

버전 6.0.1



기술적 제품 개요



버전 6.0.1



기술적 제품 개요

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, "주의사항"에 있는 일반 정보를 읽으십시오.

---

## 목차

제 1 장 비즈니스 통합 . . . . .	1	BPEL 프로세스 . . . . .	17
비즈니스 단위 사이의 통합. . . . .	2	상태 머신 . . . . .	18
엔터프라이즈 사이의 통합 . . . . .	3	비즈니스 규칙 . . . . .	19
WebSphere Integration Developer. . . . .	3	선택기 . . . . .	20
표준 . . . . .	5	사용자 타스크 . . . . .	21
Integration Developer 역할 . . . . .	5	인터페이스 맵 . . . . .	22
제 2 장 SCA(Service Component Architecture) . . . . .	7	중개 플로우 . . . . .	23
서비스 컴포넌트 . . . . .	8	독립형 참조 . . . . .	24
서비스 데이터 오브젝트 . . . . .	10	관련 정보 . . . . .	24
서비스 규정자 . . . . .	11	제 3 장 도구에 대한 정보 학습 . . . . .	27
모듈. . . . .	12	환영 보기 개요 . . . . .	27
가져오기 및 내보내기 . . . . .	14	환영 보기의 학습서. . . . .	28
서비스 가져오기 및 내보내기 바인딩 유형 . . . . .	14	환영 보기의 샘플 . . . . .	29
적절한 바인딩 선택. . . . .	16	Information Center . . . . .	30
서비스 구현 유형 . . . . .	16	주의사항 . . . . .	31
Java 오브젝트 . . . . .	17		

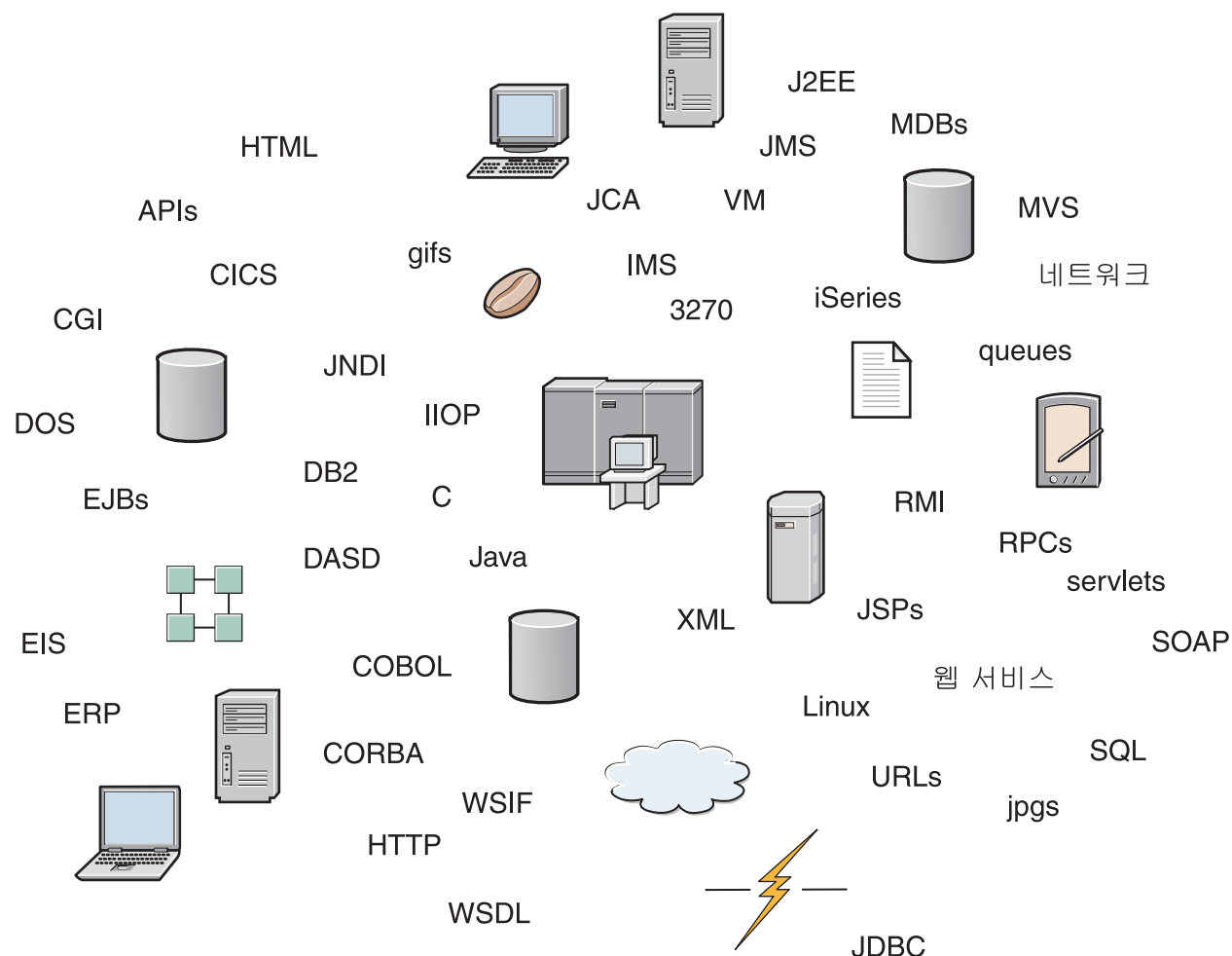


## 제 1 장 비즈니스 통합

비즈니스 통합이란 엔터프라이즈 또는 엔터프라이즈 세트 내에서 응용프로그램, 데이터 및 프로세스를 통합하는 것입니다. WebSphere Integration Developer에서 발견하게 될 task 과제 및 그 해결 방법을 점점합니다.

고객을 위해 비즈니스 응용프로그램에 대한 포털을 빌드하도록 할당되었습니다. 포털에서는 모든 비즈니스 단위 위에 전체적으로 산재해 있는 수 많은 비즈니스 응용프로그램 및 관련 데이터에 대한 액세스를 제공해야 합니다. 또한 비즈니스 파트너의 응용프로그램을 포털에 추가하도록 요청합니다. 이러한 어셈블된 응용프로그램의 시퀀스에는 몇 가지 로직이 있으므로, 통합이란 프로세스 개발을 의미하기도 합니다. 20개의 비즈니스 단위와 다양한 비즈니스 파트너가 있습니다. 사용자의 포털은 하루 24시간 내내 웹에서 사용 가능해야 합니다. 사용자를 포함하여 6명의 개발자가 있으며, 설치 및 실행을 위해 4개월의 기간이 주어졌습니다.

이러한 과제가 주어진 대부분의 사람들은 과거 수십 년 동안 쌓아온 기술을 검토하고 다음 그림과 같은 콜라주를 보게 됩니다.



어려운 일이지만 불가능한 상황도 아닙니다. 하드웨어 및 소프트웨어의 해체된 컬렉션 이외에도 사용자가 직면한 가장 어려운 문제점은 시간과 자원 제한조건입니다. 관련된 여러 엔터프라이즈에 산재해 있는 응용프로그램 및 데이터를 신속히 결합하기 위한 강력한 도구를 찾아야 합니다. 수동 코딩은 옵션이 아닙니다.

사용자만이 이러한 상황에 직면하게 될까요? 그렇지 않습니다. 널리 알려진 오래된 문제점입니다. 2001년 12월 CIO 설문조사에 따르면, 응용프로그램 통합은 꾸준히 3대 기술 우선순위 중 하나입니다. 2001년 겨울 *The Business Integrator Journal*의 최근 설문조사에 따르면, 세 명의 개발자 중 두 명은 통합 소프트웨어를 사용하여 웹 기반의 솔루션을 개발하고 있습니다. 평균적으로 이들 각각은 서로 다른 세 개 시스템을 통합하고 있습니다.

문제점을 위기 상황으로 발전시킬 수 있는 비즈니스의 두 가지 작업(엔터프라이즈 내 비즈니스 단위 간 통합 및 엔터프라이즈 간 통합)을 검토해보십시오. 그러면 WebSphere Integration Developer가 이러한 문제점을 처리하는 방법과 특히 업계 표준에 따르도록 하는 확약의 중요성을 확인할 수 있게 됩니다. 결과적으로 WebSphere Integration Developer 도구를 사용하여 이전에 제시된 문제점을 해결할 수 있는 전문가가 될 것입니다.

---

## 비즈니스 단위 사이의 통합

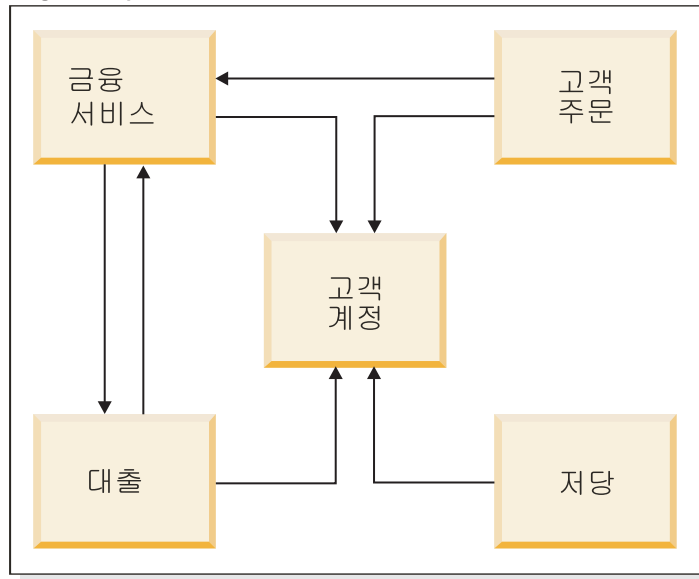
비즈니스 단위는 오늘날 협업이 자주 이루어지므로, 해당 응용프로그램 간의 밀접한 통합이 필요합니다.

기술의 발달로 비즈니스 단위를 연결시킬 수 있으며 효율성 측면에서 오버헤드를 최소화하고 결과물을 최대화할 수 있도록 보다 밀접하게 협업하는 방식으로 운영되어야 하므로 이전의 자율적인 비즈니스 단위가 통합되고 있습니다. 또한 회사의 공통적인 목적은 비즈니스 단위를 함께 이끌어 내는 것입니다. 마케팅 및 연구개발 부서 모두 이윤을 내는 제품을 생산하고자 합니다. 시장의 지식을 제품 개발 정보와 통합하면 성공적인 제품을 생산할 능력이 향상됩니다. 또한 비즈니스 단위 간의 협업으로 기업이 다른 비즈니스 컨텍스트에서 재사용을 허용하여 기존의 많은 비즈니스 응용프로그램을 강화할 수 있습니다.

비즈니스 단위 간 통합은 보안 위험이 적고 단위 간 상호작용 관리가 비교적 용이하므로 엔터프라이즈 간 통합보다 쉽습니다. 비즈니스 단위는 대개 동일한 프로토콜, 운영 체제 및 컴퓨터 언어를 사용합니다. 즉 이는 비교적 동질적인 환경입니다. 그러나 핵심은 응용프로그램을 빠르게 통합하는 올바른 도구를 확보하는 것입니다. 다음 다이어그램에서 DOM Bank에는 서로의 정보를 공유해야 하는 여러 비즈니스 장치가 있습니다. 몇 년 전, DOM Bank는 한 비즈니스 단위에서 정보 사본을 인쇄하여 다른 비즈니스 단위로 이동시켜 정보를 공유할 수 있었습니다. 현재 DOM Bank에서 경쟁자와의 경쟁 관계를 유지하려면 자체 비즈니스 단위를 통괄하는 통합된 응용프로그램을 작성해야 합니다.



## DOM Bank



## 엔터프라이즈 사이의 통합

파트너십 또는 인수 과정에서 공유 데이터 및 프로세스가 필요한 경우 비즈니스 단위 사이의 응용프로그램 통합은 엔터프라이즈 사이에서도 적용됩니다.

기술을 통해 기업들을 상호 유익한 영역으로 연결할 수 있습니다. 예를 들어, 자동차 제조업체에서 타이어 재고가 적은 경우, 이를 공급자에게 자동으로 알릴 수 있도록 타이어 공급자와 통합된 프로세스를 설정할 수 있습니다. 기업 간의 통합은 경제적 필요성에 의해 대두되고 있습니다. 기업 간에 긴밀한 연관을 맺으면 작업 완료에 따른 지연 시간과 오버헤드가 줄어듭니다. 이러한 자동화된 프로세스는 기업 간 트랜잭션 처리 시간을 줄이며 여행 비용 및 직접적인 회의 시간을 상당히 줄일 수 있음을 의미합니다. 마찬가지로 관리 비용이 감소하고 알림, 배달 및 송장 단계 사이의 진행 시간도 개선됩니다.

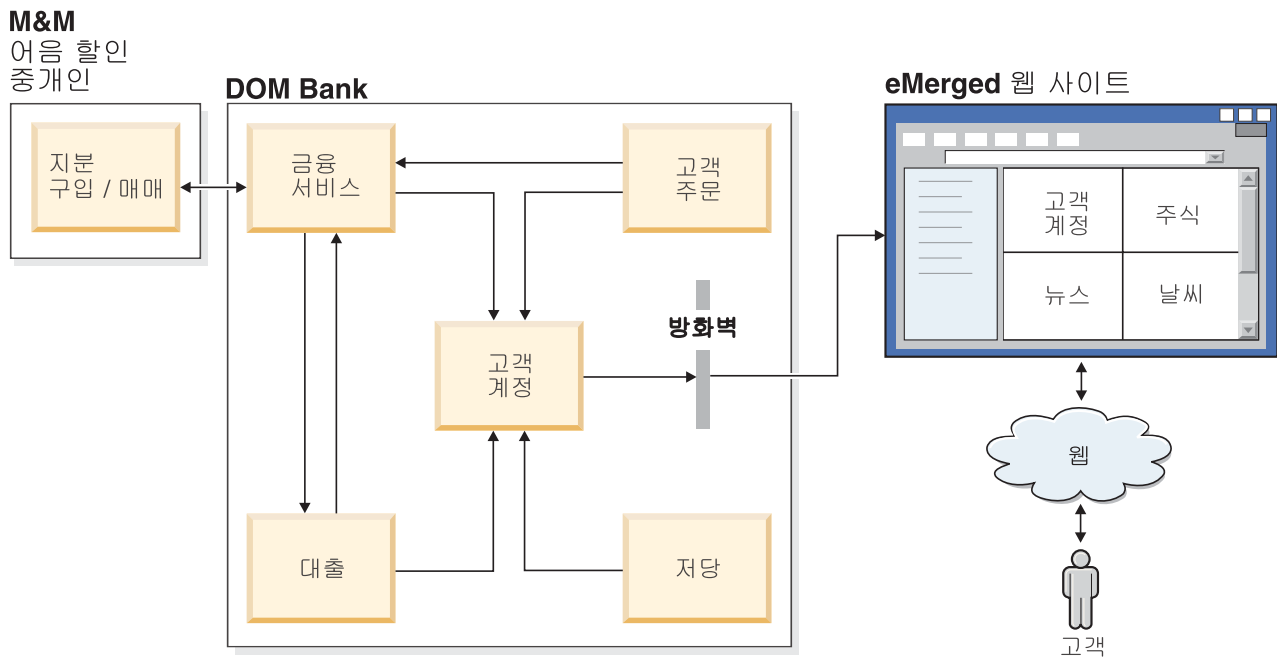
그러나 각 엔터프라이즈는 서로 다른 히스토리를 갖고 있습니다. 해당 응용프로그램은 다른 통신 프로토콜을 사용하여 각기 다른 플랫폼에서 다른 언어로 코드화됩니다. 또한 다른 조직과 작업할 때 보안의 위험은 더 커집니다. 엔터프라이즈 간의 통합에 대한 장점이나 필요성에 관계없이 적절한 도구가 없는 경우, 이를 개발하는데 필요한 비용이 매우 중요할 수 있습니다.

## WebSphere Integration Developer

WebSphere Integration Developer는 조직에서 매일 당면하고 있는 통합 과제에 대한 응답입니다. 이 제품은 빌드된 통합 응용프로그램에 적합한 완전한 통합 개발 환경으로 디자인되었습니다. 통합 응용프로그램의 개발을 단순화하고 가속화하기 위해, 이 환경에서는 추상 계층을 제공하여 작업할 시각적 컴포넌트를 기본 구현에서 분리합니다.

통합 응용프로그램은 단순하지 않습니다. 통합 응용프로그램은 EIS(Enterprise Information System)에서 응용 프로그램을 호출하고 부서 또는 엔터프라이즈에서 비즈니스 프로세스를 처리하며 다양한 언어로 작성되어 다

양한 운영 체제에서 실행되는 응용프로그램을 로컬 또는 원격으로 호출할 수 있습니다. 예를 들어, eMerged Corporation은 DOM bank와 M&M Discount broker가 병합되어 만들어졌습니다. 병합 대상은 EIS 시스템의 응용프로그램, 비즈니스 프로세스와 같은 위의 모든 항목을 의미합니다. 각각의 이전 회사 내부의 응용프로그램은 기업 사이에서 공유하여 새로운 고객들에게 계속적으로 이어질 수 있도록 해야 합니다. 그러나 다음 다이어그램에 표시된 대로 eMerged에서 타스크를 달성하고 이 두 비즈니스의 각 고객들은 모든 금융 정보에 온라인으로 액세스할 수 있습니다.



eMerged는 WebSphere Integration Developer 도구를 사용하여 기업 및 고객에 적합한 통합 응용프로그램을 빌드합니다. 이 도구에서는 EIS 시스템에서 원격으로 존재하는 응용프로그램을 포함하는 응용프로그램 및 비즈니스 프로세스를 컴포넌트로 표시합니다. Visual Editor를 통해 컴포넌트가 작성되어 기타 통합 응용프로그램(즉, 컴포넌트 세트에서 작성된 응용프로그램)으로 어셈블됩니다. Visual Editor는 컴포넌트 및 해당 구현 사이의 추상 계층을 표시합니다. 이 도구를 사용하는 개발자는 각 컴포넌트의 기본 구현에 대한 자세한 지식 없이도 통합 응용프로그램을 작성할 수 있습니다.

도구에서는 하향식 또는 상향식 접근 방식 모두를 허용합니다. 하나 이상의 컴포넌트에 대한 구현이 없으며 이 후에 추가되는 통합 응용프로그램을 빌드하는 경우가 하향식 접근 방식이며 컴포넌트가 이미 구현되고 개발자가 컴포넌트를 Visual Editor에 끌어 놓아 어셈블한 후에 회선으로 결합하여 이 컴포넌트 사이에서 논리 플로우를 작성하는 경우가 상향식 접근 방식입니다. 디버깅 및 테스트 환경은 프로덕션 서버에 응용프로그램을 배치하기 전의 전체 테스트를 의미합니다. 모니터링 지점을 설정하면 성능을 최적화하기 위해 실시간으로 응용프로그램을 사용하여 미세 조정하는 과정을 볼 수 있습니다.

WebSphere Integration Developer 도구는 서비스 지향 아키텍처를 기반으로 합니다. 컴포넌트는 서비스이며 여러 컴포넌트와 관련된 통합 응용프로그램도 서비스입니다. 작성된 서비스는 주도적인 업계 표준을 준수합니다. 또한, 컴포넌트가 되는 비즈니스 프로세스는 업계 표준 BPEL(Business Process Execution Language)을

준수하는 사용하기 쉬운 비주얼 도구로 이와 유사하게 작성됩니다. WebSphere Integration Developer는 Windows 및 Linux 플랫폼 모두에서 사용할 수 있습니다.

다음은 WebSphere Integration Developer 도구의 이점입니다.

- 학습하기 쉽습니다.
- 복잡한 통합 상황에 적용할 수 있습니다.
- 업계 전반의 표준을 준수하는 응용프로그램을 신속하게 생성할 수 있습니다.

## 표준

WebSphere Integration Developer에서 작성한 응용프로그램은 서비스 지향 아키텍처와 연관된 업계 전반의 표준을 준수합니다.

몇 년 간 지원되지 않거나 높은 라이선스 요금이 부과될 수 있는, 독점 코드와 연결된 응용프로그램을 작성하려는 사람은 없습니다. 그러므로 표준 기반의 통합은 WebSphere Integration Developer의 기본 측면입니다. 연결을 위해 J2EE Connector Architecture 표준이 사용됩니다. 데이터 전달을 보장해야 하는 대형 응용프로그램에서 종종 사용되는 비동기 메시징에서는 JMS(Java Message Service) 표준이 사용됩니다. WebSphere Integration Developer는 SOAP(Simple Object Access Protocol)을 기반으로 웹 서비스를 쉽게 통합할 수 있습니다. 서비스를 설명하기 위해 잘 설정된 WSDL(Web Services Description Language) 표준이 사용됩니다. 비즈니스 프로세스를 정의하기 위해서는, BPEL(Business Process Execution Language) 표준이 사용됩니다.

이 표준 기반 인터페이스 및 컴포넌트는 개방형, 연결 가능 아키텍처로 구성됩니다. 그러나 독점 요소는 제외되지 않습니다. 독점 요소는 표준화된 인터페이스를 사용하여 액세스합니다. 이는 WebSphere Integration Developer에서 작성된 응용프로그램이 예를 들어, .NET 응용프로그램과 상호작용할 수 있음을 의미합니다. 아키텍처 섹션에서 전체 SCA에 대한 링크가 제공되며, 이는 지원되는 많은 표준에 대한 확장 목록을 제공합니다.

## Integration Developer 역할

통합 개발자는 WebSphere Integration Developer의 주 사용자입니다. 이 사용자는 비주얼 도구를 사용하여 기본적인 구현에 대한 해박한 지식 없이도 복잡한 통합 응용프로그램을 빌드할 수 있습니다.

WebSphere Integration Developer는 응용프로그램과 비즈니스 프로세스를 컴포넌트로 표시합니다. 컴포넌트 구현은 숨겨진 상태이며 컴포넌트는 인터페이스를 통해 상호 운영됩니다. 따라서 통합 개발자는 컴포넌트를 사용하는 통합 응용프로그램을 작성하기 위해 컴포넌트의 기초적인 구현에 대한 상세한 지식이 필요하지 않습니다. 그러나 통합 개발자는 EIS 시스템, 비즈니스 프로세스 및 Java 또는 기타 언어로 코딩된 응용프로그램에 대한 이해가 필요하므로 통합 필드에 대한 폭넓은 기술 지식을 가지고 있을 수 있습니다. 예를 들어, 설계자는 각 컴포넌트가 수행하는 작업을 자세히 알지 않고도 시스템의 작동 방식을 폭넓게 이해하고 있습니다. 설계자와 마찬가지로, 통합 개발자는 조직에서 전반적인 응용프로그램을 디자인하는 사람이고 나머지 다른 사람들은 특정 컴포넌트의 구현을 코드화합니다.

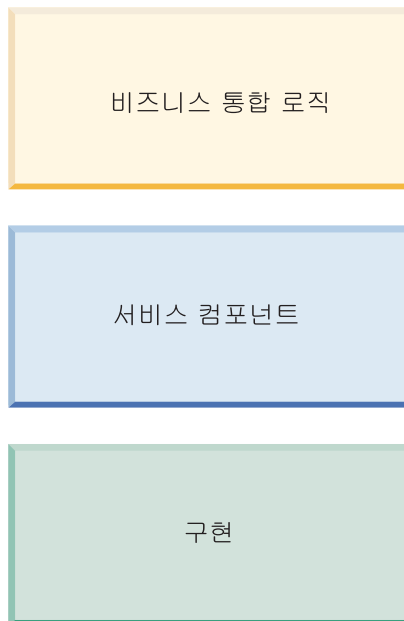


---

## 제 2 장 SCA(Service Component Architecture)

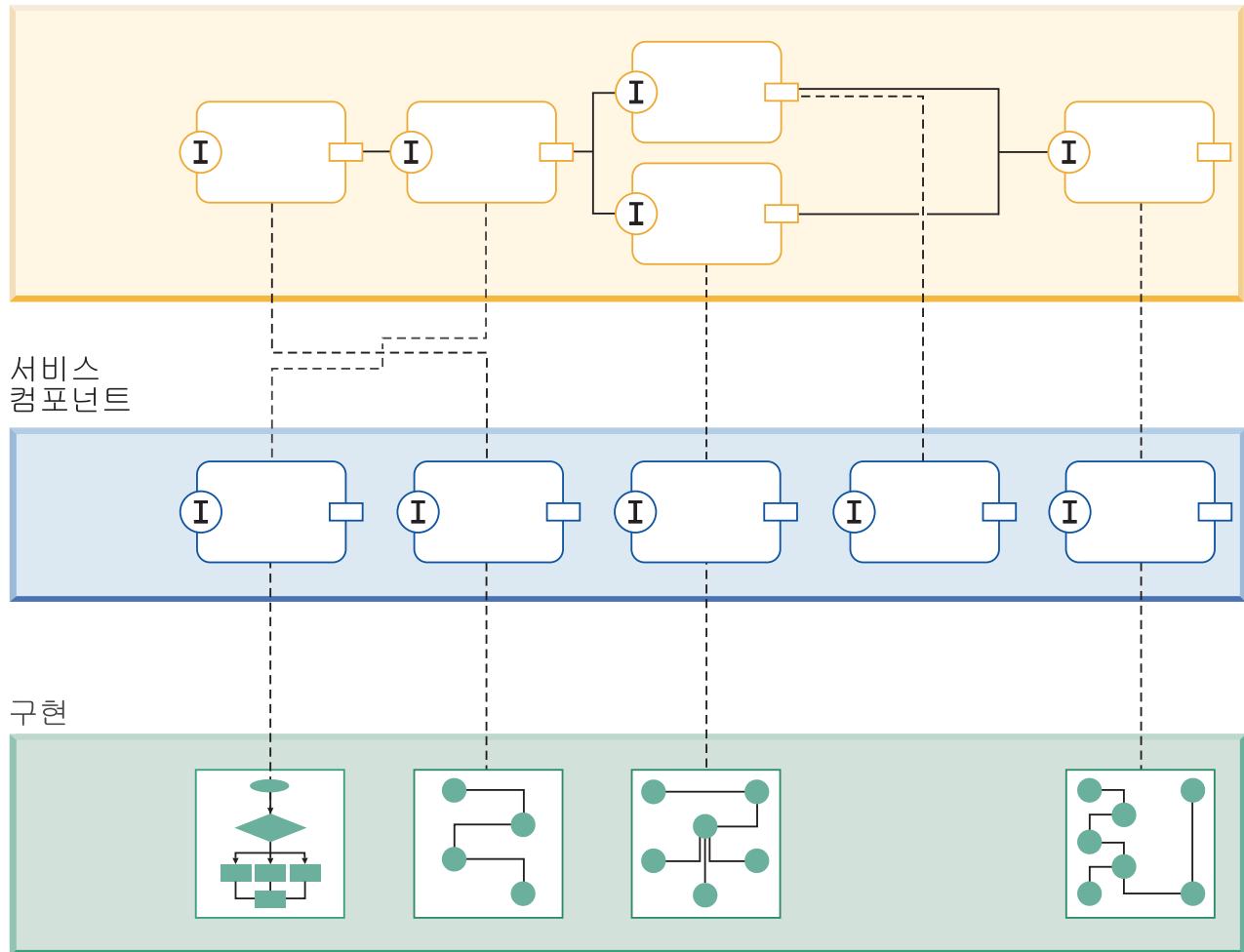
업계 표준 서비스 지향 아키텍처를 기반으로 하는 SCA는 모든 비즈니스 프로세스(데이터베이스, EIS(Enterprise Information System) 서비스 자산, 작업 흐름, 데이터베이스 등)를 서비스 지향 방식으로 표시합니다. 이 섹션에서는 서비스 및 비즈니스 로직을 함께 표현하고 비즈니스 데이터를 참조하는 이 아키텍처가 작성한 서비스 데이터 오브젝트를 상위 레벨에서 점검합니다.

SCA의 목적은 통합 개발자가 구현 세부사항보다 통합 응용프로그램의 어셈블링에 초점을 맞출 수 있도록 비즈니스 로직과 구현을 분리하는 것입니다. 이 목표를 달성하기 위해, 비즈니스 프로세스에서 요구하는 개별 서비스 구현을 포함하는 서비스 컴포넌트를 작성합니다. 결과는 다음 다이어그램에 표시된 바와 같은 3개의 계층 구조로 되어 있습니다: 비즈니스 통합 로직, 서비스 컴포넌트 및 구현



서비스 컴포넌트에는 구현이 포함되므로, 하위 레벨 구현 세부사항에 대한 지식이 없는 통합 개발자도 그래픽으로 서비스 컴포넌트를 어셈블할 수 있습니다. 서비스 컴포넌트에서는 통합 개발자를 지원하는 또는 통합 개발자가 이후에 구현을 추가할 수 있는 옵션도 제공합니다. 제품에 표시되는 바와 같이, 컴포넌트는 시각적으로 함께 어셈블됩니다. 즉, 컴포넌트 내부의 코드는 사용할 수 없습니다. 아래 다이어그램에 표시된 비즈니스 로직 레벨에서, 컴포넌트는 해당 구현과 별도로 어셈블됩니다. 그런 다음 서비스 지향 아키텍처를 통해 사용 중인 서비스 구현 기술로 관심을 분산시키는 대신 컴포넌트를 사용 또는 재사용하여 비즈니스 문제점을 해결하는 데 초점을 맞출 수 있습니다.

## 비즈니스 통합 로직

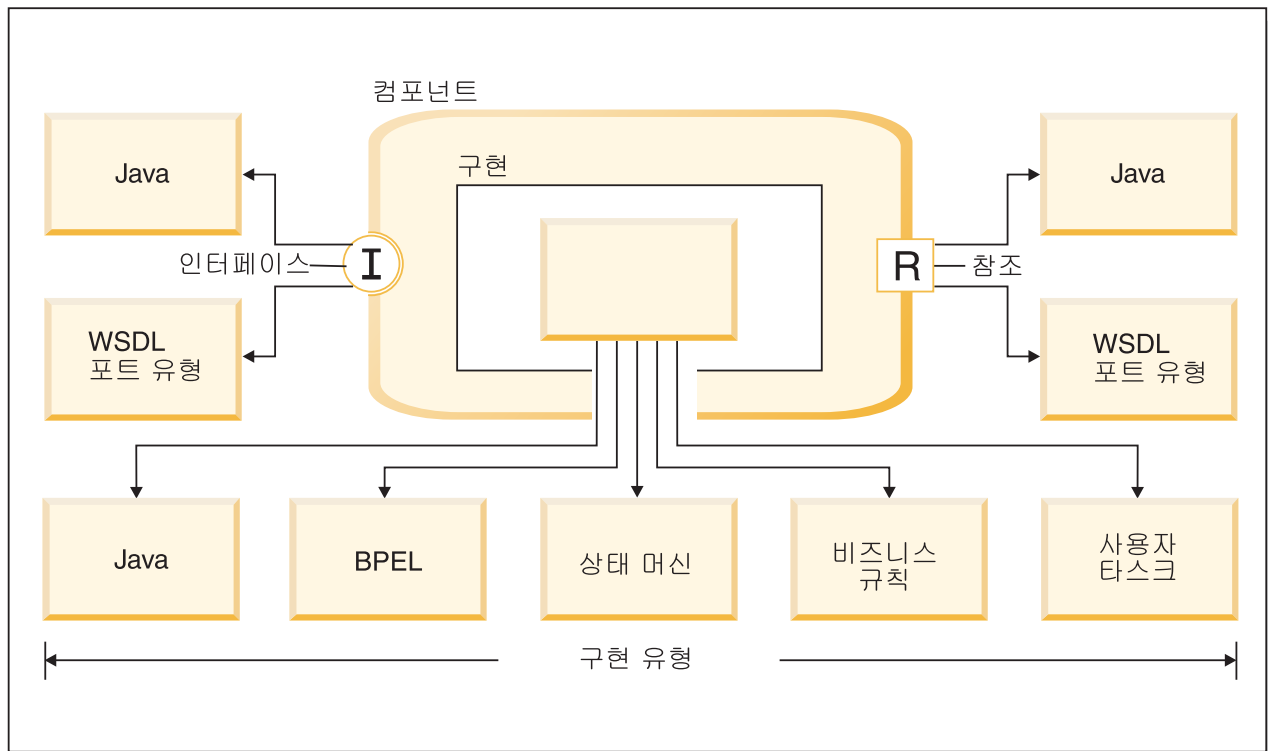


## 서비스 컴포넌트

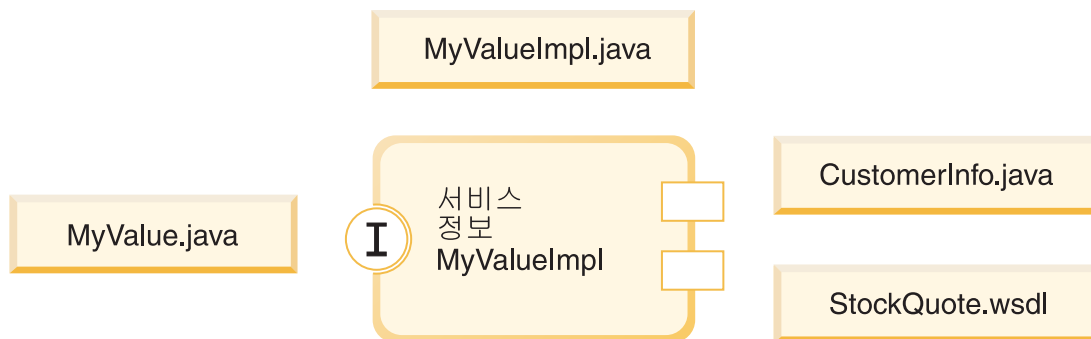
서비스 컴포넌트는 서비스 구현을 구성합니다. 서비스 컴포넌트는 표준 블록 다이어그램에서 표시됩니다.

컴포넌트는 구현(WebSphere® Integration Developer 도구를 사용할 경우 숨겨짐), 해당 입력, 출력 및 결합을 정의하는 하나 이상의 인터페이스 및 0개 이상의 참조로 구성됩니다. 참조는 해당 컴포넌트에서 요구하거나 또는 이용하는 다른 서비스 또는 컴포넌트의 인터페이스를 식별합니다. 인터페이스는 WSDL 포트 유형 또는 Java™의 두 가지 언어 중 하나로 정의될 수 있습니다. 인터페이스는 동기 및 비동기 상호작용 스타일을 지원합니다. 컴포넌트 구현은 다양한 언어로 표시될 수 있습니다.

인터페이스 유형은 WSDL 또는 Java일 수 있습니다. 그러나 인터페이스가 여러 개인 경우 Java와 WSDL을 혼합할 수는 없습니다.

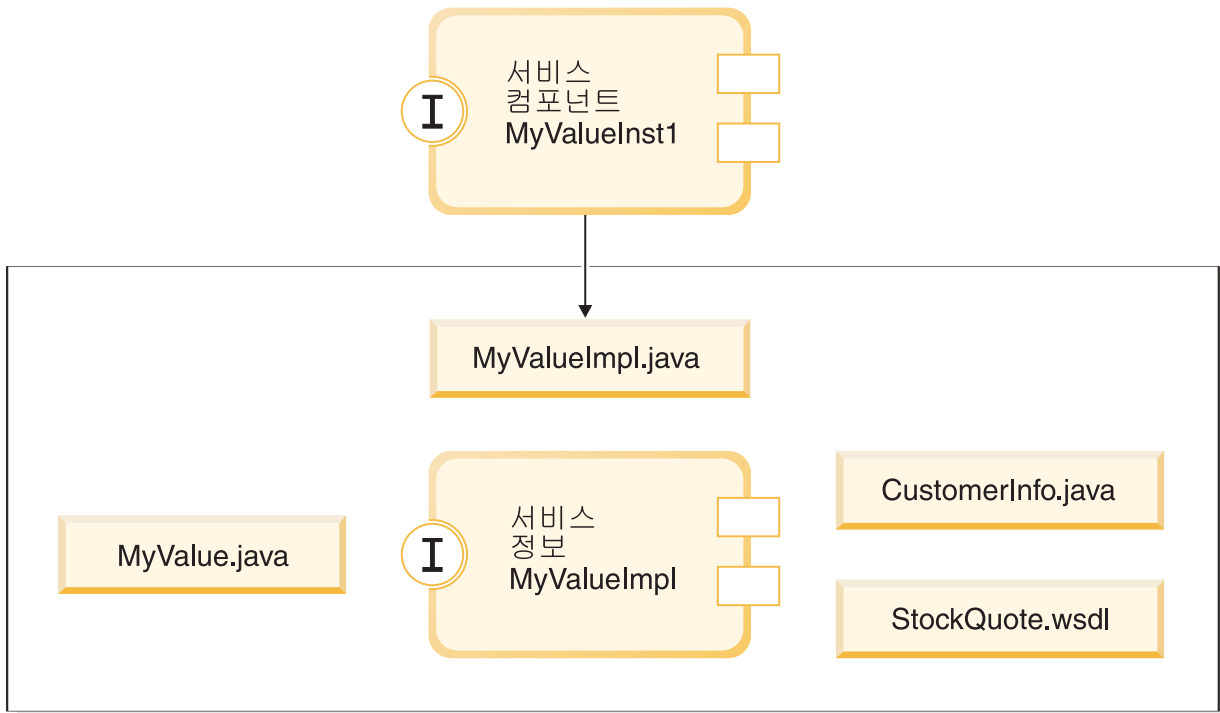


아래 그림에서는 중앙에 컴포넌트가 있습니다. 해당 구현, MyValueImpl은 인터페이스이며 Java로 되어 있습니다. 다른 Java 인터페이스 및 WSDL 인터페이스와 같은 두 가지 참조가 있습니다.



이 컴포넌트를 사용하는 경우 아래 표시된 대로 실제 컴포넌트만을 보게 됩니다. 이 컴포넌트에서의 다른 컴포넌트에 대한 참조는 시각적으로 해당 인터페이스에 한 행으로 표시됩니다. 이 컴포넌트에서의 참조는 해당 참조 지점에서 다른 컴포넌트의 인터페이스까지 한 행으로 표시됩니다. 참조란 해당 컴포넌트가 이용하는 서비스를 나타냅니다. 참조의 이름을 지정하고 해당 인터페이스를 지정하여 컴포넌트 구현 작성자는 실제 서비스를 참조하는 바인딩을 나중에 연기할 수 있습니다. 해당 시점이 되면, 통합 전문가가 참조로부터 다른 컴포넌트 또는 가져오기 인터페이스로 연결하여 바인딩을 수행합니다. 이러한 느슨한 결합은 지연된 바인딩 및 구현 재사용을 허용하며, WebSphere Integration Developer의 서비스 컴포넌트 아키텍처를 사용하는 중요한 이유 중 하나입니다.

또한 컴포넌트에는 특성 및 규정자가 있을 수 있습니다. 규정자는 인터페이스의 QoS(quality of service) 지시 문이며 런타임에 대한 참조입니다.



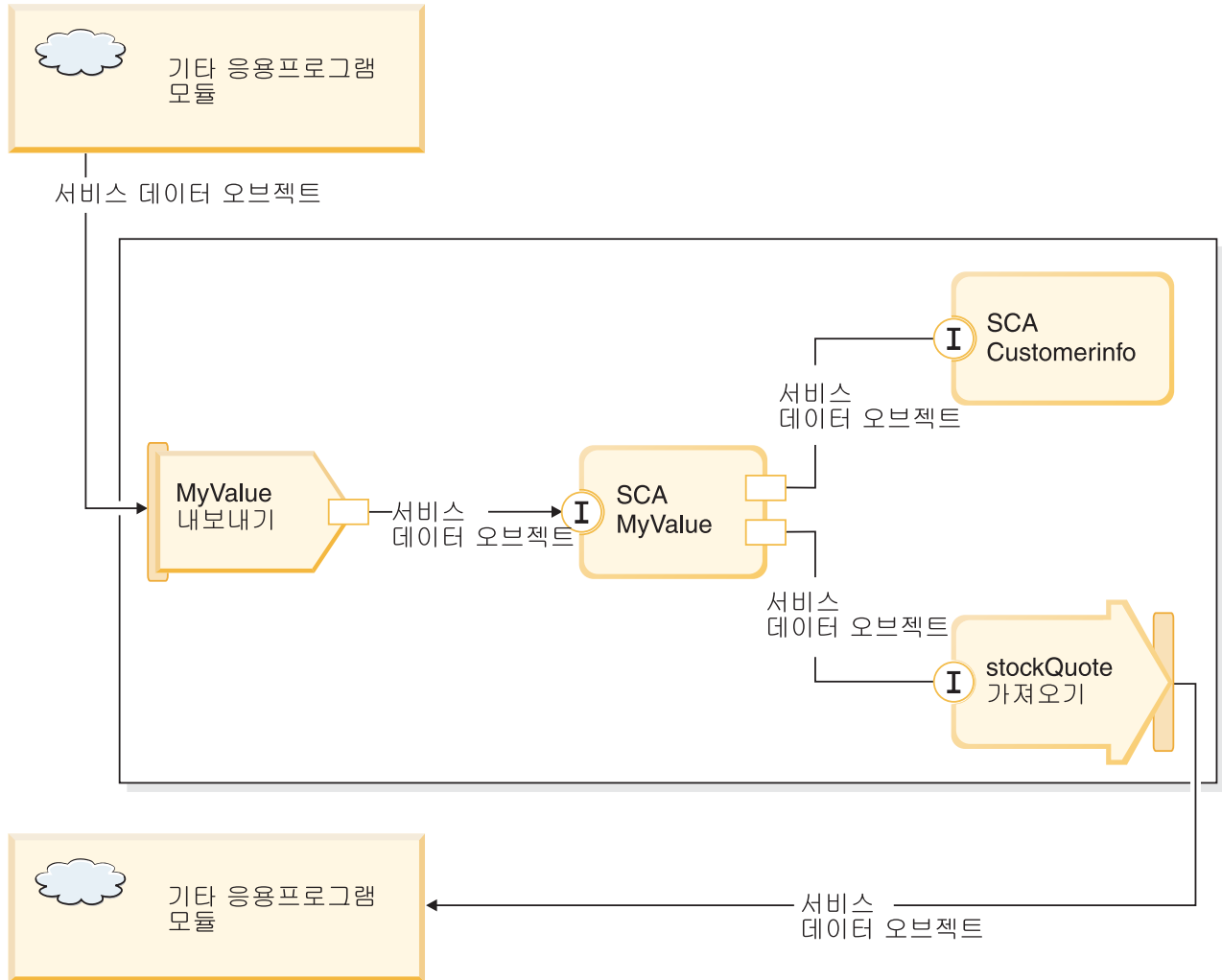
## 서비스 데이터 오브젝트

서비스 데이터 오브젝트는 SCA(Service Component Architecture)를 보완합니다. SCA(Service Component Architecture)는 컴포넌트로 서비스 및 서비스 사이의 연결성을 정의합니다. 서비스 데이터 오브젝트는 컴포넌트 사이의 데이터 플로우를 정의합니다.

각 컴포넌트는 입력 및 출력으로 정보를 전달합니다. 서비스를 호출하면, WSDL 포트 유형을 사용할 경우 문서 리터럴 인코딩을 포함하는 XML 문서로 또는 Java 인터페이스를 사용할 경우 Java 오브젝트로 데이터 오브젝트가 전달됩니다. 데이터 오브젝트는 SCA(Service Component Architecture) 서비스의 데이터 및 메타데이터에서 선호되는 양식입니다. 컴포넌트와 마찬가지로 서비스 데이터 오브젝트는 해당 구현에서 데이터 오브젝트를 분리합니다. 예를 들어, 컴포넌트는 구매 주문과 상호작용하는 반면 구매 주문 자체는 JDBC, EJB 등을 사용하여 데이터를 갱신할 수 있습니다. 서비스 데이터 오브젝트를 통해 통합 개발자는 비즈니스 아티팩트 사용에 초점을 맞출 수 있습니다. 실제로, 서비스 데이터 오브젝트는 통합 개발자에게 투명합니다. 서비스 데이터 오브젝트 JSR(Java Specification Request)에 의해 정의됩니다.

다음 다이어그램에서 서비스 데이터 오브젝트는 외부 서비스에서 내보내기로, 내보내기에서 컴포넌트로, 컴포넌트에서 컴포넌트로, 컴포넌트에서 가져오기로 그리고 가져오기에서 서비스로 전달됩니다. 다음에 나오는 가져오기 및 내보내기 섹션에서 가져오기 및 내보내기를 설명합니다.





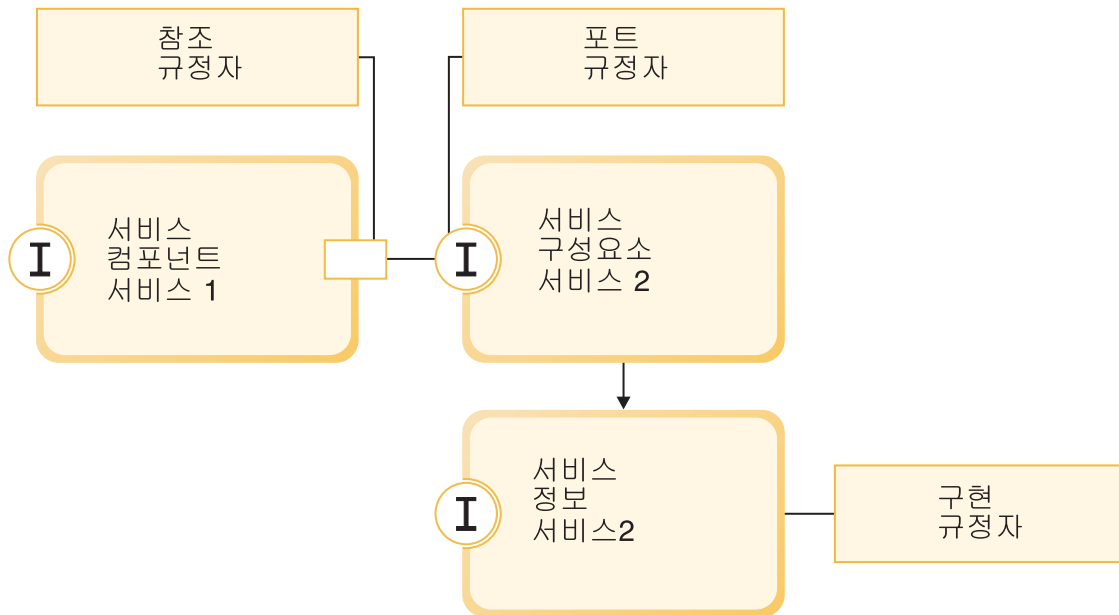
## 서비스 규정자

응용프로그램은 서비스 규정자를 지정하여 런타임 환경에 대한 QoS(Quality of Service) 요구와 통신합니다. 서비스 규정자는 서비스 클라이언트와 대상 서비스 사이의 상호작용을 제어합니다.

규정자는 서비스 컴포넌트 참조, 인터페이스 및 구현에서 지정할 수 있습니다. QoS 값 선언은 구현의 외부에서 수행하므로, 구현을 변경하지 않고도 이들 값을 변경하거나 동일한 구현의 여러 인스턴스가 다른 컨텍스트에서 사용될 때 이 값을 다르게 설정할 수 있습니다.

다음은 규정자 카테고리입니다.

- 트랜잭션 - 트랜잭션 유형의 규칙
- 활동 세션 - 활동화된 세션을 결합하는 규칙
- 보안 - 권한 규칙
- 비동기 신뢰성 - 비동기 메시지 전달 규칙



## 모듈

모듈은 EAR(Enterprise Archive) 파일에 패키징되는 아티팩트를 관별하는 배치 단위입니다. 모듈 내의 컴포넌트는 성능을 위해 나란히 위치할 수 있으며, 참조에 의해 해당 데이터를 전달할 수 있습니다. 모듈은 범위 지정 메커니즘으로 간주할 수 있습니다. 즉, 아티팩트의 구조적 경계를 설정합니다.

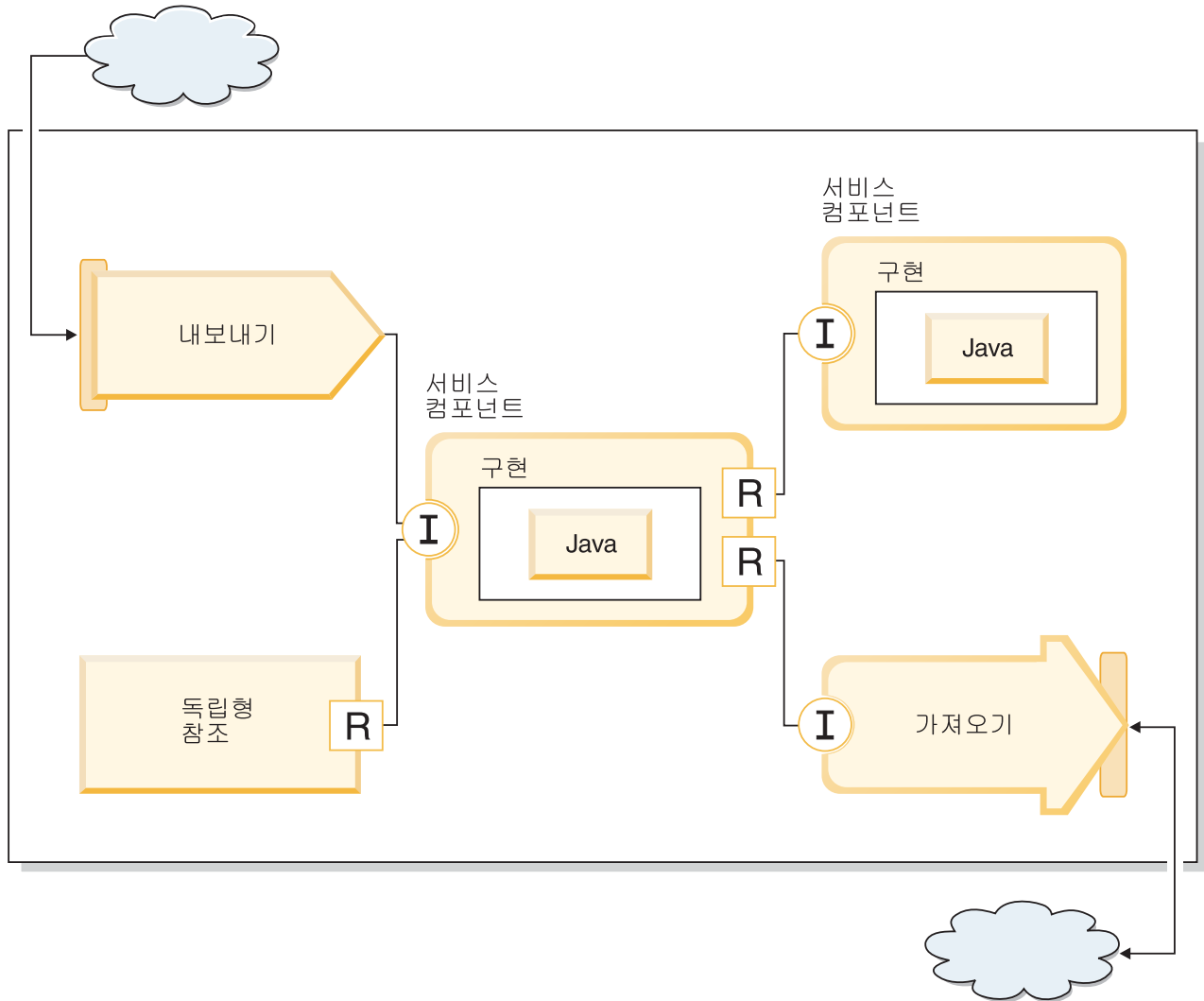
모듈은 서비스 컴포넌트, 가져오기 및 내보내기로 구성되어 있습니다. 가져오기 및 내보내기 서비스 컴포넌트는 동일한 프로젝트 및 루트 폴더에 상주하며, 가져오기 및 내보내기에 필요한 컴포넌트 및 바인딩을 링크하는 연결도 포함합니다. 모듈은 또한 가져오기 및 내보내기 컴포넌트에서 참조하는 구현 및 인터페이스를 포함하거나 또는 라이브러리 프로젝트와 같은 다른 프로젝트에 위치할 수도 있습니다.

두 가지 유형의 모듈이 있습니다. 첫 번째, 비즈니스 프로세스를 지원하기 위해 종종 사용되는 여러 가지 컴포넌트 유형 선택사항을 포함하는 모듈(때때로 비즈니스 통합 모듈이라고도 함). 두 번째, 중개 플로우 컴포넌트라는 하나의 컴포넌트와 중개 플로우 컴포넌트의 기능을 증대시키는 0개 이상의 Java 컴포넌트를 포함하는 중개 모듈.

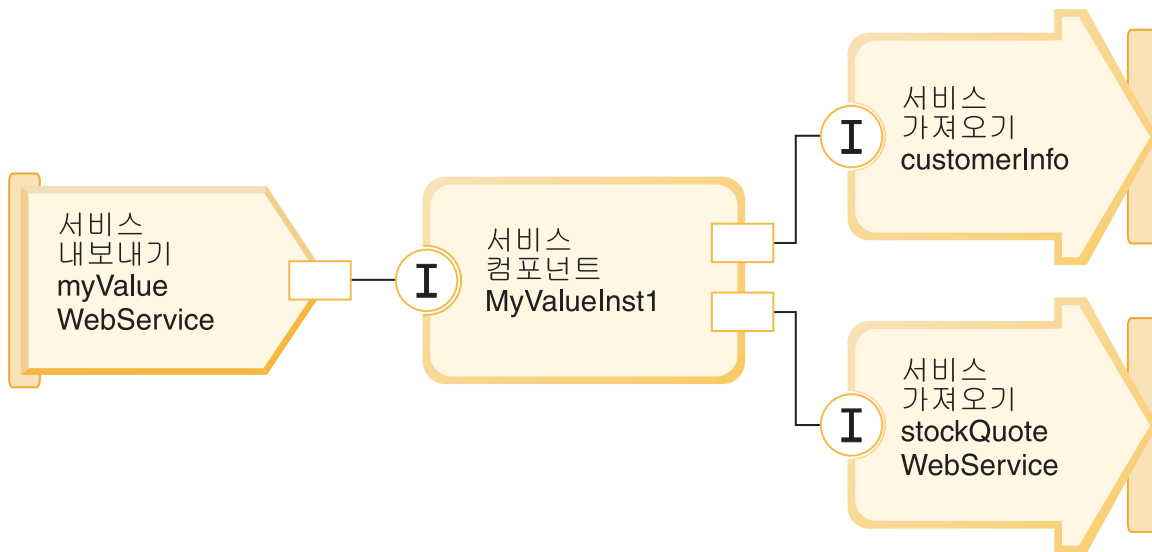
두 가지 모듈 유형이 존재하는 이유는 무엇일까요? 첫 번째 모듈 유형은 주로 비즈니스 프로세스용으로 설계되었습니다. 중개 모듈은 인터페이스 서비스 버스 아키텍처에서 일반적인, 기존 외부 서비스에 대한 게이트웨이와 유사합니다. 이들 외부 서비스 또는 내보내기는 내보내기 또는 서비스 제공자에 의해 중개 모듈에서 액세스됩니다. 중개 플로우에 의해 서비스 제공자로부터 클라이언트 서비스 요청자를 디커플링하여, 응용프로그램은 서비스 지향 아키텍처의 목표인 유동성 및 복원력을 확보합니다. 예를 들어, 중개 플로우는 수신 메시지를 로그하고, 메시지를 런타임 시 판별된 특정 서비스에 라우트하거나 또는 다른 서비스에 전달하는 데 적합하도록 데이터를 변형할 수 있습니다. 이러한 기능은 요청자 또는 제공자 서비스를 수정하지 않고 여러 번 추가 및 변경할 수 있습니다.

모듈은 WebSphere Process Server에 대해 테스트하고 배치한 서비스 응용프로그램이 됩니다. 중개 모듈은 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 서버에 대해 테스트하고 배치한 서비스 응용프로그램이 됩니다. 두 가지 모듈 유형 모두 가져오기 및 내보내기를 지원합니다.

구현, 인터페이스, 비즈니스 오브젝트, 비즈니스 오브젝트 맵, 역할, 관계 및 기타 아티팩트를 모듈 간에 자주 공유해야 합니다. 라이브러리는 이러한 공유 자원을 저장하는 데 사용되는 프로젝트입니다.



다음 다이어그램에서 모듈은 하나의 내보내기, 두 개의 가져오기 및 이들을 사용하는 서비스 컴포넌트를 포함합니다. 인터페이스 및 참조를 링크하는 연결이 표시됩니다.



모듈 및 중개 모듈 아티팩트는 다음을 포함합니다.

- 모듈 정의 - 모듈을 정의합니다.
- 서비스 컴포넌트 - 모듈의 서비스 정의. 모듈 안에서 서비스 컴포넌트의 이름은 고유합니다. 그러나 서비스 컴포넌트에는 임의의 표시 이름이 있을 수 있으며, 일반적으로 사용자에게 보다 유용합니다.
- 가져오기 - 해당 모듈에 외부적인 서비스를 호출하는 가져오기 정의. 가져오기에는 바인딩이 있으며, 가져오기 및 내보내기 섹션에서 설명합니다.
- 내보내기 - 해당 모듈에 외부적인 호출자에게 컴포넌트를 표시하는 데 사용되는 내보내기 정의. 내보내기에 는 바인딩이 있으며, 가져오기 및 내보내기 섹션에서 설명합니다.
- 참조 - 모듈의 한 컴포넌트에서 다른 컴포넌트에 대해 참조합니다.
- 독립형 참조 - 서비스 컴포넌트 아키텍처 컴포넌트로 정의되지 않은 응용프로그램을 참조합니다(예를 들어, JavaServer Pages). 이를 사용하여 해당 응용프로그램은 서비스 컴포넌트 아키텍처 컴포넌트와 상호작용할 수 있습니다. 모듈 당 하나의 독립형 참조 아티팩트만이 있을 수 있습니다.
- 기타 아티팩트 - 이 아티팩트는 WSDL 파일, Java 클래스, XSD 파일, BPEL 프로세스 등을 포함합니다.

## 가져오기 및 내보내기

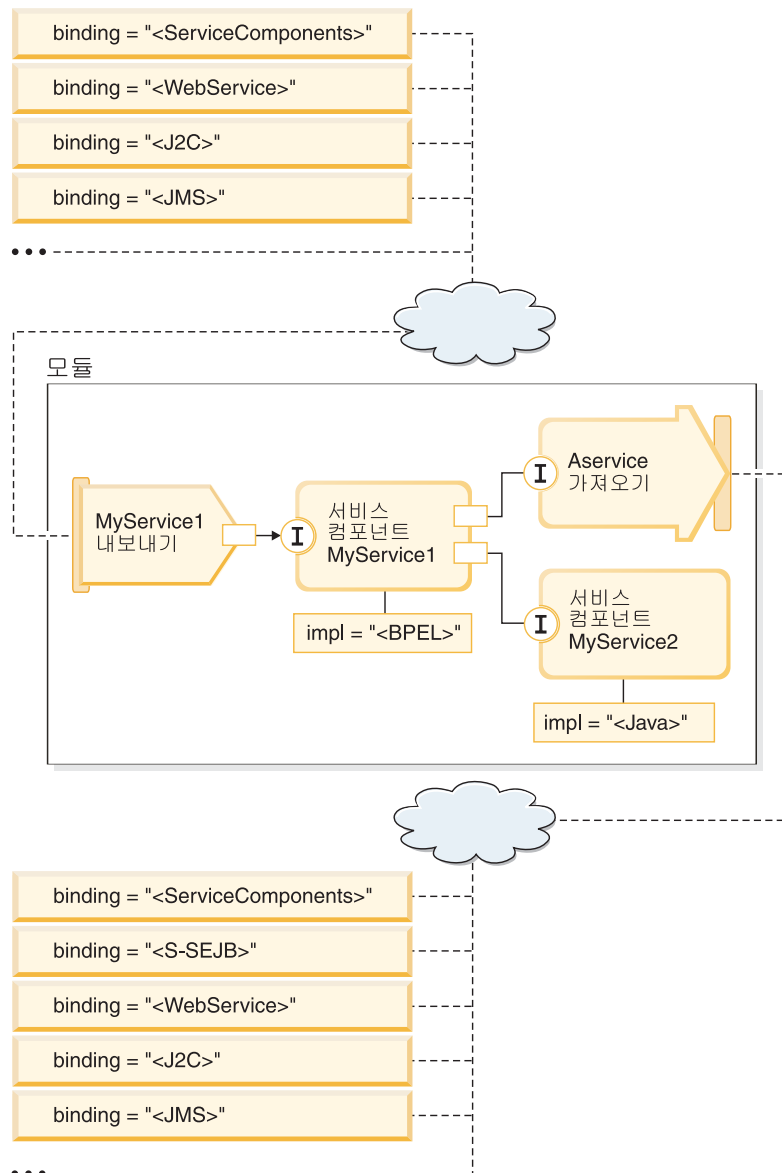
가져오기 및 내보내기에서는 모듈의 외부 인터페이스 또는 액세스 지점을 정의합니다. 가져오기는 모듈 외부의 서비스를 식별하므로 모듈 내에서 호출할 수 있습니다. 내보내기에서는 컴포넌트가 해당 서비스를 외부 클라이언트에 제공하게 합니다. 가져오기 또는 내보내기에 대해 바인딩 정보는 필수입니다. 몇 가지 바인딩이 사용 가능하며 사용자 응용프로그램에 적합한 유형에 대한 정보가 제공됩니다.

## 서비스 가져오기 및 내보내기 바인딩 유형

가져오기 및 내보내기에는 모듈에서 데이터를 전송하는 수단을 지정하는 바인딩 정보가 필요합니다. 가져오기 바인딩은 외부 서비스를 가져오기 컴포넌트에 바인드하는 특정 방법에 대해 설명합니다. 내보내기 바인딩에서는 모듈 서비스를 클라이언트에서 사용 가능하게 하는 특정 방법을 설명합니다.

SCA 또는 기본 바인딩을 통해 사용자 서비스에서 기타 모듈의 기타 서비스와 통신할 수 있습니다. SCA 바인딩이 있는 가져오기를 통해 다른 모듈의 서비스에 액세스할 수 있습니다. SCA 바인딩이 있는 내보내기를 통해 다른 모듈에 서비스를 제공할 수도 있습니다. 웹 서비스 가져오기 바인딩을 통해 외부 웹 서비스를 가져오기로 바인딩할 수 있습니다. 웹 서비스 내보내기 바인딩을 통해 외부 클라이언트에 웹 서비스로 서비스를 제공할 수 있습니다. 엔터프라이즈 서비스 발견 마법사에서 EIS 시스템의 서비스를 나타내는 가져오기 및 내보내기를 작성합니다. 작성된 바인딩은 EIS 또는 JMS(Java Message Service) 유형입니다. EIS 바인딩은 EIS 시스템의 서비스와 동기 통신할 수 있습니다. JMS 바인딩은 신뢰성 측면에서 메시지 대기열을 통해 비동기 통신이 필요한 대형 EIS 시스템과 상호작용할 때에 일반적으로 사용됩니다. 가져오기(내보내기를 통해서가 아님)에도 Stateless 세션 EJB 바인딩이 있을 수 있습니다.

어셈블리 편집기는 지원하는 바인딩을 나열하고 가져오기 또는 내보내기를 작성할 때 해당 작성을 단순화합니다. 어셈블리 편집기의 특성 보기에서는 가져오기 또는 내보내기의 바인딩 정보를 표시합니다.



## 적절한 바인딩 선택

이 섹션에서는 특정 바인딩이 응용프로그램의 요구사항에 보다 적절한 경우에 대해 설명합니다.

WebSphere Integration Developer에서 사용 가능한 바인딩에서는 선택 범위를 제공합니다. 이 목록을 사용하여 한 유형의 바인딩이 다른 유형의 바인딩보다 응용프로그램의 요구사항에 더 적합한지 식별할 수 있습니다.

다음 요소가 적용되는 경우 *SCA* 바인딩을 고려하십시오.

- WebSphere Integration Developer 모듈에 모든 서비스가 포함되어 있으며 외부 서비스가 없는 경우
- 성능이 중요한 경우
- 모듈이 단단히 결합된 경우

다음 요소가 적용되는 경우 웹 서비스 바인딩을 고려하십시오.

- 인터넷을 통해 외부 서비스에 액세스하거나 인터넷을 통해 서비스를 제공할 수 있어야 하는 경우
- 서비스가 느슨하게 결합된 경우
- 액세스하고 있는 외부 서비스의 프로토콜이나 제공하고자 하는 서비스가 SOAP/HTTP 또는 JMS/HTTP인 경우

다음 요소가 적용되는 경우 *EIS* 바인딩을 고려하십시오.

- 자원 어댑터를 사용하여 EIS 시스템의 서비스에 액세스해야 하는 경우
- 성능이 신뢰성보다 더 중요한 경우. 즉, 동기 데이터 전송이 비동기보다 우선하는 경우

다음 요소가 적용되는 경우 *JMS* 바인딩을 고려하십시오.

- 메시징 시스템에 액세스해야 하는 경우
- 서비스가 느슨하게 결합된 경우
- 신뢰성이 성능보다 더 중요한 경우. 즉, 비동기 데이터 전송이 동기보다 우선하는 경우

다음 요소가 적용되는 경우 *Stateless* 세션 *EJB* 바인딩을 고려하십시오.

- 바인딩이 가져온 서비스용이며 서비스가 EJB 자체인 경우
- 가져온 서비스가 느슨하게 결합된 경우
- EJB의 상태가 중요하지 않은 경우

---

## 서비스 구현 유형

서비스 구현 유형은 서비스 컴포넌트의 구현입니다.

이 섹션에서 서비스의 표준 구현에 대해 설명합니다. 이러한 구현은 어셈블리 편집기의 서비스에서 또는 BPEL 프로세스에서 표시됩니다.

## Java 오브젝트

Java에서 컴포넌트의 구현을 Java 오브젝트라고 합니다.

한가지 일반적인 구현은 Java로 작성된 컴포넌트입니다. 이 구현은 때때로 "기존의 일반 Java 오브젝트" 또는 POJO라고 불립니다. 일반적으로 이 구현에 Java 인터페이스가 있는 경우라도 WSDL 인터페이스 유형이 있게 됩니다. 다중의 인터페이스가 지정되어 있는 경우, WSDL 인터페이스와 Java 인터페이스를 혼합할 수 없습니다. 그러나 WSDL 인터페이스 세트로 작성한 응용프로그램을 Java 인터페이스 세트가 있는 응용프로그램과 "결합"할 수 있습니다. 환영 보기의 샘플 갤러리에 나열된 샘플에서 해당 방법을 표시합니다.

Java 오브젝트에 대해 작업하는 경우, 코드는 편집기 컨텍스트에서 숨겨진 채로 표시되지 않습니다.

Java 오브젝트는 중개 모듈에서 사용할 수 있으며, WebSphere Process Server 또는 WebSphere Enterprise Service Bus 서버에 배치할 수 있습니다.

## BPEL 프로세스

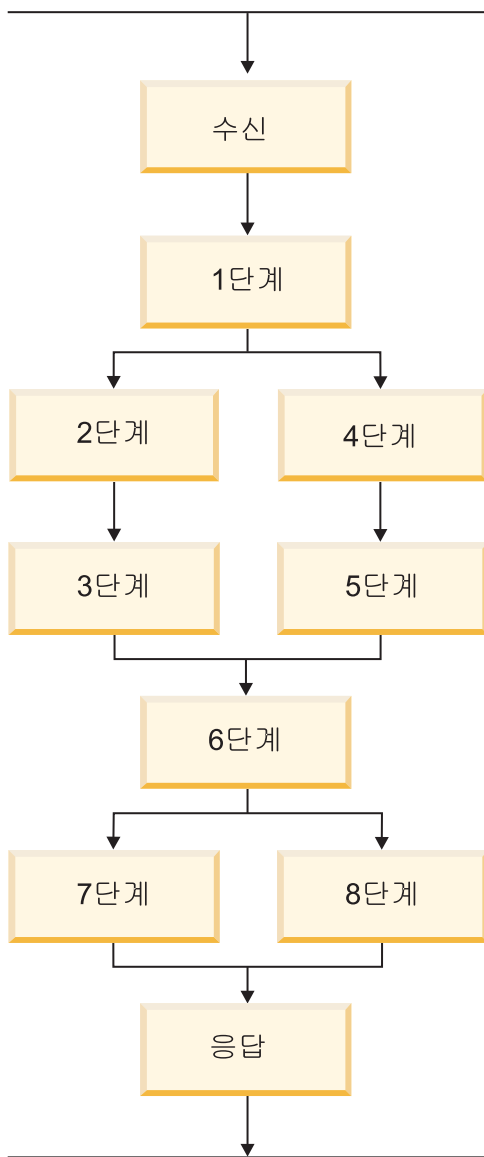
BPEL 프로세스 컴포넌트는 비즈니스 프로세스를 구현합니다.

해당 구현 언어는 업계 표준 BPEL4WS(Business Process Execution Language for Web Services)와 해당 IBM 확장입니다. BPEL 프로세스는 좀 더 많은 기본 웹 서비스 사용을 통해 오래 사용되는 상태 서비스를 구현합니다. 프로세스 편집기에서 작성한 BPEL 프로세스는 다음을 수행할 수 있습니다.

- 제어 플로우 그래프를 사용하여 기타 서비스 구성 설명
- 변수를 사용하여 프로세스 상태 유지
- 결합 처리를 통한 복잡한 오류 처리 사용
- 비동기 이벤트 지원
- 인스턴스를 식별하는 해당 요청 내에서 비즈니스 데이터를 표시하기 위해 상관 세트를 사용하여 특정 프로세스의 올바른 인스턴스와 인바운드 요청을 상관시킴
- 고급 보정 지원을 통해 확장 트랜잭션 제공

이러한 표준 BPEL 항목 이외에도, WebSphere Integration Developer에서는 사용자 타스크 지원을 통해 프로세스에 사용자를 포함시키도록 BPEL을 확장하기도 합니다. 예를 들어, 이 확장으로 사용자가 대출을 승인하는 요구사항을 프로세스에 추가할 수 있습니다.

프로세스 편집기는 BPEL 구성의 비주얼 표시를 사용하여 비즈니스 프로세스를 빠르고 단순하게 빌드합니다.



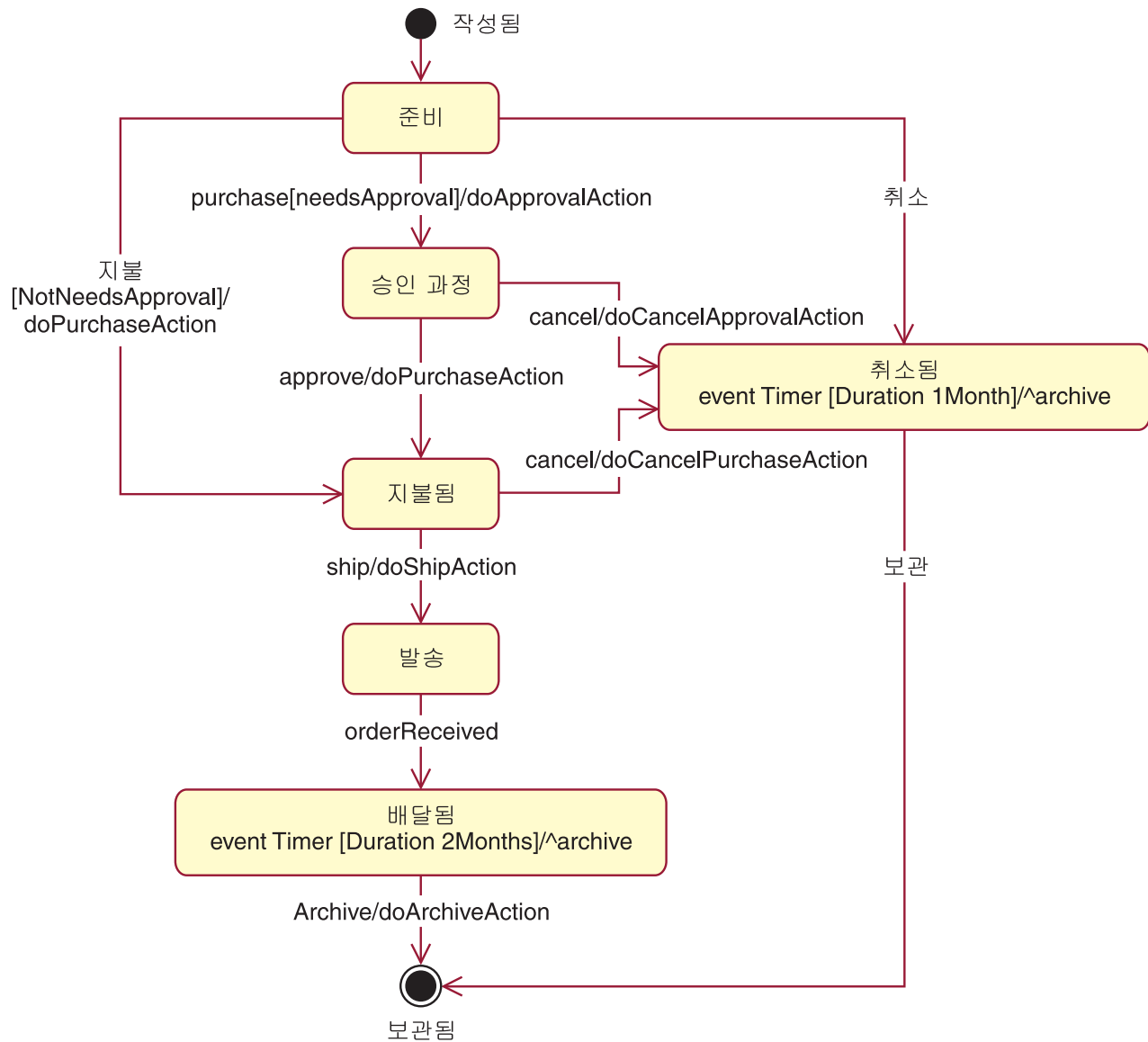
BPEL 프로세스를 중개 모듈에서 사용할 수 없으며, WebSphere Process Server에만 배치할 수 있습니다.

## 상태 머신

상태 머신은 비즈니스 프로세스를 작성하는 또다른 방법입니다. 상태 머신은 제어 플로우가 아닌 상태 변경과 관련된 프로세스에 적합합니다. 상태는 정해진 시간에 아티팩트가 수행할 수 있는 작업을 정의합니다. 상태 머신은 이러한 상태 세트의 구현입니다.

상태 머신은 진행 중인 상호 관련된 상태 세트를 표시하는 일반적인 방법입니다. 잘 알려진 상태 머신으로는 음료수 자판기가 있습니다. 머신에 동전을 넣으면 음료수 종류에 따라 상태 머신에서 사용자가 머신에 넣은 동전을 계산하여 정확한 거스름돈을 사용자에게 돌려줍니다. 아래 다이어그램에는 상태 머신 편집기에서 작성한 일반 상태 머신이 표시됩니다. 상태 머신에서는 항목을 구입하여 고객에게 제공합니다.





상태 머신을 중개 모듈에서 사용할 수 없으며, WebSphere Process Server에만 배치할 수 있습니다.

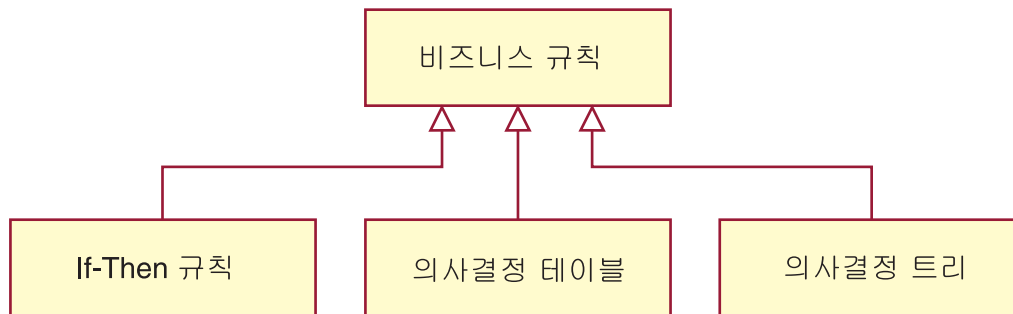
## 비즈니스 규칙

비즈니스 규칙은 비즈니스 프로세스 및 상태 머신을 보충합니다. 예를 들어, 변수를 사용하는 조건이 있을 경우 비즈니스 규칙은 해당 변수의 값을 런타임 시 변경할 수 있습니다. 비주얼 프로그래밍 언어로 작성되었으므로, 비즈니스 규칙은 컨텍스트에 따라 의사결정을 수행합니다. 의사결정은 단순하거나 또는 복잡할 수 있습니다. 비즈니스 규칙은 절차에 따르지 않으며 응용프로그램과 상관없이 규칙을 변경할 수 있습니다.

비즈니스 규칙은 컨텍스트에 기반하여 프로세스 결과를 판별합니다. 비즈니스 규칙은 일상적인 비즈니스 상황에서 사용되어 주어진 일련의 특정 상황에 대한 결정을 내립니다. 이 결정에는 모든 상황에 적용되는 많은 규칙이 필요할 수 있습니다. 비즈니스 프로세스 내의 비즈니스 규칙을 통해 응용프로그램이 변화하는 비즈니스 조건에 빠르게 대응할 수 있습니다. 예를 들어, 보험 회사에서 신청자에게 자동차 보험을 승인하는 비즈니스

규칙은 다음과 같을 수 있습니다. 신청자가 남성이고 25세 이상이며 자동차 카테고리가 스포츠 자동차이고 신청자가 지난 5년 동안 당사의 보험을 든 경우, 월 100달러에서 보험 응용프로그램을 승인합니다.

WebSphere Integration Developer는 비즈니스 규칙을 작성하는 다양한 접근 방식을 제공합니다. if-then 규칙 또는 의사결정 테이블을 작성할 수 있으며, 이들 모두 프로세스의 결과입니다. 이 규칙은 프로세스 자체와는 무관하다는 점을 참고하십시오. 즉, 프로세스를 다시 실행하지 않고도 언제든지 규칙을 변경할 수 있다는 의미입니다. 예를 들어, 비즈니스의 발생 장소에 따라 다음과 같은 규칙이 있을 수 있습니다. 날짜가 12월 26일과 1월 1일 사이인 경우, 20%의 휴가전 할인율을 제공합니다. 그러나 판매가 계속 저조한 경우 언제든지 할인율을 40%로 변경할 수 있습니다.

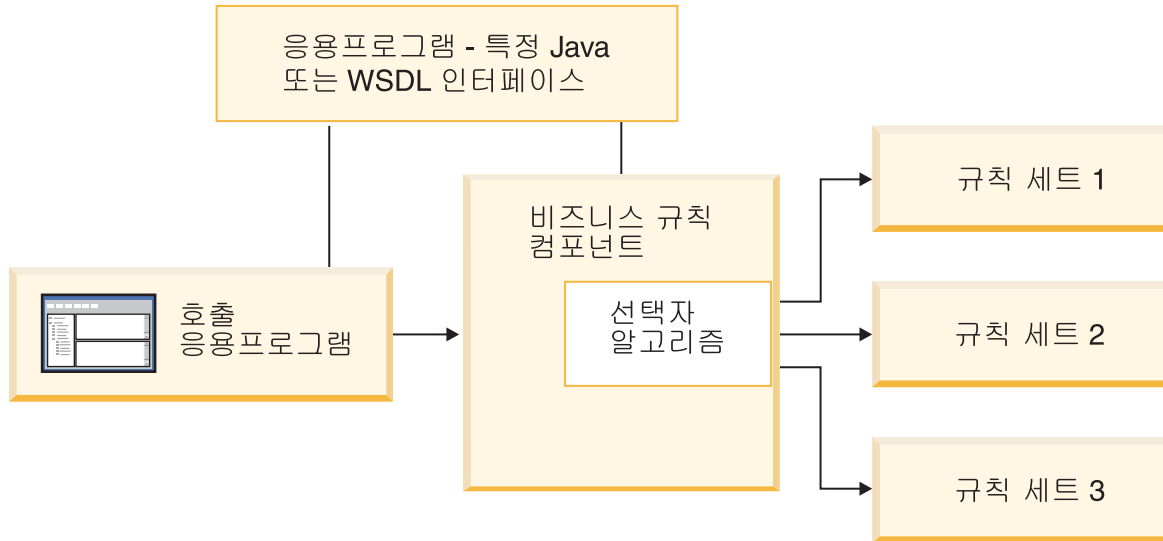


비즈니스 규칙을 중개 모듈에서 사용할 수 없으며, WebSphere Process Server에만 배치할 수 있습니다.

## 선택기

통합 응용프로그램은 여러 가지 상호작용 방법을 포함합니다. 선택기는 구현을 위해 오퍼레이션을 클라이언트 응용프로그램에서 일부 가능한 컴포넌트 중 하나로 라우팅하는 데 사용됩니다.

컴포넌트에 대한 라우팅은 날짜를 기반으로 합니다. 예를 들어 날짜를 기반으로 하는 다음과 같은 라우트가 있습니다. 개학 전 2주 동안은 학교 관련 상품을 특가로 제공합니다. 비즈니스에는 날짜를 기반으로 하는 많은 유사한 라우트가 있을 수 있습니다. 선택기는 날짜를 기반으로 런타임 시 다른 라우트 대신 하나의 라우트를 선택하기 위한 의사결정을 수행합니다. 예를 들어, 시간이 개학 바로 전인 경우 이전의 특가 판매가 제공됩니다. 그러나 학기가 끝나는 때에는 학생들이 여름을 준비하기 위한 판매를 제공할 것입니다.

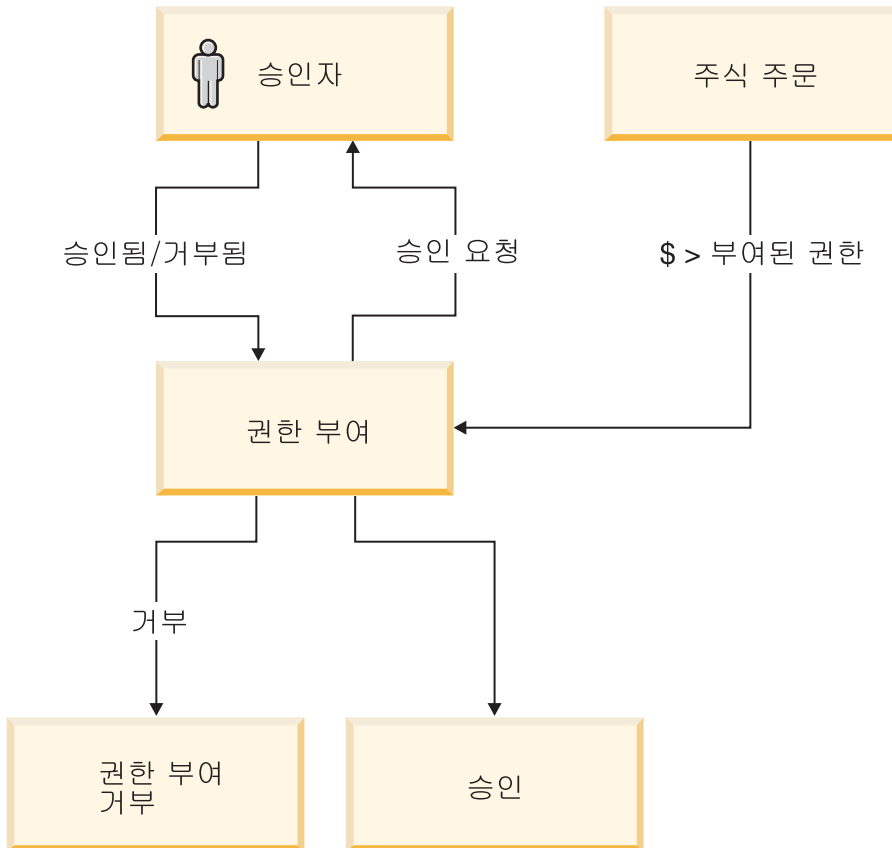


선택기는 중개 모듈에서 사용할 수 없으며, WebSphere Process Server에만 배치할 수 있습니다.

## 사용자 태스크

사용자 태스크 컴포넌트는 개인별로 수행해야 하는 태스크를 구현합니다. 이 태스크는 비즈니스 프로세스에서 사용자와의 관련성을 표시합니다.

종종 사용자가 비즈니스 프로세스에 간섭해야 하는 경우가 있습니다. 예를 들어, 고객이 신용 한도를 초과하는 항목을 구매하려고 합니다. 사용자 태스크를 통해 사용자는 고객의 구매 활동을 금지하는 비즈니스 규칙에 관여하여 해당 규칙을 무효화할 수 있습니다. 사용자 태스크에는 지정된 사용자가 사용 불가능한 경우 태스크 소유자 설정 및 에스컬레이션 프로세스 제공 등의 여러 속성을 가질 수 있습니다. 사용자 태스크 컴포넌트는 많은 프로세스에서 검토, 조사 및 승인등의 태스크에 대한 사용자 개입이 필요하다는 것을 인정합니다.



사용자 타스크를 중개 모듈에서 사용할 수 없으며, WebSphere Process Server에만 배치할 수 있습니다.

## 인터페이스 맵

인터페이스 맵에서 상호작용하는 컴포넌트 인터페이스 간의 차이점을 해결합니다.

서로 상호작용해야 하는 컴포넌트에서 인터페이스 간 차이점은 일반적입니다. 이러한 차이점은 WebSphere Integration Developer에서 종종 서로 다른 응용프로그램에 대해 작성된 컴포넌트의 조합으로 인해 발생합니다. 그렇지 않은 경우 유사한 컴포넌트를 다시 코딩해야 하므로, 이를 재사용하여 새 응용프로그램을 작성할 수 있는 기능은 WebSphere Integration Developer의 장점 중 하나입니다. 그러나 일반적으로 약간의 조정이 필요합니다.

예를 들어, 두 개의 컴포넌트에 기본적으로 동일한 조치를 수행하지만 `getCredit` 및 `getCreditRating`과 같이 서로 다른 이름을 갖는 메소드가 있을 수 있습니다. 또한 서로 다른 오퍼레이션 이름을 가질 수도 있으며 오퍼레이션의 매개변수 유형이 서로 다를 수도 있습니다. 인터페이스 맵은 해당 메소드의 오퍼레이션 및 매개변수를 맵핑하여 차이점을 해결하고 두 개의 컴포넌트가 상호작용할 수 있도록 합니다. 인터페이스 맵은 차이점에 도 불구하고 서로 연결될 수 있도록 허용하는 두 컴포넌트 인터페이스 간의 브릿지와 유사합니다.

인터페이스 맵은 사용 중인 컴포넌트에 상관없이 존재하며, 이는 컴포넌트 자체를 변경할 필요가 없음을 의미합니다.

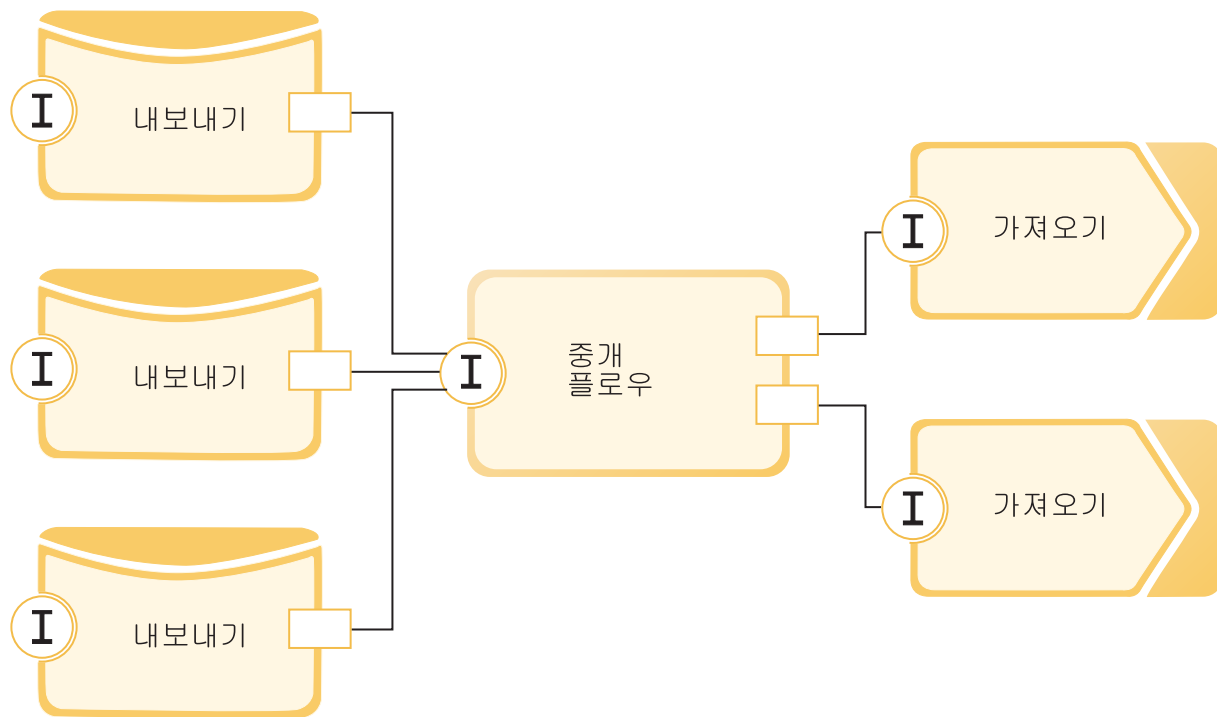
인터페이스 맵을 중개 모듈에서 사용할 수 없으며, WebSphere Process Server에만 배치할 수 있습니다.

## 중개 플로우

중개란 서비스 사이에서 동적으로 중개 또는 개입하는 방법입니다. 중개 플로우는 중개를 구현합니다.

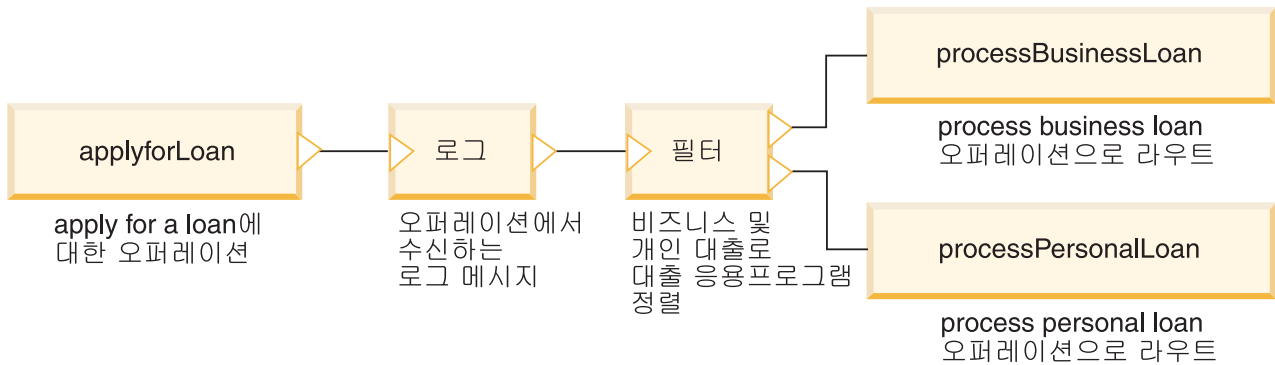
중개에는 여러 가지 유용한 기능이 있습니다. 예를 들어, 한 서비스에서 후속 서비스로 데이터를 수용 가능한 양식으로 변환할 때 중개를 사용할 수 있습니다. 메시지를 다음 서비스에 보내기 전에 서비스에서 메시지를 로깅할 수 있습니다. 한 서비스로부터 중개 플로우가 결정한 적절한 다른 서비스로 데이터를 라우팅할 수 있습니다. 중개는 연결된 서비스에 관계없이 독립적으로 작동합니다. 어셈블리 편집기에서 중개는 내보내기 및 가져오기 간의 중개 플로우 컴포넌트로 표시됩니다.

다음 다이어그램에서 세 개 서비스 요청자 또는 내보내기가 출력 데이터를 중개 플로우 컴포넌트의 인터페이스에 보냅니다. 그런 다음 중개 플로우 컴포넌트가 두 서비스 제공자 또는 가져오기에 적절한 데이터를 라우팅합니다.



중개 플로우는 중개 플로우 편집기를 사용하여 작성된 플로우와 유사한 구성입니다. 어셈블리 편집기에서 중개 플로우 컴포넌트를 선택하면 중개 플로우 편집기가 실행됩니다. 중개 플로우 편집기에서 한 서비스로부터의 오퍼레이션인 서비스 요청자 또는 내보내기는 중개 플로우 편집기가 제공하는 기능과 함께 다른 서비스인 서비스 제공자 또는 가져오기에 매핑됩니다. 이러한 기능을 중개 기능이라고 하며 다음 다이어그램에 표시된 대로 중개 플로우에서 연결됩니다. 중개 기능은 IBM에서 제공하며 또는 사용자 정의된 기능을 작성할 수 있습니다. 중개 기능은 메시지 콘텐츠 및 메시지 컨텍스트 둘 다에 작동하며, 여기서 컨텍스트란 SOAP 또는 JMS 헤더와 같은 바인딩 특정 정보 또는 사용자 정의 특성입니다.

다음 다이어그램에서 applyforLoan 오퍼레이션은 먼저 메시지를 기록하는 로그로 메시지를 보냅니다. 로그는 필터로 메시지를 보내고, 이 때 메시지에 따라 processBusinessLoan 오퍼레이션 또는 processPersonalLoan 오퍼레이션으로 메시지를 라우팅합니다.



모듈 섹션에서 설명한 바와 같이 중개 플로우 컴포넌트용 중개 모듈이 있습니다. 최대 하나의 중개 플로우 컴포넌트와 중개 플로우 컴포넌트의 기능을 증대시키는 0개 이상의 Java 컴포넌트를 포함할 수 있습니다. 중개 모듈은 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 서버에 배치할 수 있습니다.

## 독립형 참조

독립형 참조는 서비스 컴포넌트 아키텍처 컴포넌트로 정의되지 않은 응용프로그램에 대한 참조입니다(예를 들어, JavaServer Pages 또는 servlet). 독립형 참조는 해당 응용프로그램이 서비스 컴포넌트 아키텍처 컴포넌트와 상호작용할 수 있도록 허용합니다.

독립형 참조에는 인터페이스 또는 구현이 없습니다(구현이 모듈 범위를 벗어나므로). 모듈은 독립형 참조를 포함하지 않거나 또는 하나의 독립형 참조 아티팩트를 포함합니다. 독립형 참조에는 WebSphere Integration Developer에 작성된 서비스 컴포넌트 아키텍처 컴포넌트와 함께 기존 응용프로그램을 사용할 수 있게 해주는 실용적인 가치가 있습니다.

독립형 참조는 중개 모듈에서 사용할 수 있으며, WebSphere Process Server 또는 WebSphere Enterprise Service Bus 서버에 배치할 수 있습니다.

## 관련 정보

이 섹션에서는 아키텍처와 연관된 여러 가지 주제에 대해 설명합니다.

아래 주제에서는 제품의 아키텍처와 연관된 여러 가지 추가 정보를 제공합니다.

- 25 페이지의 『.NET 서비스와 함께 사용되는 개발 서비스』
- 25 페이지의 『양방향 지원』

## .NET 서비스와 함께 사용되는 개발 서비스

.NET 서비스와 함께 사용되는 서비스를 개발하려면 몇 가지 특수한 사항을 고려해야 합니다. 이러한 고려사항에는 두 가지 면이 있습니다. .NET 개발 환경에서 개발된 WSDL 파일을 WebSphere Integration Developer 개발 환경으로 가져오는 것과 WebSphere Integration Developer에서 WSDL 파일을 내보내기하여 .NET 서비스와 함께 사용할 수 있도록 하는 것입니다. 두 가지 개발 환경의 주요 차이점은 .NET 환경에서는 인라인 스키마를 사용하고 WebSphere Integration Developer에서는 사용하지 않다는 점입니다. 인라인 스키마는 스키마를 별도의 파일로 가져오도록 지정하지 않고 WSDL 파일 내에 포함시키는 방법입니다. WebSphere Integration Developer에서 제공하는 정보는 두 경우 모두에 도움이 됩니다.

인터페이스 작성 섹션의 인라인 스키마에는 .NET WSDL 파일과 같이 인라인 스키마를 포함하고 있는 WSDL 파일을 가져오는 방법이 설명되어 있습니다.

비즈니스 서비스 어셈블 섹션의 .NET 서비스에 대해 작업할 프록시 작성에는 .NET 서비스에 대해 작업하는 것과 같은 방식의 서비스를 외부적으로 공개하는 방법에 대해 설명합니다. 이 주제에서는 특히 .NET 서비스에 대해 작업할 프록시 작성을 중점적으로 다룹니다.

## 양방향 지원

WebSphere Integration Developer는 다국어 환경에서 작동합니다. 즉, 다양한 언어로 표시되는 데이터를 표시하고 조작합니다. 양방향 스크립트(아랍어 또는 히브리어)가 있는 일부 언어가 포함됩니다. 이들 언어는 오른쪽에서 왼쪽으로 쓰는 반면 라틴(또는 키릴 자모, 그리스어 등) 텍스트의 숫자 및 임베디드 세그먼트는 이 텍스트의 왼쪽에서 오른쪽으로 임베드됩니다.

IBM WebSphere Integration Developer의 양방향 스크립트 지원 개요에서는 양방향 언어 지원에 대해 설명합니다. 설명 내용에는 필수 구성, 지원을 사용할 때 해당되는 특정한 기술적 요점 및 제한사항이 포함됩니다.





---

## 제 3 장 도구에 대한 정보 학습

WebSphere Integration Developer는 비주얼 도구를 사용하여 서비스 지향의 아키텍처를 기반으로 하는 통합 응용프로그램을 작성합니다.

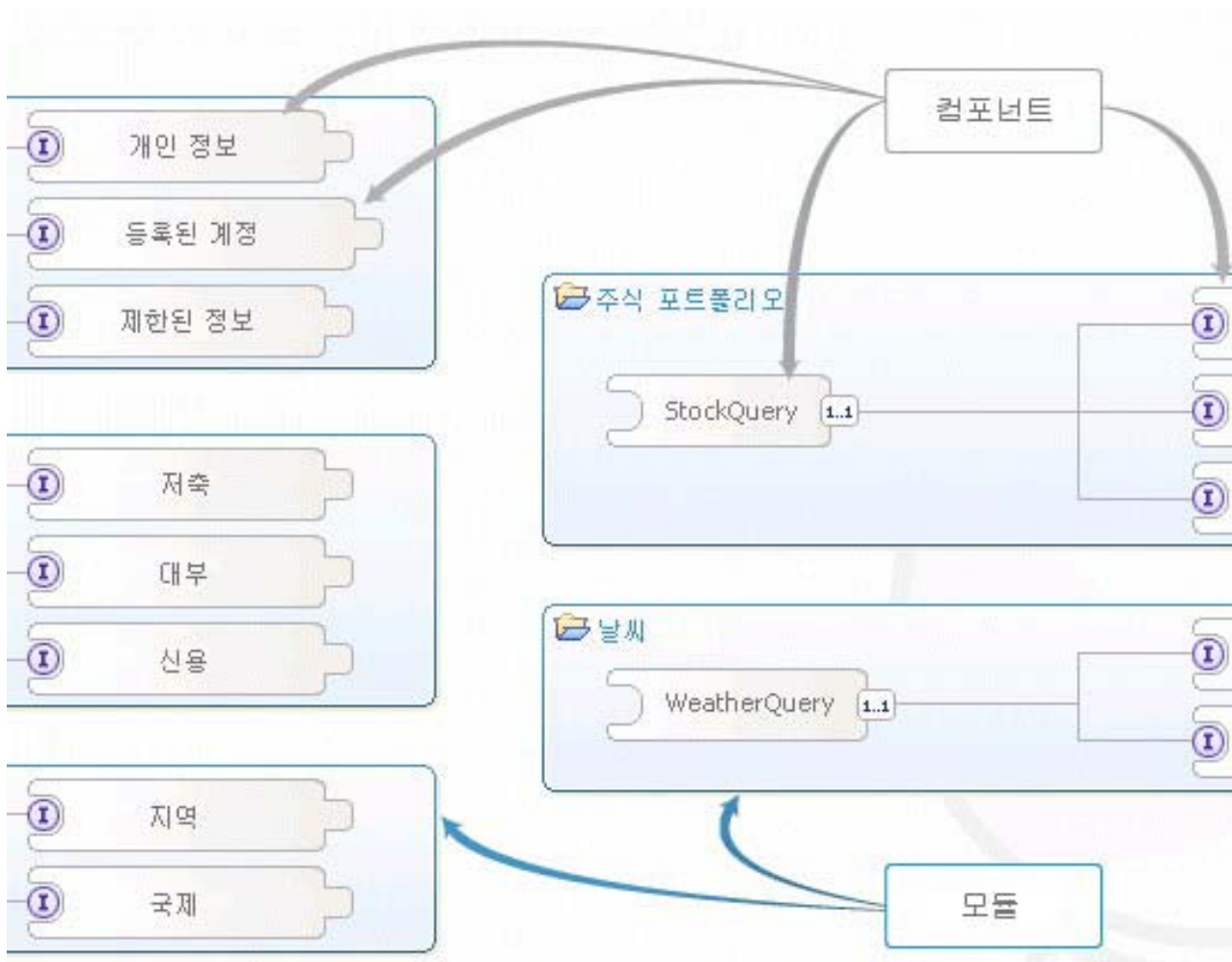
환영 보기의 개요 섹션에서는 WebSphere Integration Developer 도구를 이해하는 최상위 레벨을 제공합니다. 이 경우, 최상위 레벨은 도구의 가장 추상적인 프리젠테이션을 의미합니다. 그래픽에서는 도구가 수행하는 작업에 대해 설명합니다. 제품을 처음 접하고 해당 도구의 수행 작업에 대한 빠른 이해를 도우려면 개요 섹션을 읽으십시오. 학습서 섹션은 '환영' 보기에서도 참조할 수 있으며 조치에 대한 도구를 볼 수 있습니다. 샘플 섹션을 통해서도 실패 시 안전한 방식으로 도구를 사용하여 시작할 수 있습니다. 마지막으로 WebSphere Integration Developer로 응용프로그램 개발을 시작하는 경우, Information Center에서 개념 정보, 수행할 수 있는 task 및 참조 정보를 포함하여 각 도구에 대한 자세한 정보를 참조할 수 있습니다.

---

### 환영 보기 개요

환영 보기의 개요 섹션은 제품에 있는 도구 및 기능에 대한 최상위 레벨의 보기입니다. 개요에서 그래픽은 제품의 핵심 도구가 수행하는 작업을 전달할 때 사용됩니다.

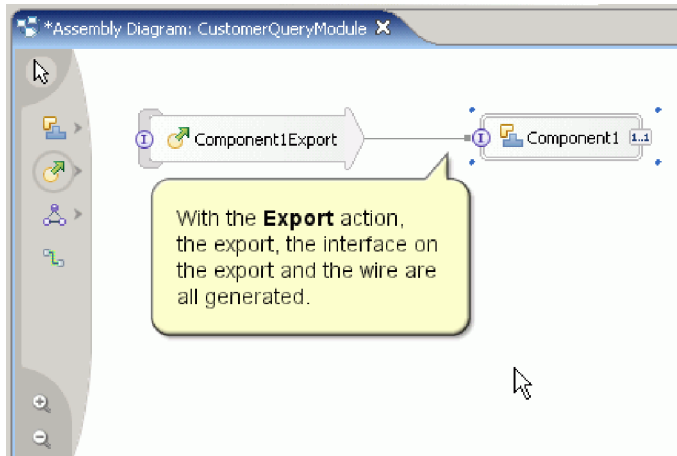
환영 보기의 개요 섹션은 체험 방식으로 제품에 대한 작업을 시작하기 전에 제품을 빠르게 학습하는 방법입니다. 제품을 처음 사용하는 대부분의 사용자들은 환영 보기부터 시작하십시오. 개요에서 아이콘을 선택하면 도구 또는 주제에 대한 그래픽이 실행됩니다. 예를 들어, 다음 다이어그램에서는 컴포넌트와 모듈의 관계를 표시합니다. 또한 제품에 대한 시청각 둘러보기를 실행할 수도 있습니다.



## 환영 보기의 학습서

학습서를 통해 실패 시 안전한 방식의 조치로 도구를 확인할 수 있습니다. 학습서를 선택하여 일부 TASK의 진행 과정을 살펴볼 수 있습니다.

제품의 개념을 이해하면 학습서를 실행하여 도구를 다룰 수 있습니다. 각 학습서는 특정 도구를 영화처럼 표시합니다. 학습서를 시청할 수 있을 뿐만 아니라 진행 내용을 풍선 텍스트로 설명합니다. WebSphere Integration Developer 학습서는 환영 보기에 있는 학습서 갤러리의 동영상 시청을 통한 학습 섹션에 있습니다.



## 환영 보기의 샘플

샘플을 사용하여 실패 시 안전한 환경에서 WebSphere Integration Developer로 비즈니스 서비스 솔루션을 편리하게 개발할 수 있습니다.

샘플은 환영 보기의 샘플 갤러리에서 찾을 수 있습니다. 샘플을 선택하면 여러 옵션이 제공됩니다. 단계별 지시사항에 따라 사용자 스스로 샘플을 빌드하거나 미리 빌드된 샘플을 사용할 수도 있습니다.

샘플은 범위에 따라 다양합니다. 기술 샘플은 특정 작업을 수행하는 특정 도구 수행에 초점을 맞춥니다. 응용프로그램 샘플은 여러 도구를 사용하여 보다 복잡한 결과를 얻습니다. 시나리오는 많은 도구로 대형 응용프로그램을 빌드하는 매우 긴 샘플입니다. 각 샘플에서는 샘플 빌드에 소요되는 시간을 표시합니다.

### Business state machine (simple)

This sample demonstrates how a business state machine can be used to moderate a sales order transaction. Specifically, the state machine emulates an on-line brokerage that manages the selling of a share.

The following tools are used in this application:

- Business state machine editor
- Human task editor

To import the ready-made sample, click the **Import** link below and click **Finish** in the opened wizard. See **Running instructions** to run the imported code.

or

If you want to build the sample for yourself, click **Step-by-step instructions**.

#### Ready-made sample

[Import](#) [Running instructions](#)

#### Build it yourself

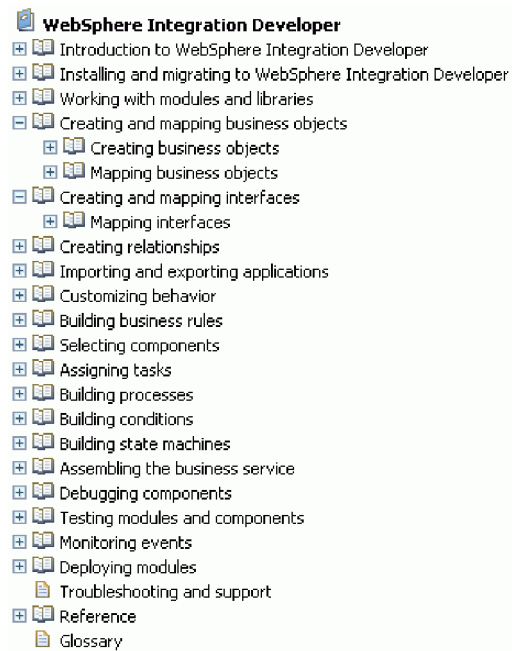
[Step-by-step instructions](#)

---

## Information Center

Information Center에서는 제품에 대한 전체 정보를 제공합니다. 학습서와 샘플 이외에도 WebSphere Integration Developer 각 도구에 대한 추가 개념, 타스크 및 참조 정보를 찾을 수 있습니다.

Information Center 탐색에서 원하는 정보에 빠르게 액세스할 수 있습니다. 또한 전체 Information Center를 검색할 수도 있습니다. 검색 결과에서 관련 정보 링크 이외에도 주제에 대한 자세한 정보를 포함하는 다양한 주제가 표시되도록 선택할 수 있습니다.



---

## 주의사항

본 IBM® 제품에 포함된 The XDoclet Documentation은 사용 허가를 취득하였으며 다음 저작권 표시가 적용됩니다. Copyright (c) 2000-2004, XDoclet Team. All rights reserved.

Portions based on *Design Patterns: Elements of Reusable Object-Oriented Software*, by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Copyright (c) 1995 by Addison-Wesley Publishing Company, Inc. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-270  
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩  
한국 아이.비.엠 주식회사  
고객만족센터  
전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통고없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(1) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (2) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 프로그램 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩  
한국 아이.비.엠 주식회사  
고객만족센터

이러한 정보는 해당 조항 및 조건에 따라(예를 들면, 사용료 지불 포함) 사용할 수 있습니다.

본 문서에 기술된 라이선스가 있는 프로그램 및 이 프로그램에 대해 사용 가능한 모든 라이선스가 있는 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

여기에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 레벨 상태의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한, 일부 성능은 추정치일 수도 있으므로 실제 결과는 다를 수 있습니다. 이 문서의 사용자는 해당 데이터를 사용자의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM의 향후 방향 또는 의도에 관한 모든 언급은 별도의 통지없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지

않습니다. 귀하는 IBM의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이러한 샘플 응용프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

(C) (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp의 샘플 프로그램에서 파생됩니다. (C) Copyright IBM Corp. 2000, 2005. All rights reserved.

이 정보를 소프트카피로 보는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

## **프로그래밍 인터페이스 정보**

프로그래밍 인터페이스 정보는 본 프로그램을 사용하는 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

귀하는 범용 프로그래밍 인터페이스를 통해 본 프로그램 톨의 서비스를 제공하는 응용프로그램 소프트웨어를 작성할 수 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용 프로그램 소프트웨어의 디버그를 돕기 위해 제공된 것입니다.

**경고:** 본 진단, 수정 및 조정 정보는 변경될 수 있으므로 프로그램 인터페이스로서 사용할 수 없습니다.

## **상표 및 서비스표**

다음 웹 사이트, <http://www.ibm.com/legal/copytrade.shtml>을 참조하십시오.





☺ 설문에 답해 주셔서 감사합니다.  
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해  
드리기 위해 최선을 다하겠습니다.







SA30-2704-01

