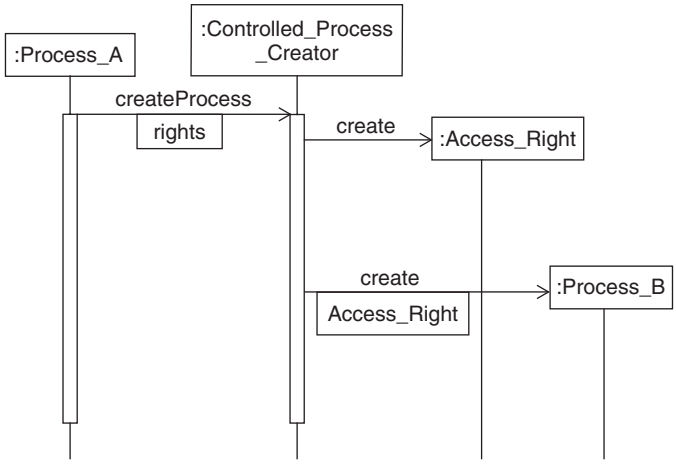
10.3 Controlled Object Factory

This pattern addresses how to specify the rights of processes with respect to a new object. When a process creates a new object through a factory (see FACTORY METHOD and ABSTRACT FACTORY [GoF95]), the request includes the features of the new object. These features include a list of rights to access the object.



Process creation dynamics

Example

In many operating systems the creator of an object gets all possible rights to the object. Other operating systems apply predefined sets of rights: for example, in Unix all the members of a file owner's group may receive equal rights for a new file. These approaches may result in unnecessary rights being given to some users, violating the principle of least privileges.

Context

A computing system that needs to control access to its created objects because of their different degrees of sensitivity. Rights for these objects are defined by authorization rules or policies that are enforced when a process attempts to access an object.

Problem

In a computing environment, executing applications need to create objects for their work. Some objects are created at program initialization, while others are created dynamically during execution. The access rights of processes with respect to objects must be defined when these objects are created, or there may be opportunities for the processes to misuse them. Applications also need resources such as I/O devices and others that may come from resource pools: when these resources are allocated, the application must be given rights to them.

The solution to this problem must resolve the following forces:

- Applications create objects of many different types, but we need to handle them uniformly with respect to their access rights, otherwise it would be difficult to apply standard security policies.
- We need to allow objects in a resource pool to be allocated and have their rights set dynamically: not doing so would be too rigid.
- There may be specific policies that define who can access a new object, and we need to apply these when creating the rights for an object. This is a basic aspect of security.

Solution

Whenever a new object is created, define a list of subjects that can access it, and in what way.

Structure

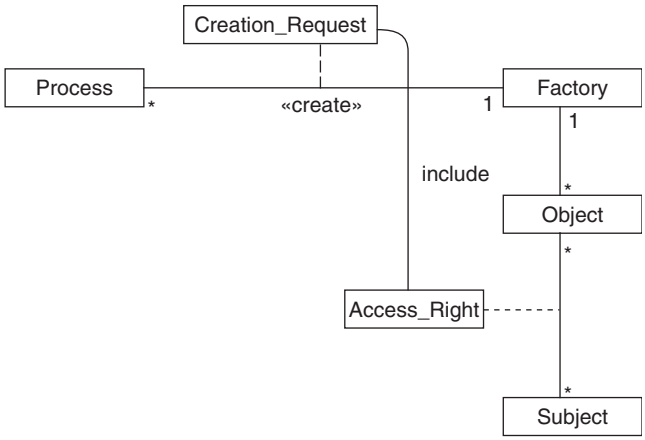
The figure on page 333 shows the class diagram for the solution. When a `Process` creates a new object through a `Factory`, the `Creation_Request` includes the features of the new object. Among these features is a list of rights that define the access rights for a `Subject` to access the created `Object`. This implies that we need to intercept every access request: this is done by CONTROLLED OBJECT MONITOR (335).

Dynamics

The figure on page 334 shows the dynamics of object creation. A process creating an object through a FACTORY defines the rights for other subjects with respect to this object.

Implementation

Each object may have an associated access control list (ACL). This will list the rights each user has for the associated object. Each entry specifies the rights that any other



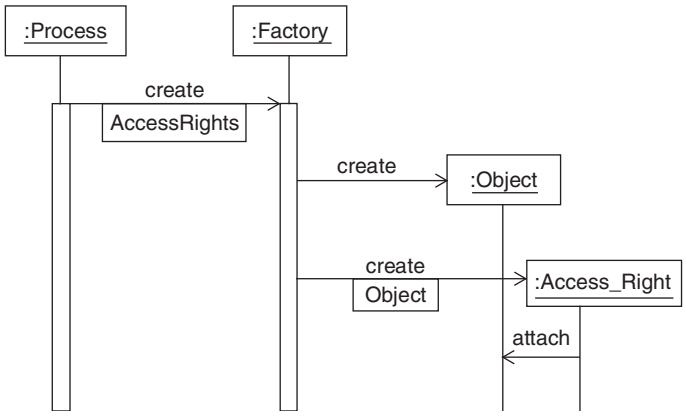
Class diagram for CONTROLLED OBJECT FACTORY

object within the system can have. In general, each right can be an 'allow' or a 'deny.' These are also known as Access Control Entries (ACE) in the Windows environment (see [Har01], [Mic00], and [Zac87]). The set of access rules is also known as the Access Control List (ACL) in Windows and most operating systems.

Capabilities are an alternative to an ACL. A capability corresponds to a row in an access matrix. This is in contrast to the ACL, which is associated with the object. The capability indicates to the secure object that the subject does indeed have the right to perform the operation. The capability may carry some authentication features in order to show that the object can trust the provided capability information. A global table can contain rows that represent capabilities for each authenticated user [And01], or the capability may be implemented as a lists for each user which indicates which object each user has access to. [Kin01]

Example Resolved

Our users can now be given only the rights to the created objects that they need. This prevents them from having too many (possibly unnecessary) object rights. Many misuses occur through processes having too many rights.



Object creation dynamics

Known Uses

The Win32 API allows a process to create objects with various `Create` system calls using a structure that contains access control information (DACL) passed as a reference. When the object is created, the access control information is associated with the object by the kernel. The kernel returns a handle to the caller to be used for access to the object.

Consequences

The following benefits may be expected from applying this pattern:

- There will be no objects that have default access rights because somebody forgot to define rights to access them
- It is possible to define access rights to an object based on its sensitivity
- Objects allocated from a resource pool can have rights attached to them dynamically
- The operating system can apply ownership policies: for example, the creator of an object may receive all possible rights to the objects it creates. The following potential liabilities may arise from applying this pattern:
- There is a process creation overhead
- It may not be clear what initial rights to define

See Also

BUILDER and other creation patterns [GoF95].