**Boundary class**

The boundary class is a class used to model the interaction between the external environment of the system and its internal operation, which involves converting events and recording changes in the system representation (such as interfaces).  
 Boundary classes model the parts of the system that depend on the environment. The entity class and the control class model the part that is independent of the external environment of the system. Therefore, if you change the GUI or communication protocol, you will only change the boundary class and have no effect on the entity class and the control class.

Because of the explicit boundary of the system, boundary classes can help people understand the system more easily. At design time, they provide a good starting point for determining relevant services. For example, if a printer interface is identified at the beginning of the design, you will soon find that you must also model the format of the print output.

Common boundary classes have Windows, communication protocols, printer interfaces, sensors, and terminals. If you are making use of GUI generators, you do not model generic interface components such as buttons as separate border classes. Typically, the entire window is the most refined boundary class object. The boundary class also helps to get interfaces that might not be facing any object's API (such as legacy code).

You should model boundary classes based on the boundary type represented by the boundary class. Communicating with other systems and communicating with the main actors (through the user interface) is very different in purpose. In user interface modeling, the most important thing to focus on is how to display the interface to the user interface. In the system communication modeling, the most important concern is the communication protocol.

A boundary object (that is, an instance of a boundary class) can survive longer than the use case instance. For example, a boundary object must show up on the screen for a period of time between two use case executions. But usually the two are as long as they live.

Find border class boundary classes to help system interfaces interact with external systems. Boundary object of the system and its external environment change (changes to interface with other systems, the change of user requirements, etc.) separate, make these changes will not affect the rest of the system.

A system may have multiple boundary classes:

· user interface classes - classes that help communicate with system users

· system interface classes - classes that help communicate with other systems

· device interface class - class that provides interfaces for devices that monitor external events, such as sensors.

Find the user interface class

Indicates that boundary classes of user interfaces may exist during user interface modeling activities; These classes can be reused during this activity as long as appropriate. If you haven't already done user interface modeling, the discussion below will help you find these classes.

Each use case protagonist has at least one boundary class. It can be thought that this object is responsible for coordinating the interaction between the main characters. This boundary object has some helper objects, and the boundary objects delegate some of its responsibilities to these helper objects. This is especially true for window-based GUI applications. In these applications, each window or form corresponds to a boundary class.

Create a sketch of the user interface prototype or screen dump to show the behavior and appearance of the boundary objects.

Only the core parts of the system are modeled and not modeled on every button, list, and widget in the GUI. The purpose of the analysis is to get a general idea of how the system is structured, rather than to design every detail. In other words, you only need to determine the boundary class for some of the phenomena in the system or some of the things that are mentioned in the event flows that are implemented in the use case.

Find the system interface class

The boundary class that communicates with the external system manages the conversation with the external system, which provides an interface to the external system for the system being built.