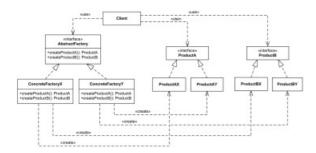
UML Class Diagrams Examples

Here we provide some examples of class diagrams and object diagrams:

──◇→ Abstract Factory Design Pattern

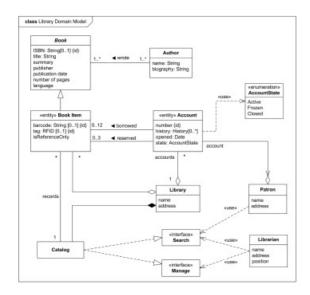
Purpose: Illustrate Abstract Factory design pattern.

Summary: Abstract Factory is creational software design pattern. This pattern provides interfaces for creating families of related or dependent objects without specifying their concrete classes.



Purpose: Describe domain area for an Integrated Library System (ILS), also known as a Library Management System (LMS) - Library, Catalog, Book, Patron, Account.

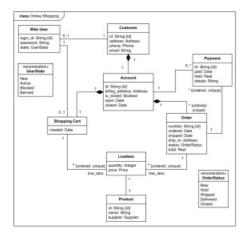
Summary: Library Domain Model describes main classes and relationships which could be used during analysis phase to better understand domain area for ILS or LMS.



Online shopping domain model

Purpose: Show some domain model for online shopping - Customer, Account, Shopping Cart, Product, Order, Payment.

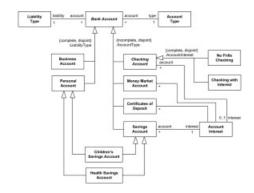
Summary: Example of a UML class diagram representing online shopping domain. Each customer could have some web user identity. Web user could be in one of several states and could be linked to a shopping cart.



□ Bank account class diagram example

Purpose: Domain model describing common types of bank accounts.

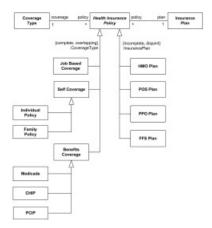
Summary: This example shows several subtypes of **Bank Account** using UML **generalization sets**. Bank accounts could be grouped into UML generalization sets based on different criteria. Example diagram shows bank accounts topology with two orthogonal dimensions and with corresponding **power types** Liability Type and Account Type.



→ Health insurance policy UML class diagram example

Purpose: Domain model describing various types of health insurance policies.

Summary: This example shows several subtypes of **Health Insurance Policy** using UML **generalization sets**. One generalization set is Coverage Type - Job Based Coverage, Self Coverage, and Benefits Coverage, and another set is based on Insurance Plan - HMO, POS, PPO, FFS.

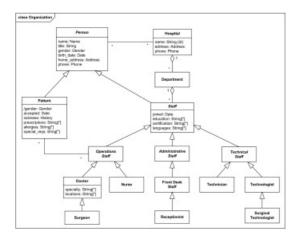


Purpose: Domain model for a hospital to show and explain hospital structure, staff, relationships with patients, and patient treatment terminology.

Summary: The domain model for the Hospital Management System is represented by several class diagrams.

Ward is a division of a hospital or a suite of rooms shared by patients who need a similar kind of care. In a hospital, there are a number of wards, each of which may be empty or have on it one or more patients. Each ward has a unique name.

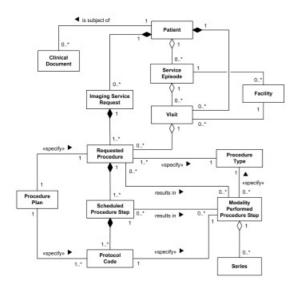
The doctors in the hospital are organised into teams (also called firms). Each team has a unique name or code (e.g. Orthopaedics or Pediatrics) and is headed by a consultant doctor or an attending physician.



Digital imaging in medicine - DICOM model of the real world

Purpose: Represent domain model ("model of the real world") for Digital Imaging and Communications in Medicine (**DICOM**) - Patient, Visit, Facility, Imaging Service Request, Scheduled Procedure Step, Modality Performed Procedure Step.

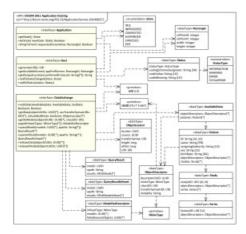
Summary: UML diagram example represents DICOM extended domain, abstract description of the real world objects used in the Modality-IS Interface. Modality is a piece of medical imaging equipment, e.g. computed tomography (CT) or ultrasound (US).



Digital imaging in medicine - DICOM Application Hosting API

Purpose: An example of UML class diagram representing **DICOM Application Hosting API**, defined in Part 19 of DICOM Standard (PS 3.19-2011). The Application Hosting API describes interfaces between two software applications - **Hosting System** and **Hosted Application**, exchanging medical data while located on the same system.

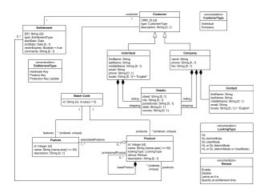
Summary: The DICOM Application Hosting API defines three interfaces - **Application**, **Host**, and **DataExchange** interface. Hosting System provides a variety of services such as DICOM object retrieval and storage to Hosted Application. The latter processes provided medical data, potentially returning back some newly generated data sets.



Sentinel HASP software licensing domain UML class diagram example

Purpose: The purpose of the domain diagram is to show major "things" used during software licensing and protection process using Sentinel HASP, and relationships between those things.

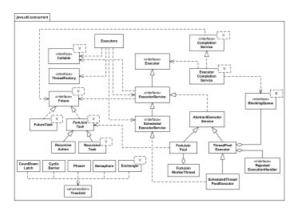
Summary: When software vendor purchases a Sentinel HASP LDK, the vendor is provided with a unique batch code and corresponding vendor key. Each protected software product has some features and is associated with a batch code. An entitlement can contain one or more products and is associated with the customer who placed the order. The customer could be either an individual customer or a company.



⇒ Java util.concurrent API UML class diagram examples

Purpose: Examples of UML **class diagram** representing most important interfaces and classes of **Java** $^{\text{TM}}$ **util.concurrent** API. Several java.util.concurrent.* packages support high-level concurrency features in Java with the new concurrent data structures in the Java Collections framework.

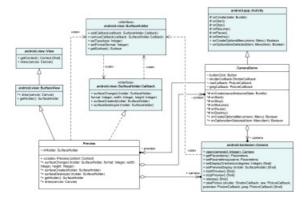
Summary: Executors define a high-level API for launching and managing threads to support large-scale applications mostly by adding thread pool management abilities. Concurrent collections reduce the need for synchronization and are designed to support concurrent access and modifications of the large collections of data. The Future < V> interface represents the result of an asynchronous computation.



Android Camera implementation classes

Purpose: An example of implementation level UML class diagram to illustrate usage of Android Camera API (Android 3.1 Platform, API Level 12).

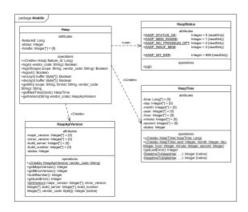
Summary: CameraDemo class extends Android's Activity class. An **Activity** is a single, focused thing that a user can do with Android. Activity usually interacts with user, and the Activity class takes care of creating a window in which we can place our user interface. CameraDemo activity will create a Preview object and will hold reference to. Preview holds back reference to the activity as its Context. The Preview object will create a Camera object and return it to the CameraDemo activity.





Purpose: Show implementation details of several HASP classes realizing the **HASP Java Native Interface Proxy component**.

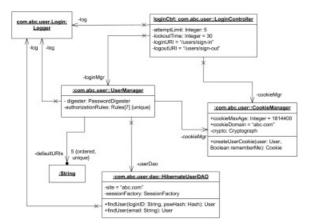
Summary: The HASP Aladdin package includes 4 classes. These classes are implementation of the HASP Java Native Interface Proxy component you can find on the **Sentinel HASP licensing component diagram**.



>> Web application Login Controller object diagram

Purpose: An example of UML object diagram which shows some runtime objects involved in the login process for a web user.

Summary: An instance of Login Controller class is associated with instances of User Manager, Cookie Manager, and Logger. Login Controller, User Manager, and Hibernate User DAO (Data Access Object) share a single instance of Logger.















This document describes UML versions up to $UML\ 2.5$ and is based on the corresponding OMG^{TM} Unified Modeling Language ($OMG\ UML\ B$) specifications. UML diagrams were created in $Microsoft\ B$ Visio\ 2007-2016 using $UML\ 2.x\ Visio\ Stencils$. Lucidchart is a nice, free UML tool that I recommend for students.

You can send your comments and suggestions to webmaster at webmaster@uml-diagrams.org.

 ${\it Copyright} © {\it 2009-2018} \ uml-diagrams.org. \ All \ rights \ reserved.$