defined by the particular role or target environment. The parameter is resolved by a reference to the privilege template by the specific roles, in this case a different value is provided for "WAS_{13} INSTALL" for each of the PayrollAdmin and HRAdmin roles, as well as the particular target platforms.

Having defined the privilege template, role definitions for two different applications that make use of the same application may then be defined. Users can be assigned roles in any number of ways, either by direct mapping of users to roles or potentially indirectly via group membership. The storage location of this mapping is immaterial to the function.

In the example of defining two roles in fragment **604**, statement **622** indicates that a role specification follows. As in the naming of the privilege template, a series of names follows the specification keyword of role. The hierarchy of names first specifies the role for /PayrollAdmin/Privileges/WAS. Statement **616** indicates the attribute of platform has a specified value applicable to the named platform of UNIX™, wherein UNIX is a registered trademark of The Open Group. In similar fashion, statement **618** provides the attribute of platform with an assigned value of Windows®, wherein Windows is a registered trademark of Microsoft Corporation in the United States, and other countries.

Statement **620** indicates use of a previously defined template named /WebSphere/AppServer. This template was previously defined in fragment **602** as a privilege template. However, this template is now being incorporated by reference into the definition for the role /HRAdmin/WAS and not /PayrollAdmin/Privileges/WAS, as defined in the first portion of fragment **604**.

From the example, users assigned to the PayrollAdmin role are conferred the privileges granted by the /WebSphere/AppServer privilege template, with UNIX and Windows environment parameters instantiated as shown. In this case, the definitions provide a user with the permission to run the startServer and stopServer commands with operating system identity payroll-admin on UNIX, or payroll-admin@servers.company.com on Windows. Definition of the HRAdmin role now requires only reference to the existing privilege template and added or changed definition of parameter values specific to the HRAdmin role.

A simple consequence of this representation is that roles may be defined hierarchically as well. For example, AllAdmin role **624** could be simply defined as having inherited all definitions below PayrollAdmin and HRAdmin, identical to the prior example and omitted here for brevity.

Prior solutions may also use hierarchies of roles as a way of simplifying administration and inheriting privilege, but at a very coarse level. The hierarchies of parameterized privileges just described represent low-level details of how privileges are actually implemented by resource providers, including operating systems, application servers, and other subsystems.

The common thread among prior solutions defined hierarchies of roles which reference privileges. The current described approach defines an intermediate object of the parameterized privilege template between the defined roles and defined privileges, which minimizes the amount of information needed to be managed across heterogeneous environments, and even within homogeneous environments. While the definition of a set of privileges is typical of a particular product or environment, that product or environment is instantiated multiple times, with each instantiation being administered separately.

With reference to FIGS. **7A-7B**, text representations of a generalized role specification, in accordance with illustrative embodiments are shown. Data structure **700** comprises a set of statements defining elements of a role. Data structure **700** has been designed to capture and use inheritance hierarchies to minimize the redundant representation of role and privilege information. The statements are presented as combinations of a keyword or identifier portion and an attribute with value portion. The set of definitions provide a data model for the specification of policies in an environment independent format.

Statement **702** has a keyword of role identifying the statement as a statement that begins a set or collection of related statements that defines a role. In the example statement, the role may have an assigned name or refer to another template from which to inherit definitions.

Statement **704** defines the environment to which the role being defined is applicable of the "applies-to-keyword". For example, named environments may typically be based on an operating system platform or a sub-system definition, such as a database representation or application. The statement provides a scope or boundary for defined privileges.

Statement **706** provides a format for the provision of a privilege name and associated value. For example, privilege-parameter-name **708** identifies the attribute of interest, while privilege-parameter-value **710** provides the value assigned to the associated parameter. The combination provides a capability to define elements and associate values with the defined elements at a fairly granular level, if needed.

Statement **708** provides a capability to define an environment specific reference name for use within the role definitions. For example, when a widget is the appropriate term in one environment while object is the corresponding term in another environment, the correct term may be used in the specific environment. The input is used by a transformation utility to provide the necessary term to avoid an error.

Use of parameter-reference-intro **712** and parameter-reference-trailer **714** provide definitions of start and stop character sequences used in the definitions. The notation used with the definitions of FIG. **7** is as follows: "A: B C" means "A" consists of elements "B" and "C, A: B|C" means "A" consists either element "B" or element "C, A: B*" means "A" consists of zero or more "B" elements, "A: B+" means "A" consists of one or more "B" elements, "A: B?" means "A" consists of zero or one "B," "A: (B)" means "A" consists of "B", and "A: "string"" means "A" is the string literal "string." The use of the parentheses indicates the content within takes precedence over everything else. Use of the "or" symbol "|" has the lowest precedence of all constructs.

In the schema definitions just described, parameter values are resolved recursively. For example, using the following privilege template sample reference:

```
/WAS
    Applies-To: All
    Command: ${WAS_INSTALL}/bin/startServer.sh
```

with a role definition fragment of:

```
/HR-WAS
    WAS_INSTALL:   ${WAS_ROOT}/profile/HR
    UNIX/
        WAS_ROOT: /opt/IBM/WebSphere/AppServer
    Windows/
        WAS_ROOT: C:\IBM\WebSphere\AppServer
```

WAS_INSTALL is defined in terms of WAS_ROOT which is further defined at a lower level of hierarchy. The transform