permissions of each such collection are implied[1] by the permissions of the corresponding or excluded collection of the other translation. Translation equivalence is only required with respect to the permission types that are the subject of the translation.

3.1.1 Policy Contexts and Policy Context Identifiers

It must be possible to define separate authorization policy contexts corresponding to each deployed instance of a Java EE module. This per module scoping of policy context is necessary to provide for the independent administration of policy contexts corresponding to individual application modules (perhaps multiply deployed) within a common Policy provider.

Each policy context contains all of the policy statements (as defined by this specification) that effect access to the resources[2] in one or more deployed modules. At policy configuration, a `PolicyConfiguration` object is created for each policy context, and populated with the policy statements (represented by permission objects) corresponding to the context. Each policy context has an associated policy context identifier.

In the Policy Decision and Enforcement Subcontract, access decisions are performed by checking permissions that identify resources by name and perhaps action. When a permission is checked, this specification requires identification of the authorization policy context in which the evaluation is to be performed (see Section 4.6, “Setting the Policy Context,” on page 52).

3.1.1.1 Policy Context Life Cycle

Figure 3.1 depicts the policy context life cycle as effected through the methods of the PolicyConfiguration interface. A policy context is in one of three states and all implementations of the PolicyConfiguration interface must implement the state semantics defined in this section.

- open

  A policy context in the open state must be available for configuration by any of the methods of the PolicyConfiguration interface. A policy context in the open state must not be assimilated at Policy.refresh into the policy statements

1. For some permission types, such as the EJBMethodPermission, it will generally not be possible to use the implies method of the PermissionCollection to compute collection equivalence (because the implies method is unable to determine when a collection contains all the permissions implied by a wild carded form of the permission).
2. An exception to this rule is described in Section 3.1.4, “EJB Policy Context Identifiers”.