Pegasus Technical Workshop July 2003

Associations Overview

July 21 2003

Karl Schopmeyer, k.schopmeyer@opengroup.org

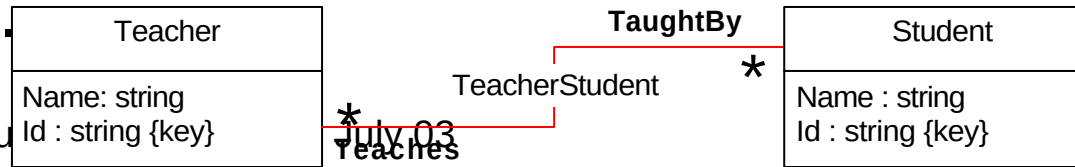
Agenda

THE Open GROUP

- Overview of Associations
  - Specification
  - Operations
- State of Associations in Pegasus
- Architecture of Associations
- The Pegasus Association Provider APIs
- Implementing Association Providers

Definition of Associations

- An association is a class that contains two or more references.
- An association represents a relationship between CIM objects
- Relationships (associations) can be established without any affect on the referenced classes
- Only associations can have references
- An association cannot be a subclass of a non-association class and any subclass of an association is an association
- An association is a type of class, a class with the association qualifier.



Definition of Associations

THE Open GROUP

- Reference Names must be unique within the scope of their defining Association.
- Reference Names obey the same rules as Property Names. Reference names are not required to be unique within the scope of the related Class. In such a scope, the Reference provides the name of the Class within the context defined by the Association.

System System System
Hosted Services Dependency
Service Service Service Service

Figure 2-2 Reference Naming

It is legal for the class System to be related to Service by two independent Associations (*Dependency* and *Hosted Services,* each with roles *System* and *Service*). It would not be legal for *Hosted Services* to define another Reference *Service* to the Service class, since a single association would then contain two references called *Service.*

References

THE Open GROUP

- *References* define the role each object plays in an Association.
- The Reference represents the role name of a Class in the context of an Association.
- Associations support the provision of multiple relationship instances for a given object.
- Properties which are links to other objects
- Value is a string that represents path to another object and includes:
  - Namespace for object
  - Class name of object
  - If object is instance, values of all key properties
- Declared in mof with the definition
  - `ClassName ref ReferenceName;`

```
[Association]
class TST_TeacherStudent {
TST_Teacher ref Teaches;
TST_Student ref TaughtBy;
};
```

Cardinality

THE Open GROUP

- A relationship between two classes that allows more than one *object* to be related to a single *object*.
- In associations, object references have cardinalities - denoted using Min and Max qualifiers.
  - Max - Indicates the maximum cardinality of the reference (i.e. the maximum number of values a given reference can have for each set of other reference values in the association). For example, if an association relates A instances to B instances, and there must be at most one A instance for each B instance, then the reference to A should have a Max(1) qualifier.
  - Min - Indicates the minimum cardinality of the reference (i.e. the minimum number of values a given reference can have for each set of other reference values in the association). For example, if an association relates A instances to B instances, and there must be at least one A instance for each B instance, then the reference to A should have a Min(1) qualifier.

Weak Associations

THE Open GROUP

Aliases

THE Open GROUP

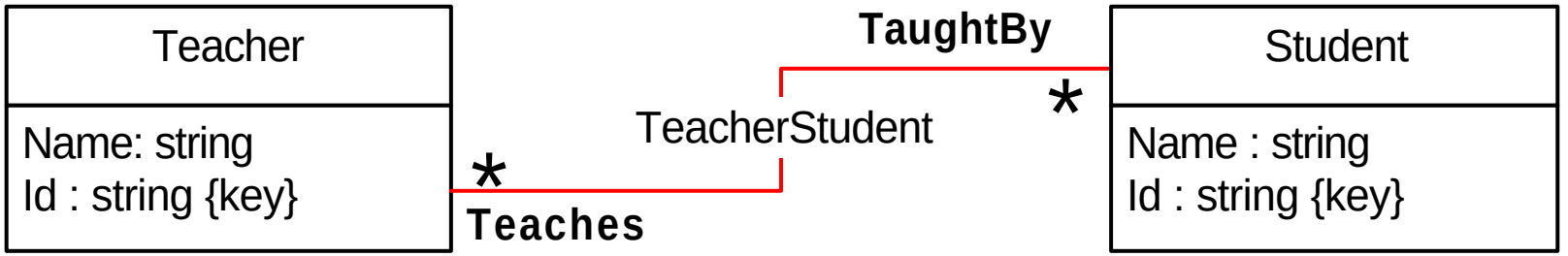
- Symbolic references to an object located elsewhere in the MOF specification.
- Aliases only have significance within the MOF specification in which they are defined, and are only used at compile time to facilitate establishment of references.

An alias can be assigned to an instance using this syntax:

```
instance of Acme_LogicalDisk as $Disk
{
// Body of instance definition here ...
};
```

```
instance of Acme_AnAssociation
{
strVal = "ABC";
obref1 = $Disk;
obref2 = $Alias2;
};
```

A Simple Example




```
class TST_Teacher
{
 [Version("1.0.0"), Description("People who "
    " teach courses")]
    string Name;
    [key, Description("Unique Id and the key")]
    string Id;
};

class TST_Student
{
 [Version("1.0.0"), Description("People who "
    " take courses")]    string Name;
    [key, Description("Unique Id and the key")]
    string Id;
```

```
 [Association, Version("1.0.0"),
     Description("Top level association between "
         "students and teachers.")]
class TST_TeacherStudent
{
    [key, Description (" ")]
    TST_Teacher ref Teaches;

    [key, Description (" ")]
    TST_Student ref TaughtBy;
};
```

Simple Instance Example

THE Open GROUP

Teacher	Teacher	Student	Student
Name : "John Smith" Id : "USC:000010002"	Name : "Tim Jones" Id : "USC:000020001"	Name : "Jane Doe" Id : "USC:001012111	Name : "Joe Nerd" Id : "USC:001020200

TeacherStudent
Teaches : "HTTP://CIMOM host/root:Student.Id=USC:001012111" TaughtBy : "HTTP://CIMOM host/root:Teacher.Id=USC:000010002"

1st Instance

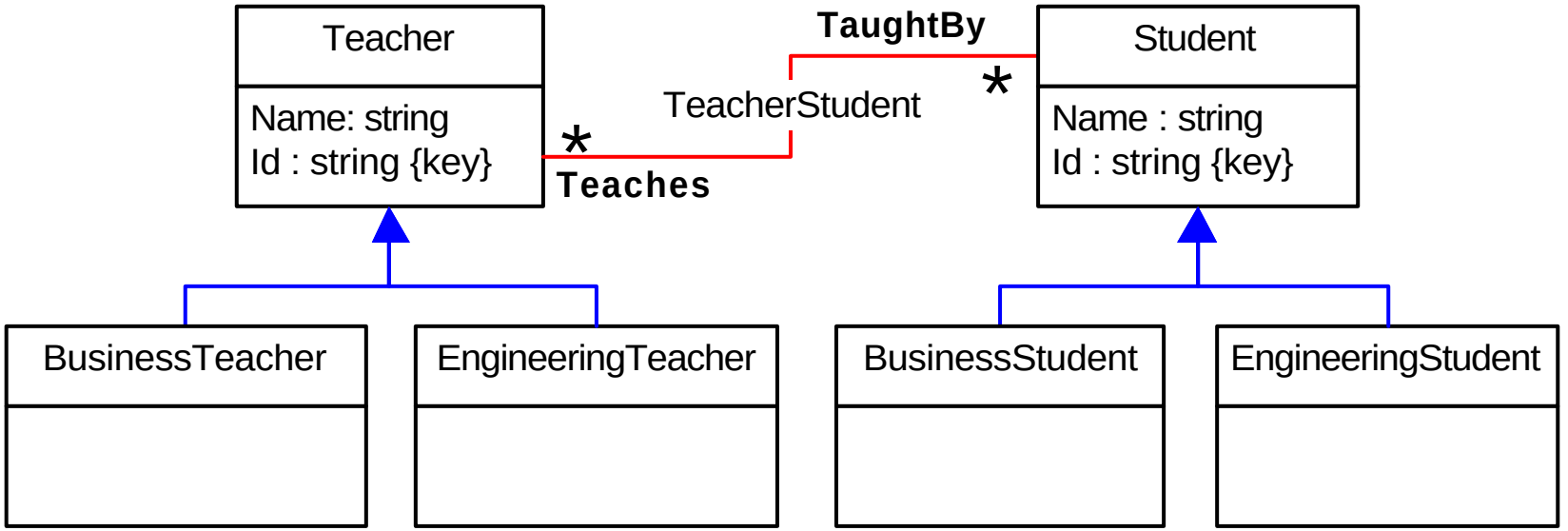
TeacherStudent
Teaches : "HTTP://CIMOM host/root:Student.Id=USC:001020200" TaughtBy : "HTTP://CIMOM host/root:Teacher.Id=USC:000010002"

2nd Instance

TeacherStudent
Teaches : "HTTP://CIMOM host/root:Student.Id=USC:001020200" TaughtBy : "HTTP://CIMOM host/root:Teacher.Id=USC:000020001"

3rd Instance

ObjectName Hiearchy



Notes: Because BusinessTeacher is subclass of Teacher, TeacherStudent relates the subclasses (ex. BusinessTeacher to EngineeringStudent

Example Instances

THE Open GROUP

BusinessTeacher	EngineeringTeacher	BusinessStudent	EngineeringStudent
Name : "John Smith" Id : "USC:000010002"	Name : "Tim Jones" Id : "USC:000020001"	Name : "Jane Doe" Id : "USC:001012111	Name : "Joe Nerd" Id : "USC:001020200

TeacherStudent	
Teaches : "HTTP://CIMOM host/root:BusinessStudent.Id=USC:001012111" TaughtBy : "HTTP://CIMOM host/root:BusinessTeacher.Id=USC:000010002"	**1st Instance**

TeacherStudent	
Teaches : "HTTP://CIMOM host/root:EngineeringStudent.Id=USC:001020200" TaughtBy : "HTTP://CIMOM host/root:BusinessTeacher.Id=USC:000010002"	**2nd Instance**

TeacherStudent	
Teaches : "HTTP://CIMOM host/root:EngineeringStudent.Id=USC:001020200" TaughtBy : "HTTP://CIMOM host/root:EngineeringTeacher.Id=USC:000020001"	**3rd Instance**

Example MOF

THE Open GROUP

```
instance of TST_BusinessStudent { name = "John Doe"; Id = “USC:001012111"; };
instance of TST_EngineeringStudent { name = “Joe Nerd"; Id = “USC:001020200"; };

instance of TST_BusinessTeacher { name = “John Smite"; Id = “USC:000010002"; };
instance of TST_EngineeringTeacher { name = “Tim Jones"; Id = “USC:000020001"; };

 Instance of TST_TeacherStudent{
 Teaches : "HTTP://CIMOM host/root:BusinessStudent.Id=USC:001012111"
 TaughtBy : "HTTP://CIMOM host/root:BusinessTeacher.Id=USC:000010002”
 };

 Instance of TST_TeacherStudent{
 …
 };

 Instance of TST_TeacherStudent{
 …
 };
```

Associations and CIM Operations

THE Open GROUP

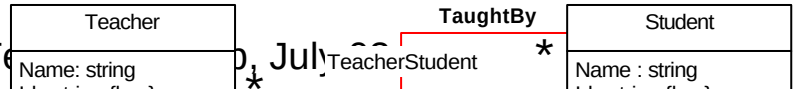
- The Association Client Operations
  - References
    - Gets the association classes or instances that refer to the specified CIM class or instance, respectively.
  - ReferenceNames
    - Gets the names of the association classes or instances that refer to the specified CIM classes or instances, respectively.
  - Associators
    - Gets the CIM classes or instances that are associated with the specified CIM class or instance.
  - AssociatorNames
    - Gets the names of the CIM classes or instances that are associated with the specified CIM class or instance.

Association Operations - References

Enumerate the association objects that refer to a particular target CIM Instance.

```
<objectPath>*ReferenceNames (
        [IN] <objectName> ObjectName,
        [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
        [IN,OPTIONAL,NULL] string Role = NULL
 )

<objectWithPath>*References (
        [IN] <objectName> ObjectName,
        [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
        [IN,OPTIONAL,NULL] string Role = NULL,
        [IN,OPTIONAL] boolean IncludeQualifiers = false,
        [IN,OPTIONAL] boolean IncludeClassOrigin = false,
        [IN,OPTIONAL,NULL] string PropertyList [] = NULL
 )
```

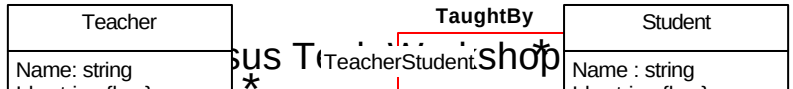


Association Operations, Associators

THE Open GROUP

Enumerate the names of CIM Instances that are associated to a particular source CIM Instance.

```
    <objectPath>*AssociatorNames (
            [IN] <objectName> ObjectName,
            [IN,OPTIONAL,NULL] <className> AssocClass = NULL,
            [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
            [IN,OPTIONAL,NULL] string Role = NULL,
            [IN,OPTIONAL,NULL] string ResultRole = NULL
    )
<objectWithPath>*References (
        [IN] <objectName> ObjectName,
        [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
        [IN,OPTIONAL,NULL] string Role = NULL,
        [IN,OPTIONAL] boolean IncludeQualifiers = false,
        [IN,OPTIONAL] boolean IncludeClassOrigin = false,
        [IN,OPTIONAL,NULL] string PropertyList [] = NULL
)
```



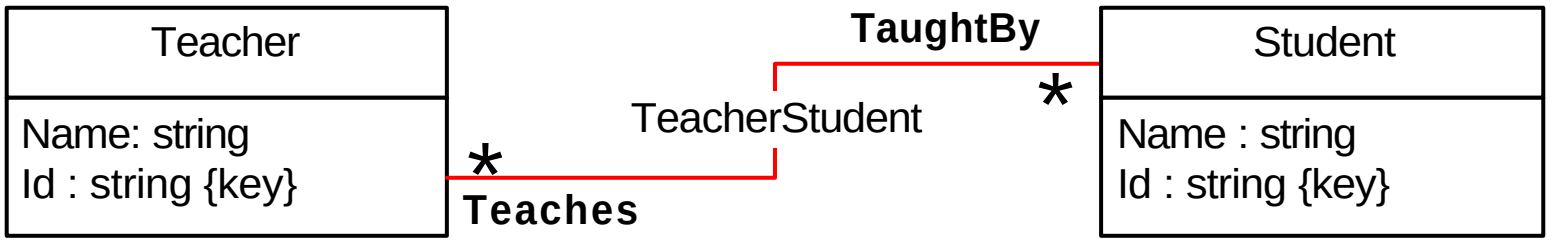
Classes and Instances

THE Open GROUP

- Associations are unique in that target (objectName) can be either class or instance.

CIMOM handles Classes, providers handle instances

Simple Operations Example - Class



REQUEST: references Teacher
RESPONSE: TeacherStudent
REQUEST: references Student
RESPONSE: TeacherStudent
REQUEST: associators Teacher
RESPONSE: Student
REQUEST: associators Student
RESPONSE: Teacher

Simple Operations Example - Instance

Request: referencenames BusinessTeacher.Id="USC:000010002"
Response : host/namespace/TeacherStudent.Id="xxx"

Complicating factors

THE Open GROUP

- Classes and Instances
- The filtering parameters:
  - Result Class/ Association Class
  - Role
  - Result Class (associators and associatorNames)

Associations in Pegasus today

THE Open GROUP

- Officially Supported in Pegasus 2.2
- Client APIs Frozen
- Provider Interfaces are experimental
  - To be frozen in Pegasus 2.3
- Pegasus supports Association Providers and static Instances
- Association providers are registered just like Instance, method providers
- Only sample providers exist today.

Special characteristics

THE Open GROUP

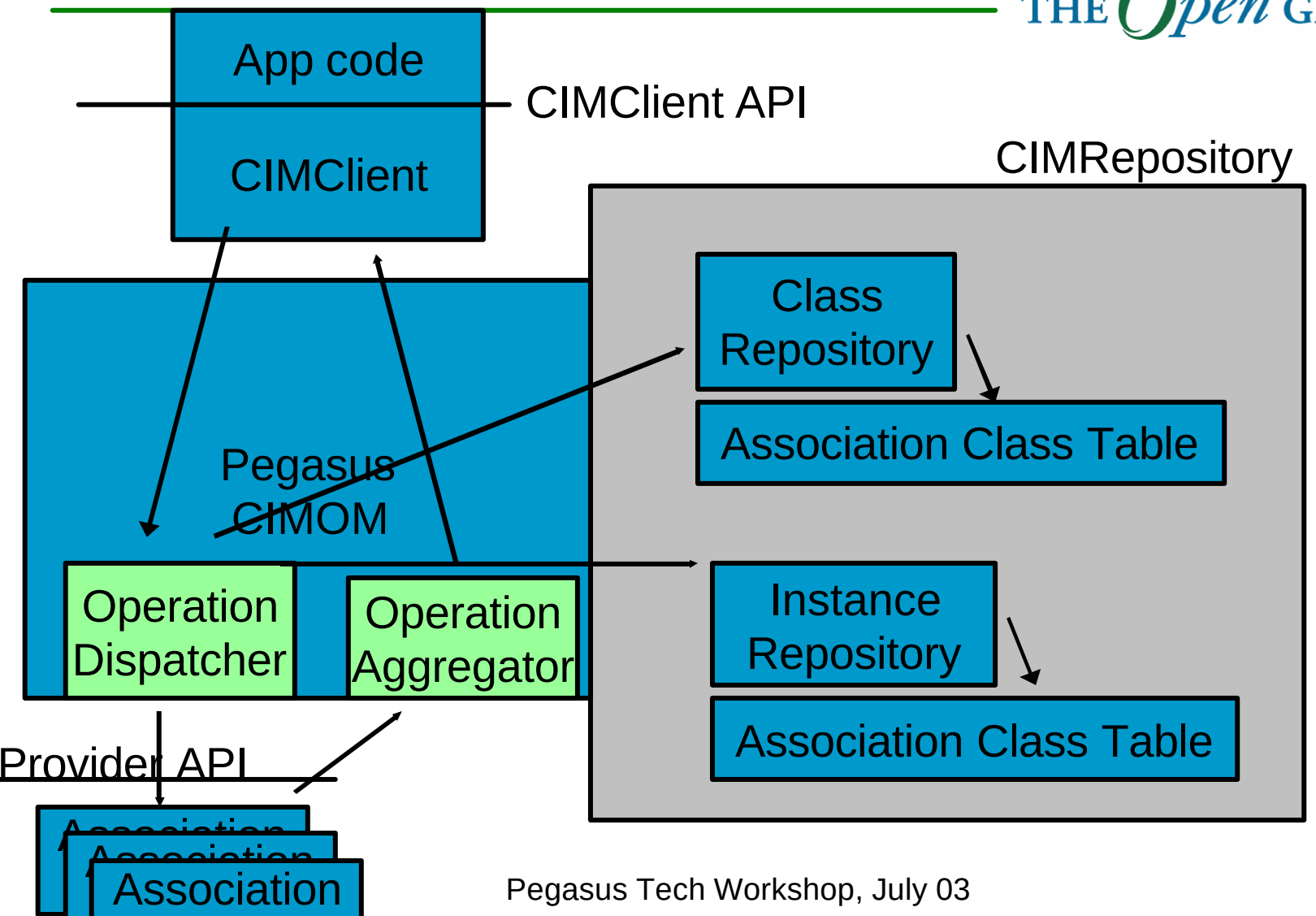
- Request Objects are either class or instance
- Requires information about instances to respond to instance level requests

Associations in the Architecture

THE Open GROUP

- CIM Client API – Association Operations
- CIM Operations Routing in Pegasus
- CIM Association Information in the repository
- CIM Provider interfaces

Association Operation in Architecture



Association Instance Operation routing

THE Open GROUP

- objectName is Class
  - CIM repository to process against the association class table
- objectName is Instance
  - Execute referenceNames operation
  - Find provider for each returned class
  - Set correct association/result class
  - Call provider for each class that has a provider.
  - If instance repository enabled, call instance repository for all classes that do not have providers.
  - Aggregate responses from providers into a single response
    - Sets host and namespace if provider did not.

CIMClient APIs

THE Open GROUP

- Frozen as of CIM 2.1
- Represent the 4 association operations
- Parallel the DMTF Operations document and other major implementations.

Comparison with the Spec.

THE Open GROUP

CIM Op SPEC

```
<objectPath>*ReferenceNames (
        [IN] <objectName> ObjectName,
        [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
        [IN,OPTIONAL,NULL] string Role = NULL
 )
```

egasus lient PI

```
Array<CIMObjectPath> referenceNames(
            const CIMNamespaceName& nameSpace,
            const CIMObjectPath& objectName,
            const CIMName& resultClass = CIMName(),
            const String& role = String::EMPTY
    );
```

CIM Client Association APIs

THE Open GROUP

```
Array<CIMObjectPath> referenceNames(
        const CIMNamespaceName& nameSpace,
        const CIMObjectPath& objectName,
        const CIMName& resultClass = CIMName(),
        const String& role = String::EMPTY
  );
```

```
Array<CIMObject> references(
        const CIMNamespaceName& nameSpace,
        const CIMObjectPath& objectName,
        const CIMName& resultClass = CIMName(),
        const String& role = String::EMPTY,
        Boolean includeQualifiers = false,
        Boolean includeClassOrigin = false,
        const CIMPropertyList& propertyList = CIMPropertyList()
  );
```

CIMClient Association APIs (cont)

THE Open GROUP

```
Array<CIMObjectPath> associatorNames(
        const CIMNamespaceName& nameSpace,
        const CIMObjectPath& objectName,
        const CIMName& assocClass = CIMName(),
        const CIMName& resultClass = CIMName(),
        const String& role = String::EMPTY,
        const String& resultRole = String::EMPTY
);
```

```
Array<CIMObject> associators(
        const CIMNamespaceName& nameSpace,
        const CIMObjectPath& objectName,
        const CIMName& assocClass = CIMName(),
        const CIMName& resultClass = CIMName(),
        const String& role = String::EMPTY,
        const String& resultRole = String::EMPTY,
        Boolean includeQualifiers = false,
        Boolean includeClassOrigin = false,
        const CIMPropertyList& propertyList = CIMPropertyList()
);
```

The Association Providers

THE Open GROUP

- Associations are one of the supported Provider types
- The interface is defined inCIMAssociationProvider
- Implemented by providers of dynamic association classes.
- The CIMOM invokes these methods when it performs association traversal.
- Same response mechanism as other provider operations
- If a Operation is on a class, the operation is solely performed by the CIMOM. If an Instance, the CIMOM invokes providers.
- Defines 4 provider operations:
  - referenceNames
  - references
  - associatorNames
  - associators

Key Difference between Clients and Providers

THE Open GROUP

- Client sees the target as objectname.
- Provider sees the target as resultClass or association class

Client Operation	Give me References for student.id=xxx
Provider	Registered for TeacherStudent. Operation is to return CIMObjectswithPath For TeacherStudent that reference student.id=x

Association Provider Registration

THE Open GROUP

- Registration the association class as the provider:
  - Ex. Register StudentTeacher, not Student or Teacher

referenceNames Operation

THE Open GROUP

- Enumerate the association objects that refer to a particular target CIM Instance. The object paths to association instances are returned. Invoked in order to perform the ReferenceNames operation

```
virtual void referenceNames(
        const OperationContext & context,
        const CIMObjectPath & objectName,
        const CIMName & resultClass,
        const String & role,
        ObjectPathResponseHandler & handler) = 0;
```

referenceNames (cont)

THE Open GROUP

- Parameters
  - objectName - CIMObjectPath defining the source CIM Instance whose associated Instances are to be returned. This argument MUST contain the modelpath of an Instance. (i.e. Keys populated)
  - resultName - CIMObjectPath defining the Association in which objectName MUST participate. The Provider uses this information to identify which Association must be traversed in the case that it supports more than one Association.
  - role - This string MUST either contain a valid Property name or be null. It filters the Instances returned to contain only Association Instances that refer to objectName in which objectName plays the specified role. (i.e. the Property name in the Association Instance that refers to objectName matches this value) If "Antecedent" is specified, then only Association Instances in which objectName is the "Antecedent" reference are returned.

references Operation

THE Open GROUP

- Enumerate the association objects that refer to a particular target CIM Instance. Entire association instances are returned. This method is invoked in order to perform the References operation

```
virtual void references(
        const OperationContext & context,
        const CIMObjectPath & objectName,
        const CIMName & resultClass,
        const String & role,
        const Boolean includeQualifiers,
        const Boolean includeClassOrigin,
        const CIMPropertyList & propertyList,
```

References (cont)

THE Open GROUP

- Parameters
  - objectName - CIMObjectPath defining the source CIM Instance whose associated Instances are to be returned. This argument MUST contain the modelpath of an Instance. (i.e. Keys populated) a
  - resultName - CIMObjectPath defining the Association in which objectName MUST participate. The Provider uses this information to identify which Association must be traversed in the case that it supports more than one Association.
  - role - This string MUST either contain a valid Property name or be null. It filters the Instances returned to contain only Association Instances that refer to objectName in which objectName plays the specified role. (i.e. the Property name in the Association Instance that refers to objectName matches this value) If "Antecedent" is specified, then only Association Instances in which objectName is the "Antecedent" reference are returned.

References (cont)

THE Open GROUP

- Parameters (cont)
  - IncludeQualifiers - If true, all Qualifiers for each Instance (including Qualifiers on the Object and on any returned Properties) are be included in the Instances returned. If false, no Qualifiers are present in each Instance returned.
  - IncludeClassOrigin - If true, the CLASSORIGIN attribute will be present on all appropriate elements in the Instances returned. If false, no CLASSORIGIN attributes are present in the Instances returned. CLASSORIGIN is attached to an element (properties, methods, references) to indicate the class in which it was first defined.
  - propertyList - An array of property names used to filter what is contained in the Instances returned. Each CIMInstance returned only contains elements for the properties of the names specified. Duplicate and invalid property names are ignored and the request is otherwise processed normally. An empty array indicates that no properties should be included in the Instances returned. A null value indicates that all properties should be contained in the Instances returned. NOTE: Properties should not be specified in this parameter unless a non-null value is specified in the resultClass parameter.

Associator Names

THE Open GROUP

- Enumerate the names of CIM Instances that are associated to a particular source CIM Instance. The object paths to the instances associated to the specified instance are returned. Invoked in order to perform the AssociatorNames operation.

```
virtual void associatorNames(
        const OperationContext & context,
        const CIMObjectPath & objectName,
        const CIMName & associationClass,
        const CIMName & resultClass,
        const String & role,
        const String & resultRole,
        ObjectPathResponseHandler & handler) = 0;
```

AssociatorNames (Cont)

THE Open GROUP

- objectName - CIMObjectPath defining the source CIM Instance whose associated Instances are to be returned. This argument MUST contain the modelpath of an Instance. (i.e. Keys populated)
- assocName - CIMObjectPath defining the Association in which objectName MUST participate. The Provider uses this information to identify which Association must be traversed in the case that it supports more than one Association.
- resultClass - This string MUST either contain a valid CIM Class name or be null. It filters the Instances returned to contain only the Instances of this Class name or one of its subclasses.
- role - This string MUST either contain a valid Property name or be null. It filters the Instances returned to contain only Instances associated to objectName via an Association in which the objectName plays the specified role. (i.e. the Property name in the Association class that refers to objectName matches this value) If "Antecedent" is specified, then only Associations in which objectName is the "Antecedent" reference are examined.

AssociatorNames (Cont)

THE Open GROUP

- resultRole - This string MUST either contain a valid Property name or be null. It filters the Instances returned to contain only Instances associated to objectName via an Association in which the Instance name returned plays the specified role. (i.e. the Property name in the Association class that refers to the Instance name returned matches this value) If "Dependent" is specified, then only Associations in which the Instance name returned is the "Dependent" reference are examined.

- RETURNS
  - If successful, an array containing CIMObjectPaths to the Instances meeting the specified criteria is returned. If no such Instances are found, null is returned.

Associator Operation

THE Open GROUP

- Enumerate CIM Instances that are associated to a particular source CIM Instance. The entire instances associated to the specified instance are returned.

Associator Operation (cont)

THE Open GROUP

```
virtual void associators(
        const OperationContext & context,
        const CIMObjectPath & objectName,
        const CIMName & associationClass,
        const CIMName & resultClass,
        const String & role,
        const String & resultRole,
        const Boolean includeQualifiers,
        const Boolean includeClassOrigin,
        const CIMPropertyList & propertyList,
        ObjectResponseHandler & handler) = 0;
```

Associator Operation(cont)

THE Open GROUP

- Parameters
  - objectName - CIMObjectPath defining the source CIM Instance whose associated Instances are to be returned. This argument MUST contain the modelpath of an Instance. (i.e. Keys populated)
  - assocName - CIMObjectPath defining the Association in which objectName MUST participate. The Provider uses this information to identify which Association must be traversed in the case that it supports more than one Association. resultClass - This string MUST either contain a valid CIM Class name or be null. It filters the Instances returned to contain only the Instances of this Class name or one of its subclasses.
  - role - This string MUST either contain a valid Property name or be null. It filters the Instances returned to contain only Instances associated to objectName via an Association in which the objectName plays the specified role. (i.e. the Property name in the Association class that refers to objectName matches this value) If "Antecedent" is specified, then only Associations in which objectName is the "Antecedent" reference are examined.

Associator Operation (cont)

THE Open GROUP

- resultRole - This string MUST either contain a valid Property name or be null. It filters the Instances returned to contain only Instances associated to objectName via an Association in which the Instance returned plays the specified role. (i.e. the Property name in the Association class that refers to the Instance returned matches this value) If "Dependent" is specified, then only Associations in which the Instance returned is the "Dependent" reference are examined.
- IncludeQualifiers - If true, all Qualifiers for each Instance (including Qualifiers on the Object and on any returned Properties) are be included in the Instances returned. If false, no Qualifiers are present in each Instance returned.
- includeClassOrigin - If true, the CLASSORIGIN attribute will be present on all appropriate elements in the Instances returned. If false, no CLASSORIGIN attributes are present in the Instances returned. CLASSORIGIN is attached to an element (properties, methods, references) to indicate the class in which it was first defined.

Associator Operation (cont)

THE Open GROUP

- PropertyList - An array of property names used to filter what is contained in the Instances returned. Each CIMInstance returned only contains elements for the properties of the names specified. Duplicate and invalid property names are ignored and the request is otherwise processed normally. An empty array indicates that no properties should be included in the Instances returned. A null value indicates that all properties should be contained in the Instances returned. NOTE: Properties should not be specified in this parameter unless a non-null value is specified in the resultClass parameter.

Associator Operation (cont)

THE Open GROUP

Returns

- If successful, an array containing CIMInstances meeting the specified criteria is returned. If no such Instances are found, null is returned.

Writing a Provider

THE Open GROUP

- General Rules
- Must define all 4 operations
  - Minimum implementation is “Not Supported”
- Not normally logical to “Not Support” a subset of the association operations.
  - References and not associations is not logical
- Uses handler to deliver responses
- Should define an instance provider for the Association class. (Enumerate and get instance). The modify and create are provider dependent.

Very simple code for provider

THE Open GROUP

TeacherStudentProvider, referenceNames

```
....
void StudentTeacherProvider::referenceNames(
        const OperationContext & context,
        const CIMObjectPath & objectName,
        const CIMName & resultClass,
        const String & role,
        ObjectPathResponseHandler & handler)
{
        throw CIMNotSupportedException(
                "StudentTeacherProvider::referenceNames");
}
```

Example CIM Operation – Client Request

THE Open GROUP

CLI

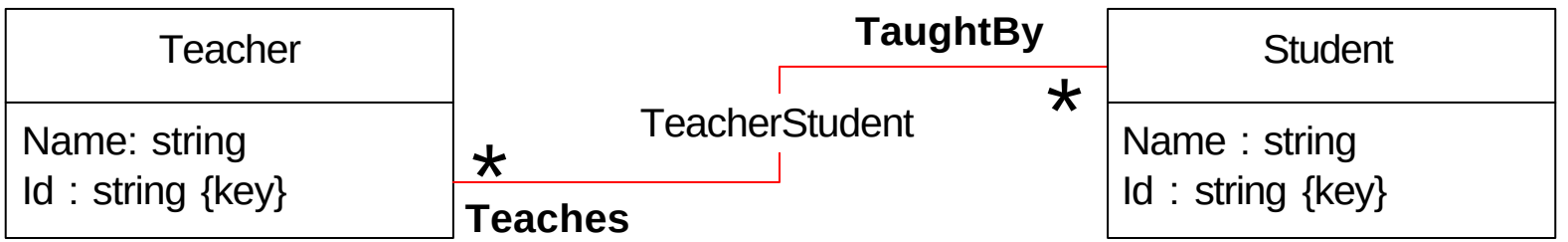
```
Referencenames objectName = Student.Id="xxx"
```

CIMClient

```
Array<CIMObjectPath cimObjectPaths;
CIMObject TBD
CIMObjectName = "TBD";
cimObjectPaths = referenceNames(
                nameSpace,
                objectName );
```

What we want is CIMObjectPath of all instances of TeacherStudent Association Class that refer to class Student, Id=xxx.

NOTE: resultClass = NULL and role = NULL.

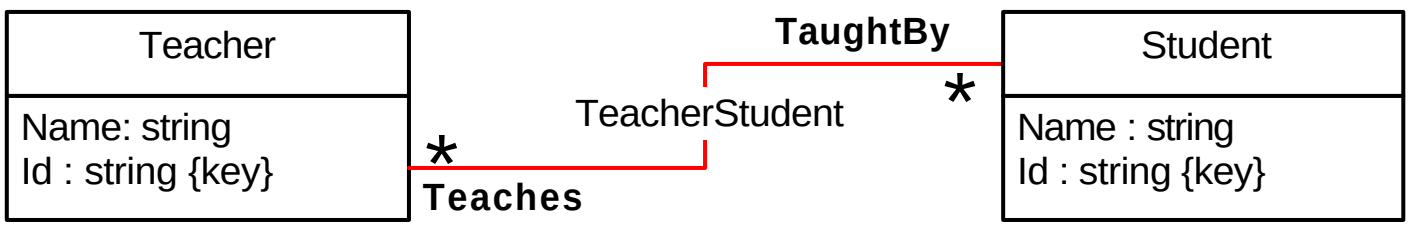


Example, Request to Provider

THE Open GROUP

```
virtual void referenceNames(
        const OperationContext & context,
        const CIMObjectPath & objectName,
        const CIMName & resultClass,
        const String & role,
        ObjectPathResponseHandler & handler) = 0;
```

resultClass = “StudentTeacher”
objectName = Student.Id=xxxx
role = String::EMPTY (translate from NULL CIM Operation parameter)



Provider Functions, general

THE Open GROUP

- Get Instances of the association Class (TeacherStudent)
- Filter to find all instances that have
  - Any referencevalue = Student.Id="xxxx"
  - Or if role exists referenceName = role and referencevalue = Student.Id="xxxx"
  - i.e. Instances of Student with Id=xxx
- If referencenames operation
  - Return array of CIMObjectPath of the association Class that pass the filter.
- If references operation
  - Return array of instances of TeacherStudent as CIMObjectwithPath that pass the filter tests (honoring includeQualifer and propertyList)
- MUST return full CIMObjectPath (with namespace and host name)

What makes Associatiors Different?

THE Open GROUP

- Associators returns the referenced Instance Name or Instance, not the associator instance.
- Associators consist of the process for referenceNames +
  - Having found the associatior instance, get all other references
  - Filter with resultrole parameter

Issues and work

THE Open GROUP

- Installed with sample provider (providers/Sample/FamilyProvider) 2.2
- Extending to more extensive tests in providers/testprovider/associationtest (for 2.3)
- Some optimization in process for repository functions.
- Considering two small changes to Provider interfaces before freezing provider interface (2.3).