)

- 이 ATM에서 수행할 수 있는 모든 오퍼레이션 세트
- 개설 은행이 인증하여 사용자가 수행할 수 있는 모든 오퍼레이션 세트

은행이 보내는 신임에는 일반적으로 1회 거래 시 인출할 수 있는 금액 제한 즉, ATM 자체에서 통제되는 제한이 포함됩니다. ATM 시스템 설계자는 ATM으로 들어오거나 ATM에서 나가는 모든 정보 흐름을 로깅하여 감사 추적을 제공해야 합니다.

그렇다면 은행과 ATM 간의 통신은 어떻게 수행될까요? 은행이 ATM에서 얻은 정보 또는 ATM이 은행에서 얻은 정보를 어떻게 신뢰할 수 있겠습니까? 이와 같은 보안 프로토콜, 데이터 암호화 등과 관련된 기술적인 세부사항은 중요하고 관심을 가질 만합니다. 그러나 이러한 내용이야말로 상위 레벨 동작 모델의 범위에서 벗어나는 세부사항 유형입니다.

ATM 예제는 세 가지 카테고리의 보안 관심사항 또는 도메인을 보여줍니다.

- 사용자 신원 파악 (식별, 인증)

- 수행할 수 있는 오퍼레이션 (권한 부여)
- 각 사용자가 볼 수 있는 내용 (개인 정보 보호)

네 번째 도메인은 약간 애매하지만 다른 세 도메인과 관련되어 있습니다.

- 발생 내용 (감사)

대부분의 IT 응용프로그램 개발 시 감사 도메인은 나중에 관심을 갖는 경향이 있습니다. 반면 핵심 보안과 같은 일부 비즈니스 영역 및 규제 관심사항 측면에서 상당한 감사 능력이 요구되는 EDI(Electronic Data Interchange)와 같은 응용프로그램에서는 감사 기능이 명확하고 중요한 보안 관심사항이 됩니다. 이 접근 방식에서는 감사를 내부적인 목적으로 사용하므로 도입하는 기본사항은 모두 세부 동작의 감사 추적을 포함합니다. 따라서 예를 들어, A 관계자가 B 관계자와 협력하기 전에 B 관계자를 인증하도록 요구한다는 점을 표시할 때 날짜/시간 및 기타 구현 세부사항이 포함된 인증 요청 및 응답을 모두 감사해야 함을 의미합니다.

그림 1은 이러한 도메인 간의 종속성을 나타냅니다. 예를 들어, 인증 없이는 권한 부여를 구현할 수 없습니다. 반면 구현이 아닌 분석 및 부인방지를 위해 모든 예외사항을 캡처하도록 권한 부여 및 인증이 모두 감사에 종속되어 있습니다.

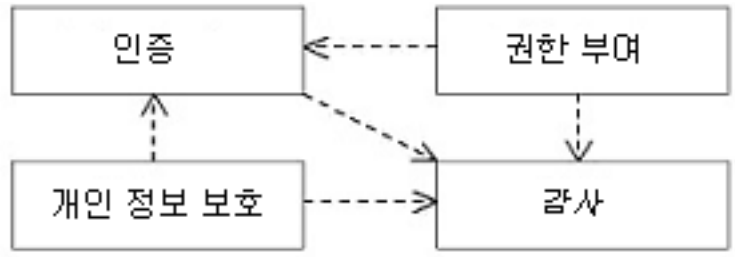

그림 1: 보안 도메인 간 종속성

이 문서에서는 이러한 도메인에 공통적으로 적용되는 기본 목적에 대해 설명하며, 이러한 기본사항이 모델에 도입되어 주어진 기술로 구현이 작동되는 방법에 대해 설명합니다.

사용자 신원 파악

이 도메인은 주로 하나 이상의 통신 또는 협업 관계자 식별과 관련이 있습니다. 실제로 이를 두 가지 고유 관심사항, 즉정적 개념인 식별과 동적 개념인 인증으로 구분할 수 있습니다. ATM 예제에서, ATM 카드는 정적이며 은행이 발행한 ID인 반면, 카드/PIN 쌍은 ATM에서 카드 소유자를 계정 보유자로서 동적으로 인증할 수 있게 해줍니다. 식별보다 인증 개념을 모델링하는 것이 훨씬 중요합니다. 일반적으로 인증에 비해 식별은 구현의 기술적 세부사항과 관련이 깊습니다. 예를 들어, 관계자들 간의 특정 협업 시 관계자들 간의 인증이 필요하다는 점을 확언할 수 있습니다. 이 작업은 명시적 또는 내재적으로

수행할 수 있습니다. 예를 들어, 신뢰와 같은 다른 개념을 사용하여 신뢰 영역을 설명할 수 있습니다. 동일한 신뢰 영역에 속한 관계자들은 서로 인증할 필요가 없다고 가정할 수 있지만 신뢰 경계선을 넘는 통신의 경우 인증이 반드시 필요합니다.

신뢰 영역은 유용한 개념이지만 모든 필요를 충족시키지는 않습니다. 신뢰 영역은 주로 계층 구조로 이루어져 있으며 들어오는 관계자를 인증하지 않아도 외부 관계자와의 일부 통신이 가능합니다. 비즈니스는 직원이 아닌 다른 사람이 방화벽 내의 네트워크 자원에 액세스할 수 없다는 점에서 단일 신뢰 영역으로 간주할 수 있습니다. 그러나 엔터프라이즈 자원 조달 계획(ERP)과 고객 관계 관리(CRM) 응용프로그램을 구분하는 신뢰 영역도 있는 경우에는 각각 자체의 응용프로그램 레벨 보안을 구현하므로 단일 영역 그림은 적합하지 않습니다. 비즈니스는 거래하는 엑스트라넷의 멤버쉽을 통해 자체 비즈니스를 외부 신뢰 영역의 구성원으로 간주할 수도 있습니다. 이러한 점에서 신뢰 및 인증은 협업 모델에 적용할 수 있는 명시적이지만 중복되는 목적입니다.

수행할 수 있는 오퍼레이션

이 도메인에서는 주로 대상을 파악한 후에 수행하도록 권한 부여된 오퍼레이션으로만 옵션을 제한할 수 있도록 보장합니다. 수행할 작업을 결정하기 전에 대상을 파악해야 하므로 인증을 수행하는 기능이 필요합니다. 비즈니스 이해 당사자는 이러한 인증 수행자를 알고 수행된 시기를 아는 방식으로(감사를 통해) 안전하게 수행해야 하는 기능을 식별하고자 합니다. 물론 권한 부여가 필요하지 않은 기능도 분명히 있습니다. 즉, 모든 사람의 액세스를 허용하거나 성능 상의 이유로 서비스에 대한 요청자의 액세스 권한을 안전하게 가정할 수 있는 신뢰 영역을 식별한 경우입니다.

보안 구현에 비용이 상당히 많이 드는 경우도 있으므로 성능을 추가 관심사항(때로는 경쟁적 관심사항)으로 고려할 때 이 신뢰 영역 개념이 중요해집니다. 고객의 미결 주문 수를 리턴하는 기능을 고려해 보십시오. 데이터베이스에서 조회하면 간단하지만, 인증, 권한 부여 및 개인 정보 보호를 요구할 경우에는(아래 참조) 상당한 비용을 감수해야 합니다.

- 요청자에게 신임을 제공하도록 요구해야 합니다.
- 원격 서비스 등을 이용하여 이러한 신임을 확인해야 합니다.
- 리턴된 정보를 암호화해야 합니다.

요청자와 제공자가 모두 동일한 응용프로그램의 서비스임을 파악한 경우 단일 신뢰 영역으로 식별하여 오버헤드를 줄이고 성능을 향상시킬 수 있습니다. 아키텍처 관점에서 신뢰 영역 간의 모든 통신을 파악하는 것도 중요합니다. 이러한 통신은 전체 보안 구현에서 실패(또는 공격) 지점이 될 가능성이 매우 높으므로 가능하면 이러한 통신을 최소한으로 줄이고 제어해야 합니다.

구현 측면에서 두 가지 기본적인 권한 부여 접근 방식에 관심을 가져 볼 만합니다.

- **개별 관계자의 권한 부여** — 모든 관계자에게 기능에 대한 명시적 액세스 권한 세트를 지정할 수 있습니다(그러나 프로세스를 최적화하기 위해 명시적 액세스 권한이 없는 경우 해당 액세스 권한의 내재적 승인 또는 거부로 처리하도록 동의할 수 있음).
- **역할을 통한 권한 부여** — 각 응용프로그램에 대해 다양한 역할을작성할 수 있으며 위의 설명에 따라 개별 관계자가 아닌 역할에 액세스 권한을 지정할 수 있습니다. 각 관계자가 인증을 받으면 관계자에게 제공된 신임에는 관계자의 역할이 포함되어 있으므로 관계자가 특정 기능에 액세스하도록 권한을 받았는지 여부가 결정됩니다.

여기서 주의해야 할 점은 상위 레벨 동작 모델에서는 항상 이런 세부사항을 제외하고자 한다는 점입니다. 예를 들어, 상위 레벨 모델링 단계의 목적은 권한 부여를 수행하는 방법을 자세히 설명하는 것이 아니라, 단순히 주어진 기능을 수행하려면 권한 부여가 필요하다는 점을 지적하는 것입니다.

각 사용자가 볼 수 있는 내용

개인 정보 보호 도메인의 관심사항은 볼 수 있는 권한을 받은 정보만 보고 권한이 없는 다른 관계자는 정보를 보지 못하게 하는 것입니다. 비즈니스는 비즈니스 운영을 지원하기 위해 저장, 조작 및 전달되는 대량의 데이터를 사용하고 생성합니다. 민감한 특성을 가진 정보를 보호하고, 정보와 서비스를 요청 또는 제공하는 사람을 파악하는 방법을 제공해야 합니다. 즉, 단순히 어느 서비스가 요청자 또는 제공자인지 파악하는 대신, 인증을 받은 일반 사용자가 해당 서비스를 이용하는 트랜잭션의 관계자인지를 파악해야 합니다.

개인 정보 보호 도메인에는 두 가지 다른 목적이 있습니다.

- 메시지 또는 문서를 작성한 일반 사용자 또는 서비스를 식별하도록 메시지 또는 문서에 서명하려고 합니다(잠재적으로 여러 서명 사용). 이는 디지털 서명에 대한 여러 가지 공통 표준을 따르며 B2B 거래, 정부 거래 및 기업 전자 우편 통신에서 폭넓게 사용됩니다. 문서에는 여러 개의 서명이 포함될 수 있습니다. 예를 들어, 공급 요청서에는 제안자 및 해당 관리자(승인자)가 서명하고 최종적으로 주문 제출 시 구매 부서가 서명할 수 있습니다. 이들 모든 서명이 문서와 함께 제공됩니다.

- 보안 매체를 통해 메시지를 보내거나, 암호화하거나, 내용에 대한 보안을 설정하여, 메시지에 대한 개인 정보 보호를 보장하려고 합니다. 이 목적은 구현 선택사항이 가장 다양한 영역이라고 할 수 있습니다. 예를 들어, 서비스들 간에 전송되는 디지털 메시지의 경우, 송신자가 메시지 자체를 암호화한 후에 비보안 채널을 통해 보내거나, HTTPS 또는 TLS(둘 다 서비스의 일부로 암호화 기능을 제공하지만 송신자가 제어할 수 없음)와 같은 보안 전송을

통해 일반 텍스트로 메시지를 보내거나, 메시지에 다른 보안 방법(예: 복사 방지 용지에 인쇄 및 운송 기관을 통한 전송)을 통해 보낸 문서의 ID가 포함될 수도 있습니다.

또 다른 주요 관심사항은 데이터 무결성입니다. 메시지 또는 문서의 내용은 다양한 거래 관계자들 사이에서 이동하는 중 변경되면 안됩니다. 데이터 무결성이 있는 메시지 또는 문서를 부인방지 기능이 설정되었다고 합니다. 개인용 메시지의 경우 부인방지 기능이 설정된 것으로 예상됩니다. 그러나 개인 정보 보호 솔루션이 없어도 간단하게 메시지에 부인방지 기능이 설정되었음을 표시할 수 있습니다.

따라서 구현자는 비즈니스에서 설정한 가이드라인과 공통되게 데이터 개인 정보 보호 요구사항을 충족시키는 솔루션을 제공해야 합니다. 예를 들어, 전자상거래 웹 사이트에서 검증할 각 고객의 신용 카드 번호를 은행에 전송할 때 운송 기관을 이용하는 것은 적합하지 않지만 최고 기밀 관리 문서를 전달하는 경우 운송 기관을 이용하는 것이 가장 좋습니다.

부인방지

또 다른 관심사항은 많은 EDI 문서에서 발견되는 출처 및 내용의 부인방지 개념입니다. 부인방지는 추후에 거래 관계자 중 하나가 거래 완료를 거부할 수도 있다는 개념에 대한 큰 명제입니다. 또는 거래 관계자 중 하나가 거래 발생은 인정하지만 해당 거래의 특정 세부사항에 대해 이의를 제기할 수도 있습니다. 예를 들어, 100,000주를 구입한 A 관계자가 나중에 주식의 가치가 하락하자 100주만 거래했다고 주장할 수 있습니다. 이러한 논쟁을 조정하기 위해 많은 산업 및 지역에는 레코드 장기 보유에 대한 법률 규제가 있습니다.

이와 관련하여 이전에 소개한 감사 관심사항은 메시지를 저장하는 데 필요한 기능을 제공합니다. 감사 자체만으로는 부인방지 목적을 충족시킬 수 없지만 메시지 교환 감사(내용 증명)를 인증(출처 증명)과 함께 수행하면 일반적으로 필요한 수준의 증명이 제공됩니다.

모델에 기본사항 적용

다음은 이 문서의 "프로파일 세부사항" 절에 자세히 설명된 프로파일 후보를 사용하여 두 관계자들 간의 문서 교환에 대한 간단한 상위 레벨 모델의 보안 목적에 대해 설명하는 방법의 예제입니다. 다음은 예제에서 사용되는 목적 요소를 요약한 표입니다.

목적	*설명*
감사	지정된 통신을 감사할 예정임을 표시하는 데 사용합니다. 감사에는 인증된 모든 관계자의 신원과 관계자들 간에 전송된 데이터가 모두 포함된다고 가정됩니다.
인증	지정된 협업의 범위에서 인증을 받아야 하는 관계자를 표시하는 데 사용합니다.
권한 부여	두 관계자 간의 통신에서 요청자가 요청을 수행할 권한이 있는지 확인해야 함을 표시하는 데 사용합니다.
개인용	표시된 정보를 개인용 정보로 취급해야 하며 데이터가 개인용 데이터이고(권한 없는 열람 방지) 부인방지 기능이 설정되었음을(수정 없이 목적지에 도착하도록 보장됨) 확인하기 위해 합리적인 모든 노력을 기울이도록 표시하는 데 사용합니다.
서명	표시된 정보에 문서와 관련된 관계자의 디지털 서명을 포함하도록 표시하는 데 사용합니다.
부인방지	관계자들 간에 전송된 데이터를 송신자가 보낼 때와 동일한 형식, 동일한 내용 및 의미로 수신자에게 도착하도록 보장해야 함을 표시하는 데 사용합니다.
신뢰	명시적 신뢰 영역에 참여하는 협업의 관계자 집합을 표시하는 데 사용합니다.

다음은 구매자와 판매자 간의 구매 주문 교환을 보여 주는 UML 활동 모델의 일부분입니다. 이 예제의 목적 요소는 일반 모델 요소에 적용되는 UML 스테레오타입으로 실현됩니다.

그림 2는 이 협업의 세 가지 보안 목적을 보여줍니다.

- 판매자가 구매자를 인증해야 합니다.
- "PO 승인" 조치에 권한 부여가 필요합니다.
- 구매 주문 오브젝트 플로우에 서명하고 일부의 경우 부인방지 기능을 설정해야 합니다.

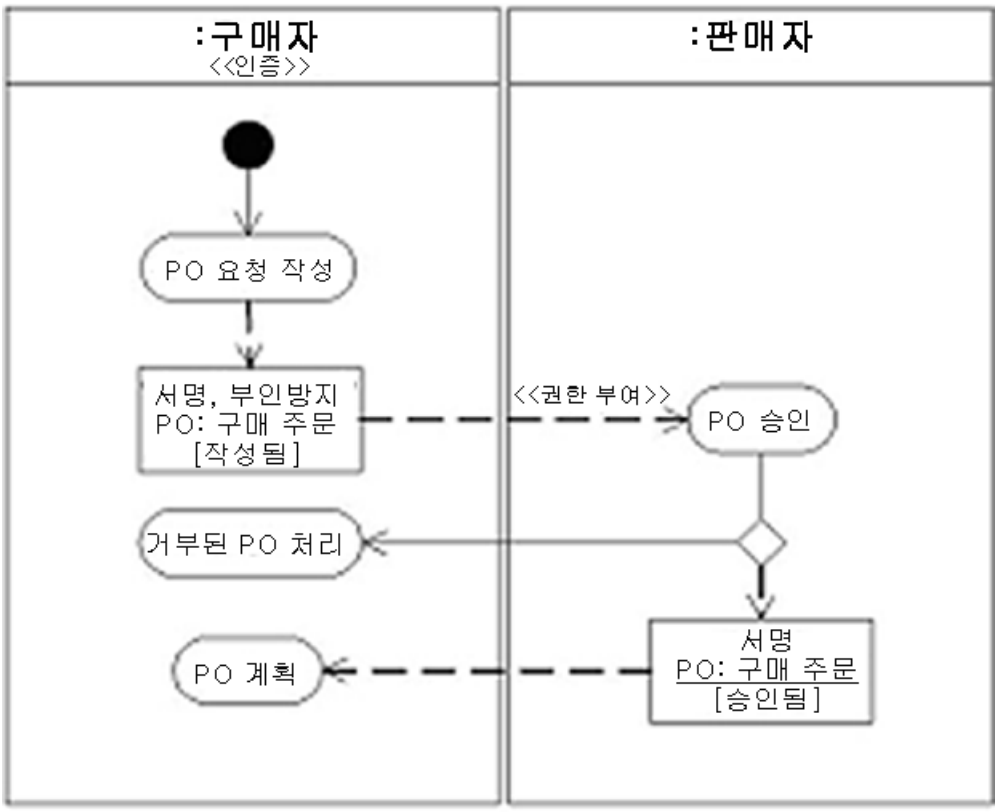

그림 2: 구매 거래의 보안 목적

이 모델에는 확실히 추가 기술 관심사항이 없습니다. 즉, 서비스, 엔드포인트, 인터페이스, 스키마, XML 등에 대한 언급이 없습니다. 이러한 형태는 비즈니스 이해 당사자의 보안 관련 요구사항을 도출하는 수단으로 권장하는 상위 레벨 플랫폼에 공통되는 시점입니다.

이 문서의 "프로파일 세부사항" 절에 제공된 프로파일은 플랫폼 중립 모델을 개발하는 데 채택할 수 있는 모델링 도구 또는 방식에는 거의 제한을 두지 않습니다. 예를 들어, 그림 2에서는 활동 다이어그램을 사용하여 비즈니스 동작을 모델링하지만 그림 4에서는 시퀀스 다이어그램을 사용합니다. 또 협업 다이어그램 및 상태 머신을 사용하여 비즈니스 응용프로그램의 동작을 모델링할 수도 있습니다.

기본사항과 구현의 맵핑 예제

다음 예제에서는 특정 기술 디자인에 도입된 일부 기본사항의 잠재적 맵핑을 보여줍니다. 이 문서의 후반에서는 실제 구현 선택사항에 대해 설명합니다. 다시 MDA 접근 방식에 대해 알아 봅니다.

이 접근 방식에서는 모델 간 변환을 통해 상위 레벨 모델이 구현별 모델로 변환되며 이 경우 아래에 설명된 패턴으로 구체화됩니다.

프로토콜 및 패턴

먼저 기술 구현을 표시하는 방법이 필요합니다. UML에서는 템플리트 방식의 협업을 사용하여 패턴을 표시하고, 이 패턴은 다시 특정 모델 요소에 바인딩하여 추가 세부사항으로 확대될 수 있습니다. 여기에서는 모델의 각 목적에 대한 기술별 구현을 표시하는 패턴을 개발해야 합니다.

그림 3은 신뢰 유효성 검증 서비스를 사용하는 방식의 권한 부여 패턴을 보여 줍니다. IT 그룹에는 여러 가지 구현 또는 성능과 같이 잠재적으로 여러 가지 특성에 대해 선택할 수 있는 패턴 카탈로그가 들어 있습니다. 예제에서 패턴에는 세 가지 매개변수 즉, 요청자 오브젝트, 권한 부여 메소드 및 사용할 유효성 검증 서비스가 사용됨을 알 수 있습니다. 그림 3은 신뢰 영역 표시(제공자 및 유효성 검증자 오브젝트를 묶은 상자) 사용을 보여 줍니다. 이 표시를 통해 권한 부여된 메소드의 제공자가 유효성 검증 서비스를 신뢰하려는 요구사항을 시각적으로 확인할 수 있습니다. 신뢰 영역의 존재를 알면 자주 상호작용하는 두 서비스 간에는 보안을 구현할 필요가 없습니다. 이 경우 성능에 맞게 두 서비스를 최적화합니다.

그림 3: 권한 부여 패턴

이제 패턴에서 구현의 동작을 설명할 수 있습니다. 그림 4는 제공자가 요청자의 신임에 대한 유효성을 검증하도록 인증자에게 요청하고, 인증자의 응답에 따라 메소드를 실행하거나 권한 부여 예외사항을 알리는 방법을 보여 줍니다.

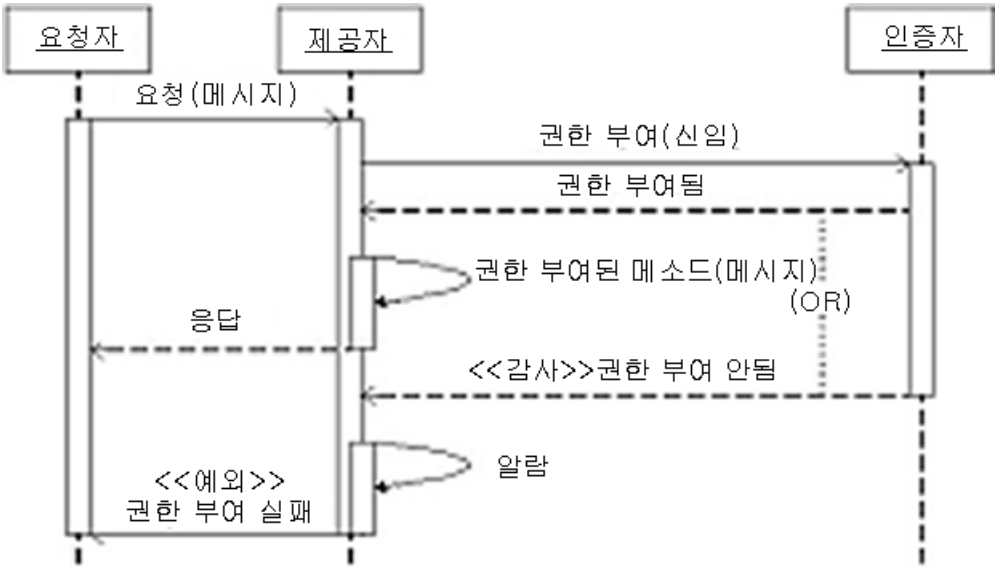

그림 4: 구현 동작

이 예제에서는 그림 3에 정의된 세 관계자들 간의 이벤트 시퀀스를 보여 주는 UML 메시지 시퀀스 다이어그램을 활용합니다. 그림 4에는 추가 스테레오타입이 도입되었으며 특히 권한 부여 실패를 표시하는 유효성 검증 서비스의 모든 메시지 응답에 대해 감사가 수행됩니다.

초반에 대부분의 명시적 목적은 감사 요구사항을 내포하고 있음을 언급했습니다. 그러나 그림 4에서는 audit 스테레오타입을 통해 인증자의 권한 부여 안됨 응답이 명시적으로 표시되었음을 알 수 있습니다. 이유는 이 이벤트를 감사해야 하는 비즈니스 요구사항(내재된 보안 요구사항 이외)이 있기 때문입니다.

스테레오타입 예외는 제안된 프로파일의 일부분이 아니라 코어 UML 스펙의 일부분이라는 점에 유의하십시오. 또한 제한조건 {OR}를 사용하여 단일 다이어그램에서 권한 부여 방식의 가능한 결과가 모두 모델링됨에 유의하십시오.

예제 활동 모델에 패턴을 연결하려면 그림 5와 같이 패턴의 매개변수를 모델의 요소로 대체해야 합니다. 구매자 오브젝트가 요청자이고 PO 승인 조치가 보호 방식이며 XWSKeySvr 서비스가 유효성 검증자인 바인딩을 작성합니다. 그러면 모델에 이 패턴의 인스턴스가 작성됩니다. 그런 다음 동일한 패턴을 플랫폼 중립 모델에서 "PO 승인"과 같은 메시지의 다른 여러 인스턴스에 바인드할 수 있습니다.

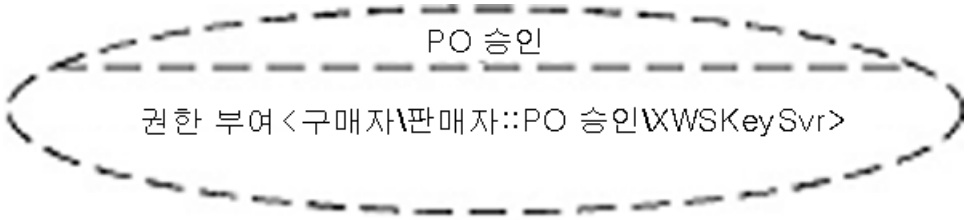

그림 5: 모델에 패턴 바인딩

이 바인딩은 모델에서 계속 유지되므로 나중에 이 바인딩을 조작하여 대체 구현을 작성할 수 있습니다(예를 들어, 한 서비스에서 다른 서비스로 유효성 검증자 변경).

구현 선택사항

권한 부여 또는 개인 정보 보호와 같은 목적을 구현하는 데 사용되는 디자인 패턴을 개발하는 경우, 여러 가지 디자인 접근 방식 및 플랫폼별 기술 솔루션이 있습니다. 예를 들어, 권한 부여를 구현할 때 관계자를 역할과 구분하는 개념은 이미 설명했습니다. 실제로 이 접근 방식은 매우 일반적인 접근 방식이며 J2EE 및 Microsoft .NET 미들웨어에서 모두 볼 수 있습니다. 이 문서에서는 이러한 디자인 결정사항 및 패턴에 대해 자세히 설명하지 않지만, 특정 SOA 패턴이 Gang of Four[6]와 같은 일반 패턴의 변형으로 표시되거나 서비스 지향 응용프로그램 하부 구조의 기술적 제한조건 및 현실에만 해당되는 완전히 새로운 패턴으로 나타날 것으로 예상됩니다.

표준 역시 SOA의 채택 및 SOA의 형태를 정확히 예측하는 데 중요한 역할을 담당합니다. W3C(World Wide Web Consortium) 또는 OASIS(Organization for the Advancement of Structured Information Standards)의 표준과 같은 기본 표준이 기초를 이룹니다. RosettaNet 표준과 같은 산업 표준은 비즈니스를 내부적으로 결속하고 관계자와 연결하는 내용 및 프로세스를 제공합니다. 그러나 너무 낙관하지는 마십시오. 이러한 많은 표준들은 비교적 새로운 표준으로 개발 시 여러 가지 상업적 및 학문적 이해가 표출되었으며 표준을 체계화하기 위해 표준을 관리하는 조직까지 설립되었습니다(WS-I).

다음 표는 웹 서비스 공간에서 보안과 관련된 현재 스펙의 스냅샷을 나타냅니다. 이러한 스펙들 간의 관계 및 이러한 스펙과 기본 W3C 및 OASIS XML 스펙 간의 관계와는 별도로, 이 스펙은 복잡하며 변경될 수 있다는 문제가 있습니다.

XML	*메시지 전달*	*보안*	*관련*
XML	SOAP	XML Encryption	WS-Policy
XML Namespaces	MTOP	WS-Security	WS-Policy Assertions
XML InfoSet	WS-Addressing	WS-SecureConversation	WS-Policy Attachment
XInclude	WS-Routing	WS-Trust	WS-SecurityPolicy
XPath		WS-Federation	XML Query
		Active Requestor Profile	
		Passive Requestor Profile	
		Web Services Security Kerberos Binding	
		Web Services Security Kerberos Binding	

서비스 지향 아키텍처와 웹 서비스 구현은 개방된 통합 기회에서 많은 IT 조직에 상당한 이점을 줄 것입니다. 또한 웹 서비스를 사용한 기존 응용프로그램의 재설계 시도는 신중하게 모델링하고 철저히 이해해야 하며 비즈니스 이해 당사자와의 협업을 통해 모든 중요 관심사항을 다뤄야 할 것입니다.

프로파일 세부사항

이 절에서는 비즈니스 이해 당사자의 요구사항을 캡처하는 UML 요소에 적용할 수 있는 스테레오타입으로 목적 요소를 표시하는 UML의 프로파일 후보에 대해 설명합니다.

audit 스테레오타입

메타 클래스

ActivityNode, Message

설명

지정된 통신을 감사할 예정임을 표시하는 데 사용합니다. 감사에는 인증된 모든 관계자의 신원과 관계자들 간에 전송된

데이터가 모두 포함된다고 가정됩니다.

감사는 "authorize"로 스테레오타입화된 모든 통신에 내포되어 있으며 인증 구현 및 서명 데이터와 개인용 데이터의 예외사항 처리에서 명시됩니다.

ActivityNode에 적용하는 경우 활동 다이어그램에서 조치, 구조화된 활동 및 제어 노드(예: 결정)에 어노테이션을 지정할 수 있습니다. 상호작용의 Message에 적용하는 경우, 표시된 모델 요소들 간에 전송된 메시지에 어노테이션을 지정할 수 있습니다.

특성

없음.

표기법

필수 표기법 없음.

authenticate 스테레오타입

메타 클래스

ActivityPartition, Lifeline

설명

지정된 협업의 범위에서 인증을 받아야 하는 관계자를 표시하는 데 사용합니다. 이 스테레오타입은 협업의 관계자 인스턴스를 표시하는 동작 모델의 요소에 적용됩니다.

ActivityPartition(활동 다이어그램의 경우) 또는 Lifeline(상호작용
다이어그램의 경우)에 적용하는 경우, 표시된 모델 요소에 어노테이션을 지정할 수 있습니다.

특성

없음.

표기법

필수 표기법 없음.

authorize 스테레오타입

메타 클래스

ActivityNode, Message

설명

두 관계자 간의 통신에서 요청자가 요청을 수행할 권한이 있는지 확인해야 함을 표시하는 데 사용합니다. 이 스테레오타입은 호출되는 동작이 인증 확인으로 보호됨을 표시하도록 동작 모델의 메시지 및 플로우에 적용됩니다.

특성

없음.

표기법

필수 표기법 없음.

private 스테레오타입

메타 클래스

ObjectNode, Class

설명

통신에서 전송된 데이터를 개인용 데이터로 취급해야 하며 데이터가 부인방지 기능이 설정되고 보안 설정되었음을 확인하기 위해 합리적인 모든 노력을 기울이도록 표시하는 데 사용합니다.

private에 tamperproof가 내포되므로private과 tamperproof 스테레오타입을 동일한 요소에 적용하지는 마십시오. 즉, private 스테레오타입을 적용하면 데이터가 개인용 데이터이고(권한 없는 열람 방지) 부인방지 기능이 설정됨을(수정 없이 목적지에 도착하도록 보장됨) 지정합니다. 이와 반대로 tamperproof 스테레오타입을 적용하면 권한 없는 열람을 방지하지 않아도 데이터가 수정 없이 목적지에 도착하도록 지정합니다. Class(또는 파생 UML 요소)에 적용할 경우, 이 요소가 동작 모델에 표시될 때마다 서명됨을 표시합니다. 또한 이 스테레오타입을 동작 모델의 인스턴스에 적용하여 이러한 특정 경우에 요소를 달리 취급하도록 표시할 수 있습니다.

특성

없음.

표기법

필수 표기법 없음.

signed 스테레오타입

메타 클래스

ObjectNode, Class

설명

통신에서 전송된 데이터에 관계자를 식별하는 서명의 일부 개념을 포함시키도록 표시하는 데 사용합니다. 통신 관계자가 요소에 서명할 필요가 없으며 여러 개의 서명이 포함될 수 있다는 점에 유의하십시오.

프로파일은 필요한 서명을 지정하지 않으며 필요한 서명 수도 지정하지 않습니다.

Class(또는 파생 UML 요소)에 적용할 경우, 이 요소가 동작 모델에 표시될 때마다 서명됨을 표시합니다. 또한 이 스테레오타입을 동작 모델의 인스턴스에 적용하여 이러한 특정 경우에 요소를 달리 취급하도록 표시할 수 있습니다.

특성

없음.

표기법

필수 표기법 없음.

tamperproof 스테레오타입

메타 클래스

ObjectNode, Class

설명

관계자들 간에 전송된 데이터를 송신자가 보낼 때와 동일한 형식, 동일한 내용 및 의미로 수신자에게 도착하도록 보장해야 함을 표시하는 데 사용합니다.

private에 tamperproof가 내포되므로 private과 tamperproof 스테레오타입을 동일한 요소에 적용하지는 마십시오. 즉, tamperproof 스테레오타입을 적용하면 권한 없는 열람을 방지하지 않아도 데이터가 수정 없이 목적지에 도착하도록 지정합니다. 이와 반대로 private 스테레오타입을 적용하면 데이터가 개인용 데이터이고(권한 없는 열람 방지) 부인방지 기능이 설정됨을(수정 없이 목적지에 도착하도록 보장됨) 지정합니다.

특성

없음.

표기법

필수 표기법 없음.

trusted 스테레오타입

메타 클래스

ConnectableElement

설명

명시적 신뢰 영역에 참여하는 협업의 관계자 집합을 표시하는 데 사용합니다.

이 경우 ConnectableElement는 협업에서 역할을 표시하는 요소 집합이 됩니다(그림 6 참조).

특성

이름	*유형*	*설명*
영역	문자열	사용하는 지정된 모델에서 요소 집합이 됩니다.

표기법

특히 협업 다이어그램에서 영역 경계선을 그래픽으로 표시하면 유용합니다. 권한 부여 패턴의 스펙에서 그 예를 찾아볼 수 있습니다. 그림 6과 같이 신뢰 영역은 참가자 주변에 그려진 점선 경계선으로 표시됩니다.

그림 6: 신뢰 영역 경계선 표시

참조 문헌

[1] Brown, A., Johnston, S. 및 Kelly, K, Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications, Rational 소프트웨어 백서

[2] Lopes, C.V. 및 Hursch, W.L., Separation of Concerns,

Tech Report of College of Computer Science, Northeaster University, Boston, MA, Feb. 24, 1995.

[3] OMG, MDA, An Introduction, OMG

[4] OMG, UML 2.0 Superstructure Specification, OMG

[5] RosettaNet Consortium[www.rosettanet.org]

[6] Gamma, E., Helm, R., Johnson, R. 및 Vlissides, J., Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley

이 문서는 IBM developerWorks 웹 사이트에서 공개한 원본 문서의 재판본입니다.

® Copyright 2004 IBM

Corporation IBM

Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America All Rights Reserved
2005년 3월.

IBM, IBM 로고, Rational, Rational Rose, Tivoli, WebSphere 및 XDE는 미국 또는 기타 국가에서 사용되는 International Business Machines Corporation의 상표입니다.

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Microsoft 및 Visual Studio는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표 또는 등록상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

이 책에서 IBM 제품 또는 서비스를 언급했다고 해서 해당 제품 또는 서비스를 IBM이 운영되는 모든 국가에서 사용할 수 있다는 것을 의미하지는 않습니다.

IBM이 제시하는 방향 또는 의도에 관한 언급은 특별한 통지없이 변경되거나 철회될 수 있고, 단지 예상되는 목표와 계획을 제공하기 위한 것입니다. 모든 정보는 일체의 보증없이 "현상태대로" 제공됩니다.

IBM의 인터넷 홈 페이지는 ibm.com입니다.