

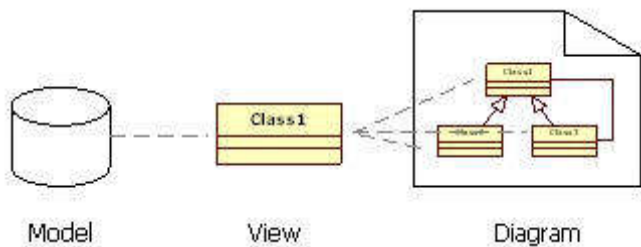
제 2 장. 기본 개념

이 장에서는 StarUML™을 효과적으로 사용하기 위해 필요한 필수적인 개념들을 소개합니다. StarUML™에서 사용되는 기본 개념인 모델과 뷰 그리고 다이어그램에 대해 알아보고, 프로젝트와 유닛, 모델 조각의 운용방법을 기술합니다. 또 확장 모듈과 거기에 포함되는 접근법, 프로파일, 프레임워크 등에 대한 개략적인 개념을 설명합니다.

- 모델, 뷰, 그리고 다이어그램
- 프로젝트와 유닛
- 모듈

모델, 뷰, 그리고 다이어그램

StarUML™에서는 모델과 뷰 그리고 다이어그램의 개념을 서로 분리해서 사용합니다. 우선, **모델(Model)**은 소프트웨어 모델에 관한 정보를 담고 있는 요소를 의미합니다. **뷰(View)**는 모델이 담고 있는 정보를 시각적으로 표현한 것을 의미하며, **다이어그램(Diagram)**은 뷰 요소들의 집합으로써 사용자의 일정한 생각을 표현한 것을 의미합니다.



프로젝트와 유닛

프로젝트

프로젝트(Project)는 StarUML™에서 다루는 가장 기본이 되는 단위를 의미합니다. 프로젝트는 하나 혹은 그 이상의 소프트웨어 모델들을 관리할 수 있으며 항상 존재하는 최상위 패키지(Top-level Package)로도 이해될 수 있습니다. 하나의 프로젝트는 일반적으로 하나의 파일에 저장됩니다.

프로젝트 구성

프로젝트 하위에는 다음의 요소들이 포함되며 관리될 수 있습니다.

프로젝트 하위요소	설명
모델(Model)	하나의 소프트웨어 모델을 관리하기 위한 요소입니다.
서브시스템(Subsystem)	하나의 서브시스템을 표현한 모델들을 관리하기 위한 요소입니다.
패키지(Package)	요소들을 관리하기 위한 가장 일반적인 요소입니다.

프로젝트 파일

프로젝트 파일은 XML 형태로 저장되며 확장명은 ".UML" 입니다. StarUML™에서 작성된 모든 모델, 뷰, 다이어그램들은 하나의 프로젝트 파일에 저장됩니다. 프로젝트를 여러 파일에 나누어 저장하려면 유닛을 사용할 수 있습니다. 프로젝트 파일에는 다음과 같은 정보들이 저장됩니다.

- 프로젝트가 사용하는 UML 프로파일들
- 프로젝트가 참조하는 유닛 파일들
- 프로젝트에 포함된 모든 모델 정보
- 프로젝트에 포함된 모든 다이어그램 및 뷰 정보

유닛

프로젝트는 기본적으로 하나의 파일에 저장되지만 프로젝트를 여러 명이 작업하거나 하는 등의 이유로 여러 개의 파일로 나누어서 다루어야 할 경우가 있습니다. 이러한 경우, 프로젝트를 여러 개의 유닛(Unit)으로 만들어서 다룰 수 있습니다. 유닛은 계층적으로 구성될 수 있어서 유닛의 하부에 여러 개의 서브 유닛을 가질 수도 있습니다. 유닛은 .UNT 파일에 저장되며, 프로젝트 파일(.UML) 혹은 다른 유닛 파일(.UNT)에서 참조됩니다.

유닛의 단위

패키지(Package), 서브시스템(Subsystem) 그리고 모델(Model) 요소만이 하나의 유닛(Unit)이 될 수 있습니다. 이러한 패키지 류 요소들의 하위에 포함된 모든 요소들은 해당 유닛 파일(.PUX)내에 저장됩니다.

유닛 계층구조

프로젝트가 하위에 여러 개의 유닛을 관리할 수 있는 것 처럼, 유닛 자체도 하위에 여러 개의 유닛을 관리할 수 있습니다. 상위 유닛은 하위 유닛들에 대한 참조를 가지게 되고 이러한 관계에 의하여 유닛은 계층구조(Hierarchical Structure)를 이루게 됩니다.

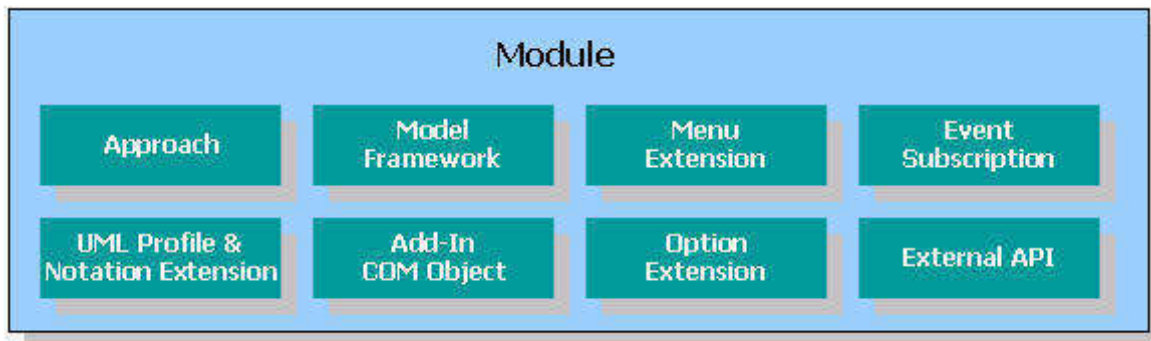
모델 조각

모델 조각은 프로젝트의 일부분을 별도의 파일로 저장한 것을 말합니다. 모델 조각의 대상은 모델(Model), 서브시스템(Subsystem), 패키지(Package)에만 해당되며 이것은 ".MFG"라는 확장명의 파일로 저장됩니다. 이렇게 저장된 모델 조각 파일은 언제든지 어떤 프로젝트에서 쉽게 포함시킬 수 있으며 일단 포함된 모델 조각은 프로젝트의 일부로 완전히 포함되므로 유닛과는 다소 차이가 있습니다.

모듈

모듈(Module)

모듈은 StarUML™을 확장하여 새로운 기능들과 특징들을 제공하기 위한 하나의 패키지입니다. 모듈은 아래의 <그림>과 같은 여러 확장 요소들의 조합으로 만들 수 있습니다. 또한 목적에 따라 하나의 확장 요소만으로 독립적인 모듈을 구성하거나, 모듈 내에 같은 유형의 확장 요소들을 여러 개 만들 수도 있습니다.



StarUML™ 모듈을 사용하면 다음과 같은 것들이 가능해 집니다.

- 메인 메뉴 혹은 팝업 메뉴의 확장
- 새로운 접근법의 추가
- 새로운 프레임워크의 추가
- 새로운 프로파일 추가
- 스테레오타입과 노테이션 확장을 통한 새로운 요소의 추가
- 새로운 기능에 대한 구현 (COM Server를 통해 혹은 단순한 Script 파일을 통해서 가능)
- 다른 애플리케이션들과의 연동
- 그 외의 확장팩 지원 기능

접근법

소프트웨어를 개발하기 위한 방법론은 수도 없이 많으며, 각 회사나 조직마다 독자적인 방법론을 가지고 있거나 이미 존재하는 것을 조금씩 다른 방법으로 사용하기도 합니다. 또한 개발할 소프트웨어에 대한 애플리케이션 영역(Application Domain)과 사용될 프로그래밍 언어와 플랫폼도 모두 다릅니다. 이러한 특성 때문에 소프트웨어 모델링 도구는 초기에 설정해야 할 사항들이 많아집니다. StarUML™에서는 이러한 내용들을 한번에 설정할 수 있는 접근법(Approach)이라는 개념을 제공합니다.

접근법의 구성

하나의 접근법은 다음의 항목들로 구성됩니다.

접근법 구성 항목	설명
프로젝트 구조(Project Structure)	시작할 프로젝트의 기본 구조를 설정합니다. 패키지, 서브시스템, 모델 요소들로 기본적인 구조를 만들고 적절한 위치에 다이어그램을 기본으로 배치시킬 수 있습니다.
사용할 프로파일들(Import Profiles)	시작할 프로젝트에 기본으로 사용할 UML프로파일들을 자동으로 포함시킵니다.
사용할 프레임워크들(Import Frameworks)	시작할 프로젝트에 기본으로 사용할 프레임워크들을 자동으로 로드하여 포함시킵니다.
읽어들일 모델 조각들(Import Model fragments)	시작할 프로젝트에 기본으로 삽입할 모델 조각들을 자동으로 읽어들이 특정 요소 하위에 포함시킵니다.

프레임워크

StarUML™에서의 프레임워크(Framework)는 클래스 라이브러리(Class Library)나 MFC, VCL, JFC 등과 같은 애플리케이션 프레임워크 등을 표현하는 소프트웨어 모델을 의미합니다. 프레임워크가 프로젝트에 포함되어 사용되어지면, 사용자는 특정 클래스 라이브러리나 애플리케이션 프레임워크에 의존하는 소프트웨어를 모델링하는데 있어서 매우 편리합니다.

프레임워크의 구성

프레임워크는 하나의 프레임워크 파일(.FRW)과 하나 이상의 유닛(.UNT)으로 구성됩니다.

구성요소	설명
프레임워크 파일(.FRW)	프레임워크 파일에는 어떤 유닛이 포함될지와 사용될 UML 프로파일에 대한 정보를 담고 있습니다.
유닛 파일(.UNT)	유닛 파일은 프레임워크에 대한 실질적인 모델 정보들을 담고 있습니다.

UML 프로파일

UML(Unified Modeling Language)은 어떠한 생각이나 개념도 표현할 수 있을 만큼 일반적입니다. 그러나, 그것은 또 다른 측면에서의 약점이 되기도 하는데, 특정 영역(Domain)의 개념들에 대해서는 정확하게 표현할 수 없습니다. 이러한 약점을 보완하기 위해서 UML을 확장할 수 있는 메커니즘을 제공하고 있는데, 그것이 바로 UML 프로파일(Profile)입니다. StarUML™은 UML 프로파일의 개념을 직접적으로 수용하여 UML을 쉽게 확장할 수 있도록 지원합니다.

UML 프로파일의 구성요소

UML 프로파일은 다음의 구성요소들로 이루어집니다.

구성요소	설명
스테레오타입(Stereotype)	스테레오타입은 특정 UML 요소에 붙여짐으로써 의미를 더 명확하게 하고 더불어 부차적으로 확장속성을 부여하여 더 정확한 모델링이 가능하도록 합니다. 스테레오타입은 그래픽 노테이션을 표현하기 위하여 아이콘 파일을 지정할 수도 있지만, 확장 노테이션 정의 파일(.PNX)를 사용하여 노테이션 도식 방법을 직접 정의할 수도 있습니다. 확장 노테이션에 관한 자세한 내용은 개발자 매뉴얼을 참고하십시오.
확장속성(TagDefinition)	확장속성은 UML 요소가 가지고 있는 프로퍼티만으로는 정확한 모델링이 불가능할 경우, 새로운 속성을 부여하여 부가적인 정보를 요소에 기록할 수 있도록 합니다. StarUML™에서 확장속성은 특정 스테레오타입에 포함되거나 아니면 독립적으로 존재할 수도 있습니다.
데이터타입(DataType)	프로파일이 기본적으로 포함하는 데이터타입들을 나타냅니다.
다이어그램타입(DiagramType)	다이어그램타입은 StarUML™에서 제안하고 있는 확장 요소로 다이어그램에 적용하여 새로운 다이어그램을 정의할 수 있도록 해 줍니다.
요소 프로토타입(ElementPrototype)	요소 프로토타입은 StarUML™에서 제안하고 있는 확장 요소로 기존에 정의된 요소에 속성들을 미리 설정하여 요소 생성을 위한 하나의 견본을 정의할 수 있도록 합니다. 이렇게 정의한 요소 프로토타입은 팔레트에 연결하여 요소를 직접 생성하거나 외부 API를 통해 요소를 생성할 수 있습니다.
모델 프로토타입(ModelPrototype)	모델 프로토타입은 StarUML™에서 제안하고 있는 확장 요소로 요소 프로토타입과 유사하나 모델에 국한되어 적용한 것입니다. 모델 프로토타입으로 정의한 요소는 모델 추가 메뉴에 표시됩니다.
팔레트(Palette)	팔레트는 StarUML™에서 제안하고 있는 확장 요소로 StarUML™에 기본적으로 포함되어 있는 팔레트 외에 추가적인 팔레트를 구성할 수 있도록 해 줍니다.

프로파일 작성에 대한 자세한 내용은 StarUML™ 개발자 가이드를 참고하십시오.

UML 프로파일의 응용

UML 프로파일은 다음과 같은 용도 등으로 응용될 수 있습니다. 그리고 OMG(Object Management Group)에서는 특정 용도를 위한 UML 프로파일의 표준을 제정하고 있으므로 그것들을 참고하실 수 있습니다.

- 특정 프로그래밍 언어(C/C++, Java, C#, Python)용 프로파일
- 특정 개발 방법론(RUP, Catalysis, UML Components, ...)을 위한 프로파일
- 특정 도메인(EAI, CRM, SCM, ERP, ...)을 위한 프로파일
-

모듈의 추가

사용자가 개발하거나 써드 파티 벤더에 의해 배포되는 모듈을 추가적으로 설치하면 확장된 기능을 StarUML™ 내에서 사용할 수 있습니다. 시스템에 새로 추가되는 모듈을 설치하는데 별도의 복잡한 등록과정이 필요치는 않습니다. 모듈을 설치하려면 해당 모듈을 <install-dir>\modules\ 하위 디렉토리에 서브 디렉토리를 만든 다음 모듈을 구성하는 파일들을 복사하면 됩니다.

StarUML™ 추가 제공 모듈

StarUML™은 StarUML™ 플랫폼 위에 동작하는 몇 가지의 모듈을 추가적으로 제공하고 있습니다.

- StarUML™은 UML 표준 프로파일과 몇 가지의 접근법, 시퀀스 및 협동 다이어그램간의 변환 등의 기능을 제공하는 기본 모듈(Standard Module)을 기본으로 제공합니다.
- 템플릿 기반으로 각종 문서와 코드를 생성할 수 있게 해주는 Generator 모듈을 제공합니다.
- Java 언어의 프로파일과 J2SE/J2EE 프레임워크, 코드 생성 및 역공학을 지원하는 Java 언어 모듈을 제공합니다.
- C++ 언어의 프로파일과 MFC 프레임워크, 코드 생성 및 역공학을 지원하는 C++ 언어 모듈을 제공합니다.
- C# 언어의 프로파일 .NET BCL 프레임워크, 코드 생성 및 역공학을 지원하는 C# 언어 모듈을 제공합니다.
- 모델 상호 교환을 위한 XMI 가져오기와 내보내기를 지원하는 XMI 모듈을 제공합니다.
- Rational Rose 파일을 읽어 오기 위한 Rose 모듈을 제공합니다.
- 디자인 패턴을 지원하는 패턴 모듈을 제공합니다.