

MatrixOne

Business Modeler

Chennai

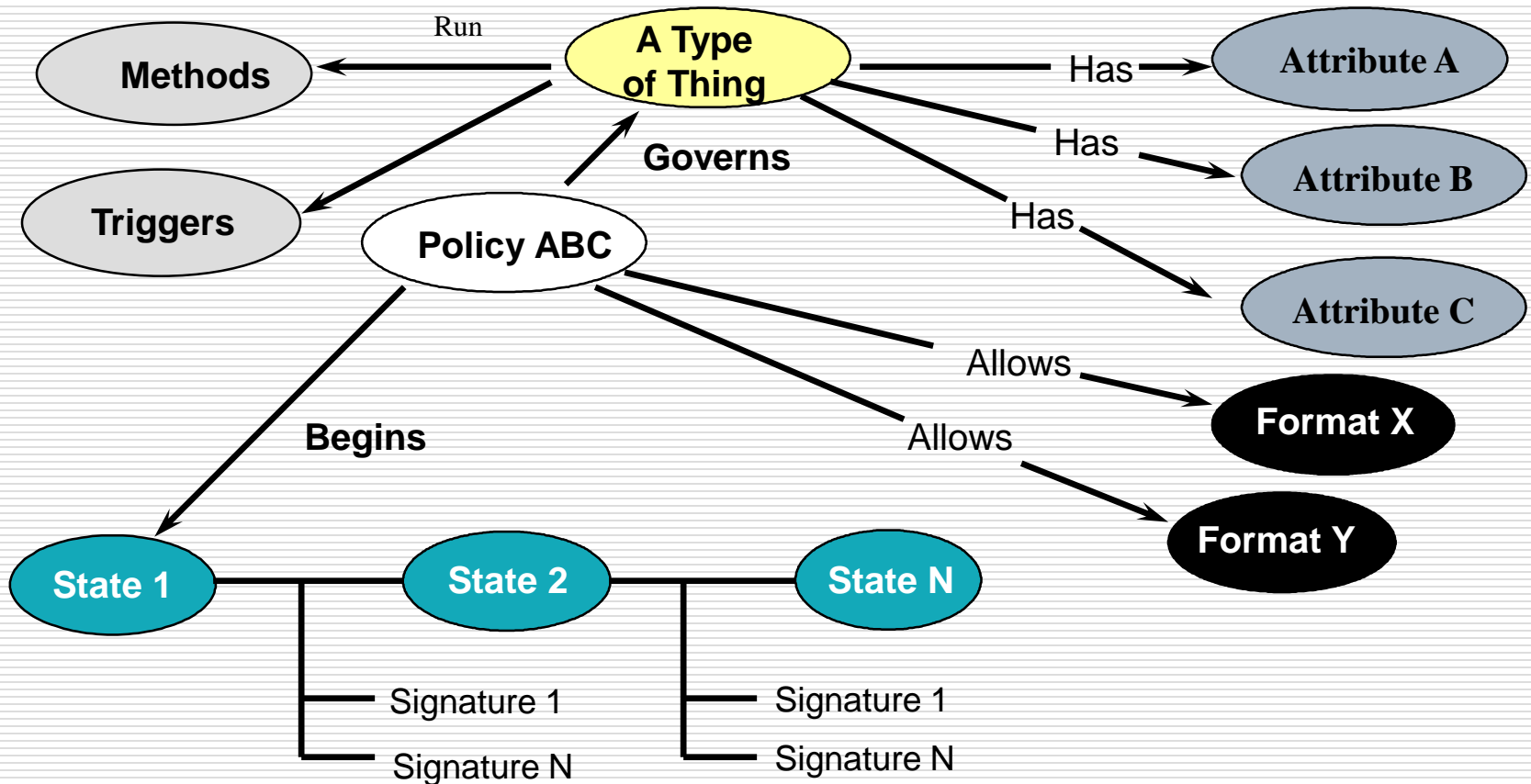
Agenda

Business Modeling Interface

Matrix Schema, Its Object and Relationship

Design and Implement a Matrix Schema

Matrix Schema

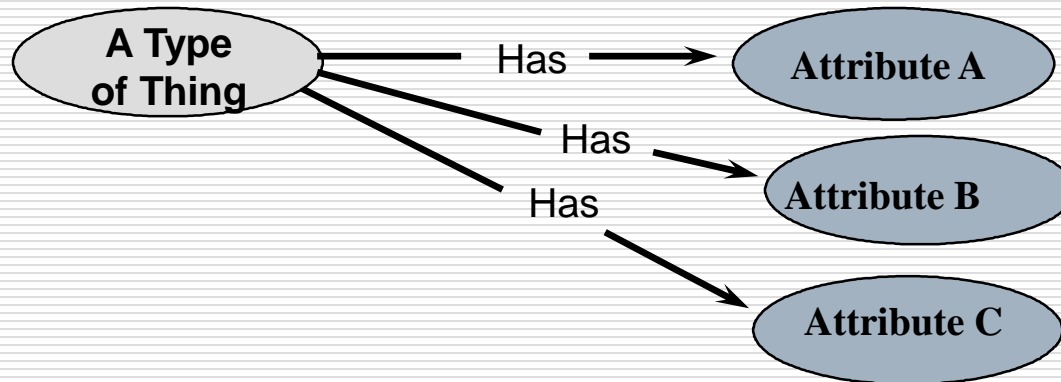


The Business Object

A Type of Thing

- The first thing needed in a schema is the business object
- Business object is anything the customer wants to manage
- Users own business objects
- Objects may be passed to someone else
- Business objects are also something that the user wants to manage

Attributes



- An attribute is a discrete characteristic of the business object
Eg: Color, age, gender etc
- When an attribute is assigned, the user will be prompted to enter an attribute value when the object is created

Creating Attribute

The screenshot shows the 'New Attribute' dialog box with the following fields and controls:

- Basics** (selected tab)
- Name**: Text input field
- Description**: Text input field
- Type**: Dropdown menu
- Default**: Text input field
- ☐ Hidden
- ☐ Multiline
- Icon**: List of icons and a selection button
- Create** button
- Cancel** button

Basics : Specify the Name, Description Type of the Attribute.

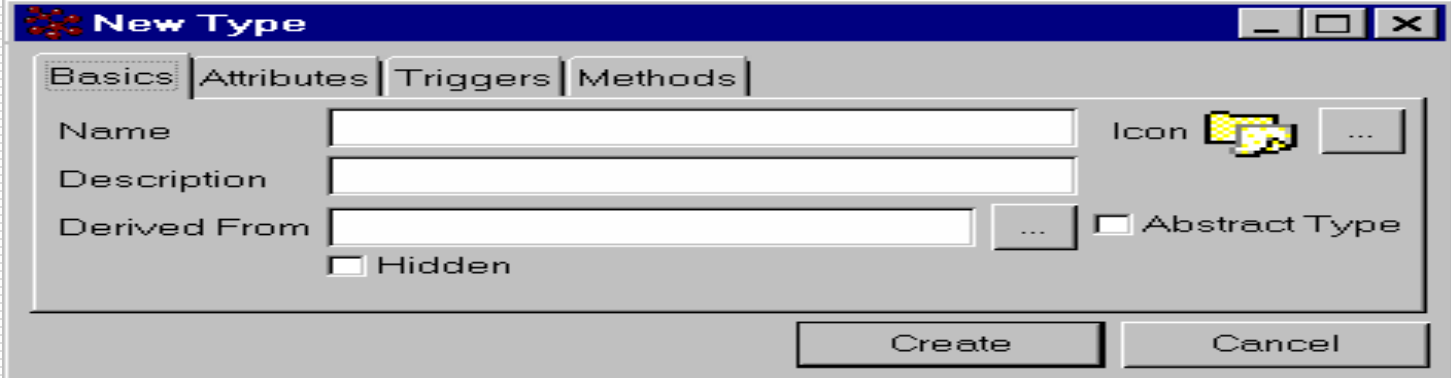
Ranges : Specify the attribute range using operators.

Triggers : To handle various actions on attributes include appropriate triggers.

Type

- A type defines a kind of business object and the collection of attributes it can have
- Types can inherit properties from other types
- Abstract types are not used to create any actual instances of the type
- Abstract types are useful only in defining characteristics that are inherited by other object types, and in categorizing classes of similar objects such as documents, components, drawings etc

Creating Type



The screenshot shows the 'New Type' dialog box with the following fields and options:

- Basics** (selected tab):
 - Name: [Text Input]
 - Description: [Text Input]
 - Derived From: [Text Input] ...
 - ☐ Hidden
 - Icon: [Icon] ...
 - ☐ Abstract Type
- Attributes**: [Tab]
- Triggers**: [Tab]
- Methods**: [Tab]
- Buttons: Create, Cancel

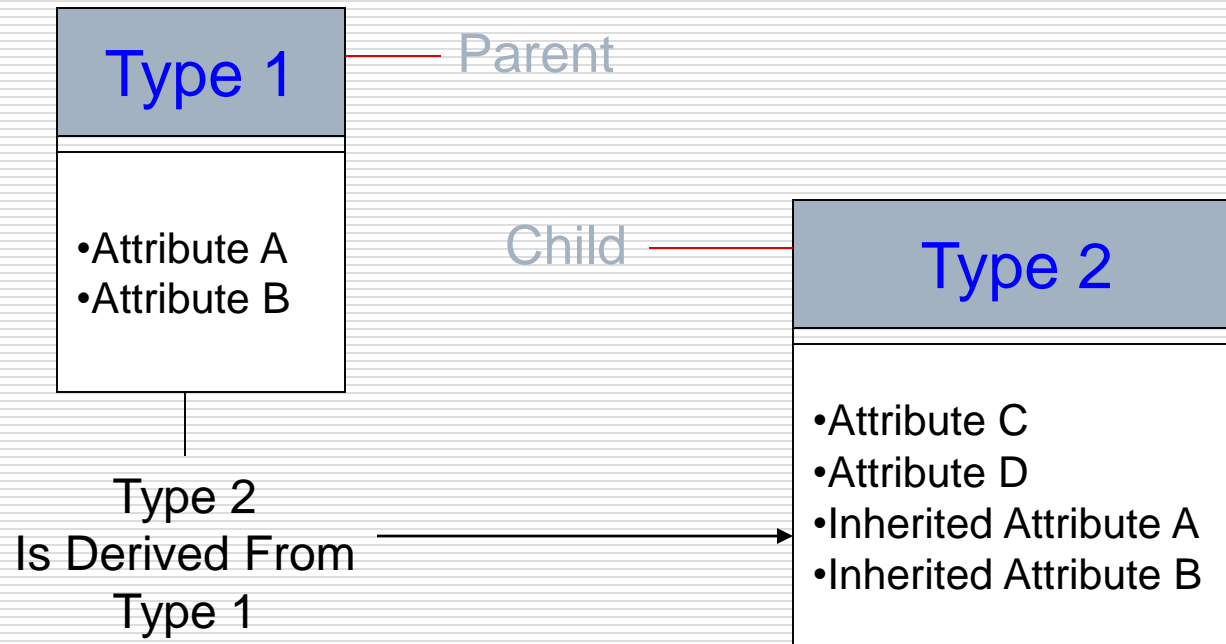
Basics : Specify the Name, Description and Derived from which Type.

Attributes : Specify the attribute range using operators.

Triggers : To handle various actions on Type include appropriate triggers.

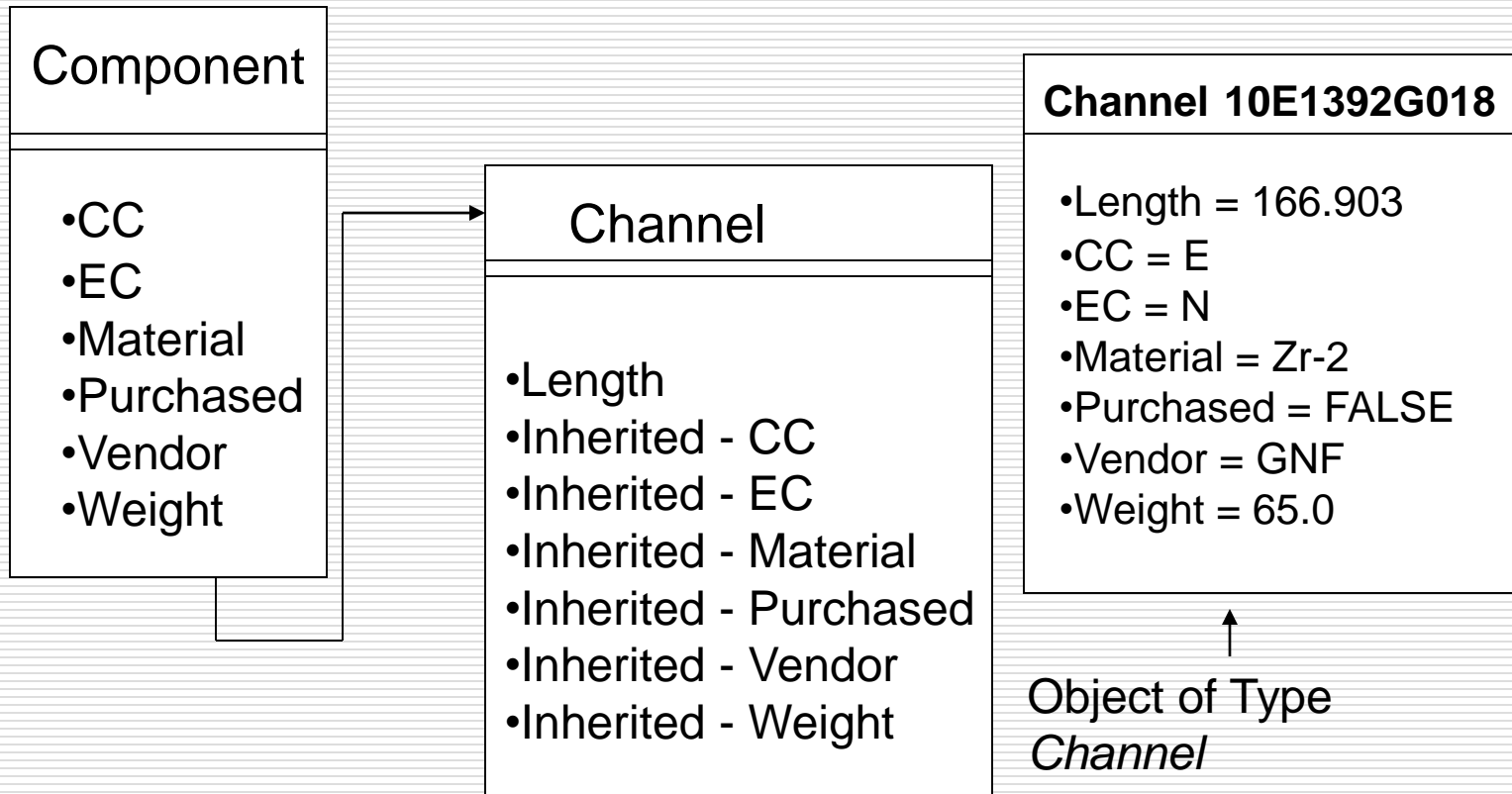
Methods : can add programs.

Derived Types



- Derived types are objects that are derived from another object type
- A derived type inherits the attributes of the type from which it was derived

Derived Type Example:



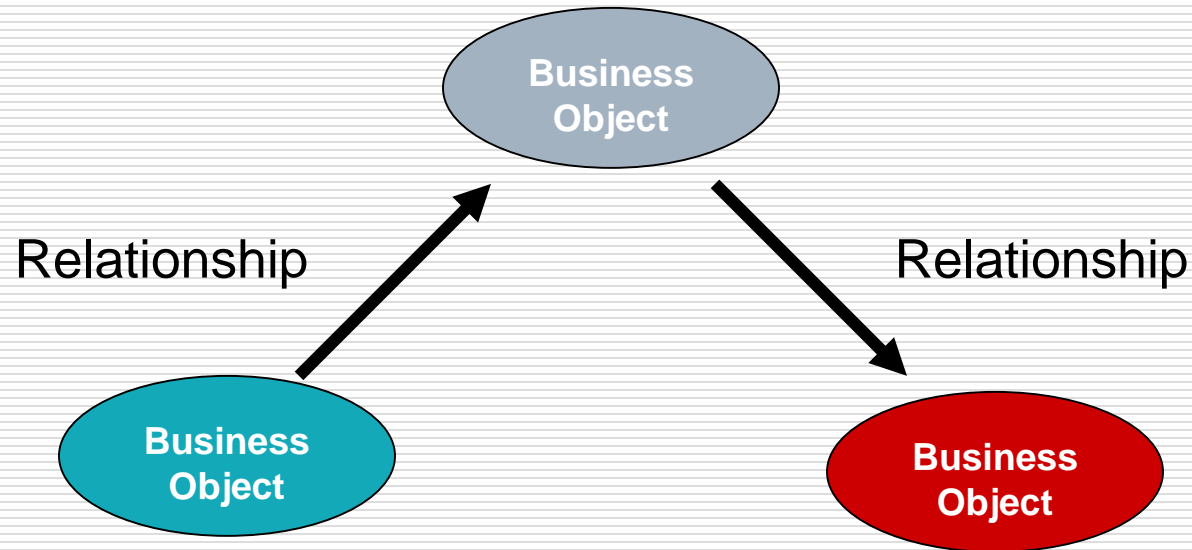
Agenda

Business Modeling Interface

Matrix Schema, Its Object and Relationship

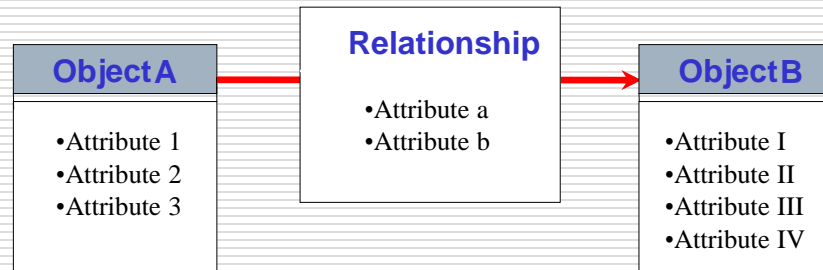
Design and Implement a Matrix Schema

Relationships



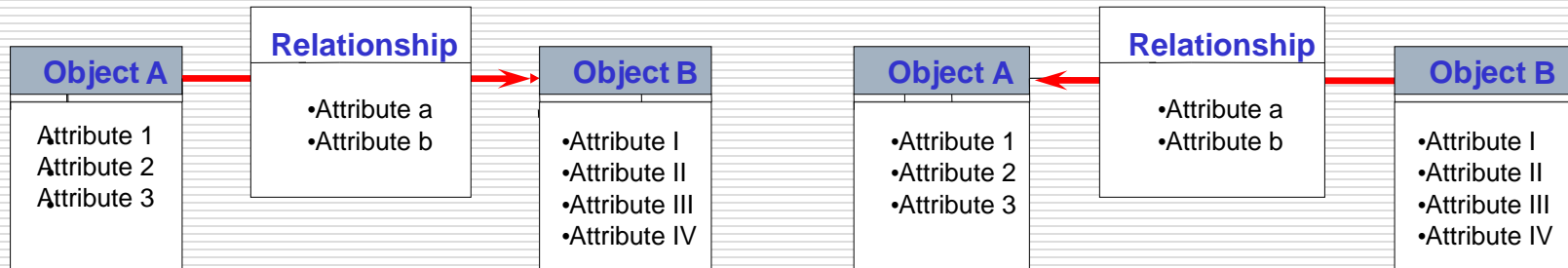
- Relationships connect or link business objects

Relationship Example



- FROM SIDE TO SIDE
- The relationships are directional, they have a starting point - the “FROM” side and an ending point - the “TO” side

Relationship Types



- Equitable Relationships - Relationships are equal to each other
- For e.g Object A will be From side for certain relationship
The same Object will be To side for some other relationship.

FROM SIDE
FROM SIDE
FROM SIDE

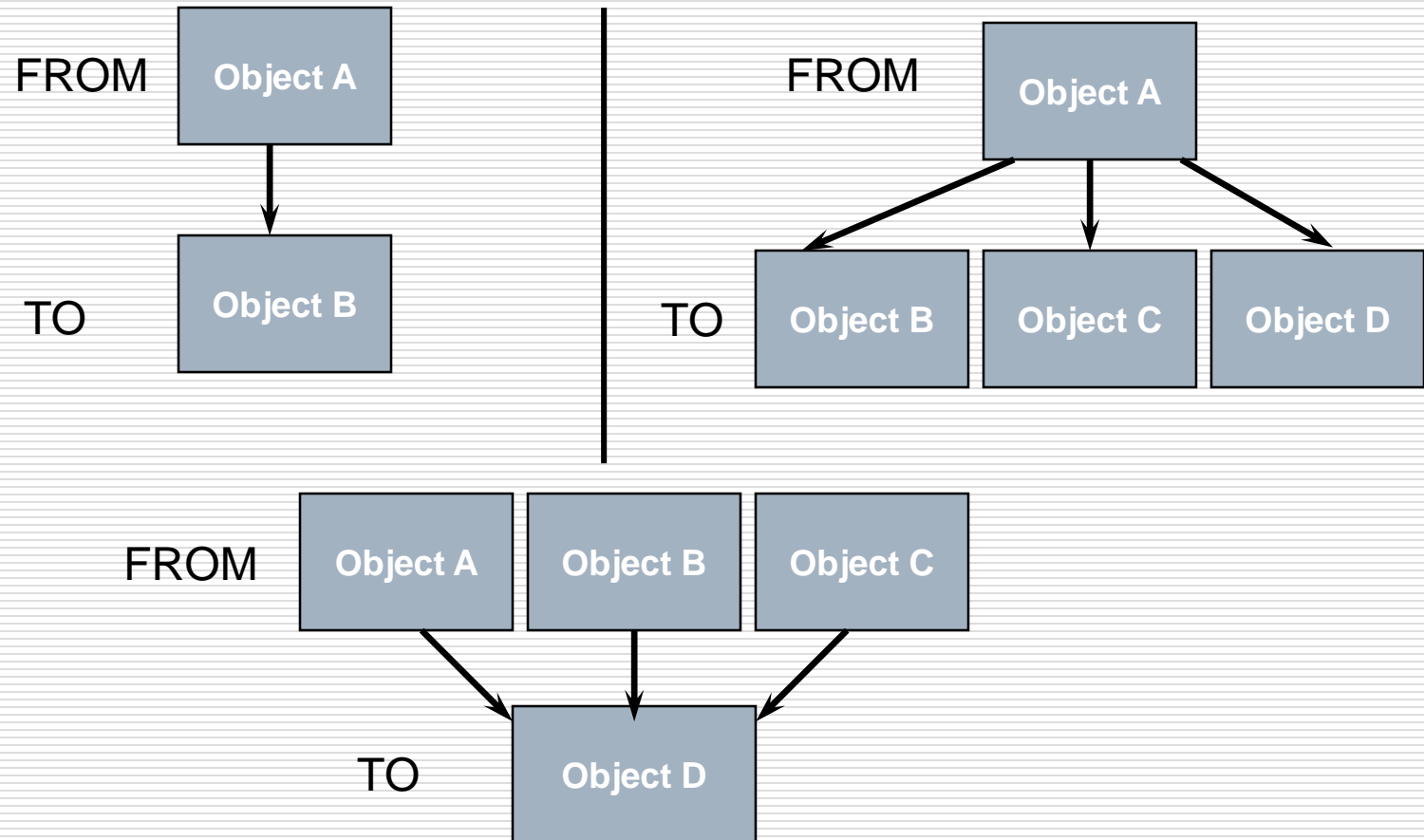
TO SIDE
TO SIDE
TO SIDE

TO SIDE
TO SIDE
TO SIDE

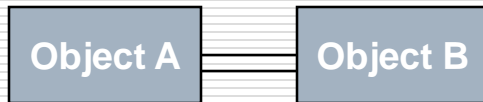
TO SIDE
TO SIDE
TO SIDE

FROM SIDE
FROM SIDE
FROM SIDE

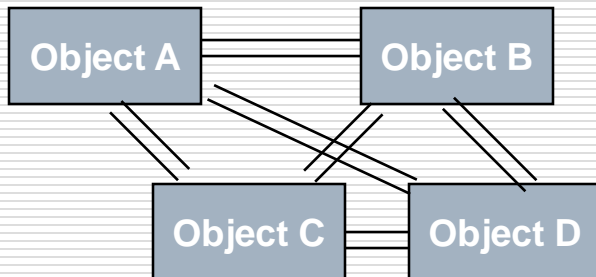
Hierarchical Relationship



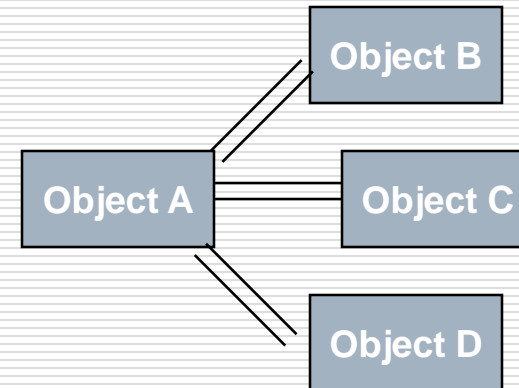
Relationship Cardinality



One-to-One Relationships



Many-to-Many Relationships



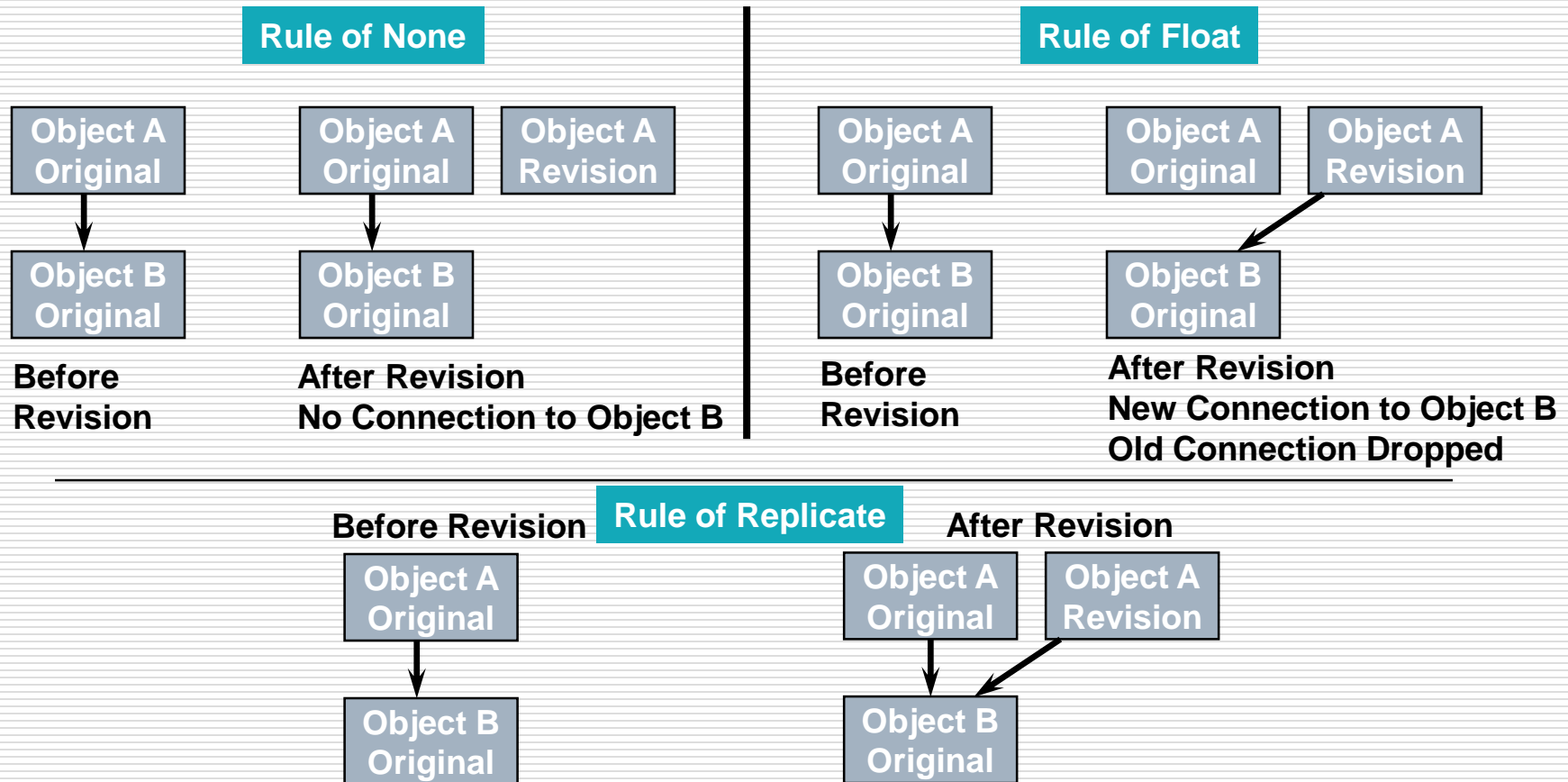
One-to-Many Relationships

- Cardinality refers to the number of connections of this type that business objects can have
- Cardinality can be either **One** or **Many**

Relationship Revisions

- The Revision Rule specifies how revisions to the connected objects are handled
- There are three types of revisions:
 - None
 - Float
 - Replicate

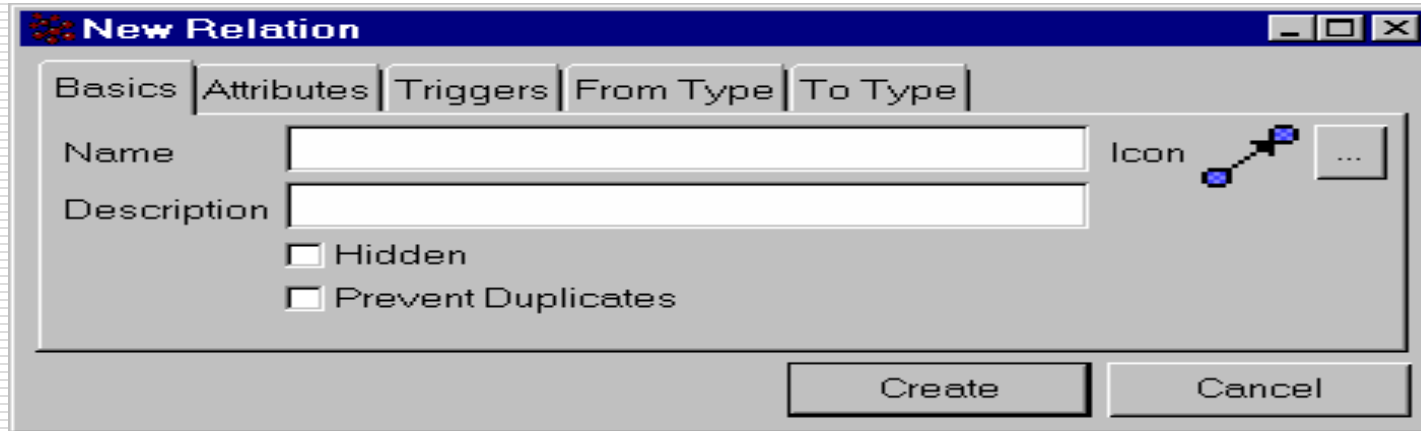
Relationship Revisions



Relationships and Cloning

- Cloning a Business Object allows to create a new Business Object of the same Type but with a different Name
- When defining a Relationship, one has to decide what action has to be taken when cloning a Business Object that is already connected to one or more Business Objects

Creating Relationship



New Relation


Basics | Attributes | Triggers | From Type | To Type

Name:

Description:

☐ Hidden

☐ Prevent Duplicates

Icon: 

Basics : Specify the Name, Description of the Relationship

Attributes : Specify the attributes using relationships.

Triggers : To handle various actions on Type include appropriate triggers.

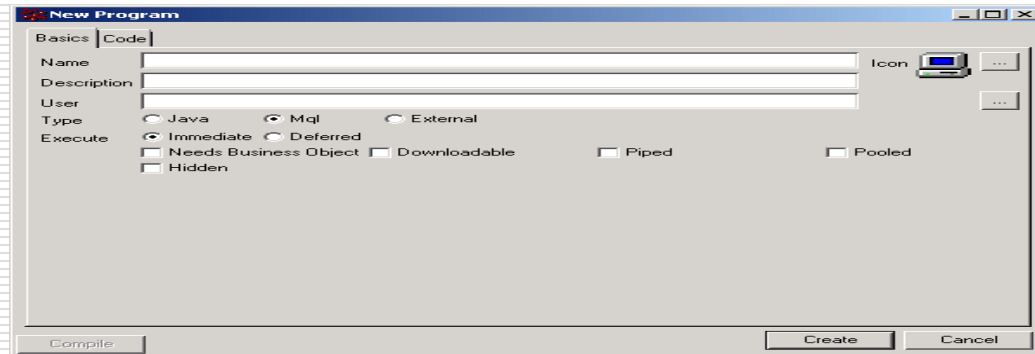
From Type : add Type as From connection.

To Type : add Type as To connection.

Programs

- A Program is a collection of software code
(usually MQL, TCL/TK or an external application - e.g Word, AutoCAD etc)
- Programs can be used for automating Matrix functions and/or viewing/editing/printing files
- Programs are used in Types, Formats and Policies

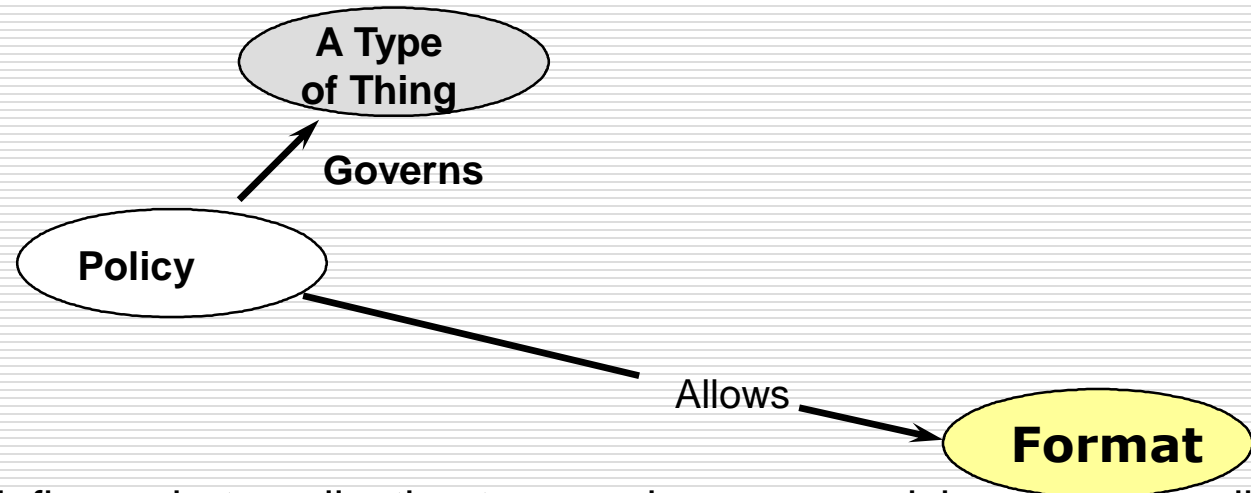
Creating Program



Basics : Specify the Name, Description ,User and which type of program

Code : Enter a program to execute while calling this program.

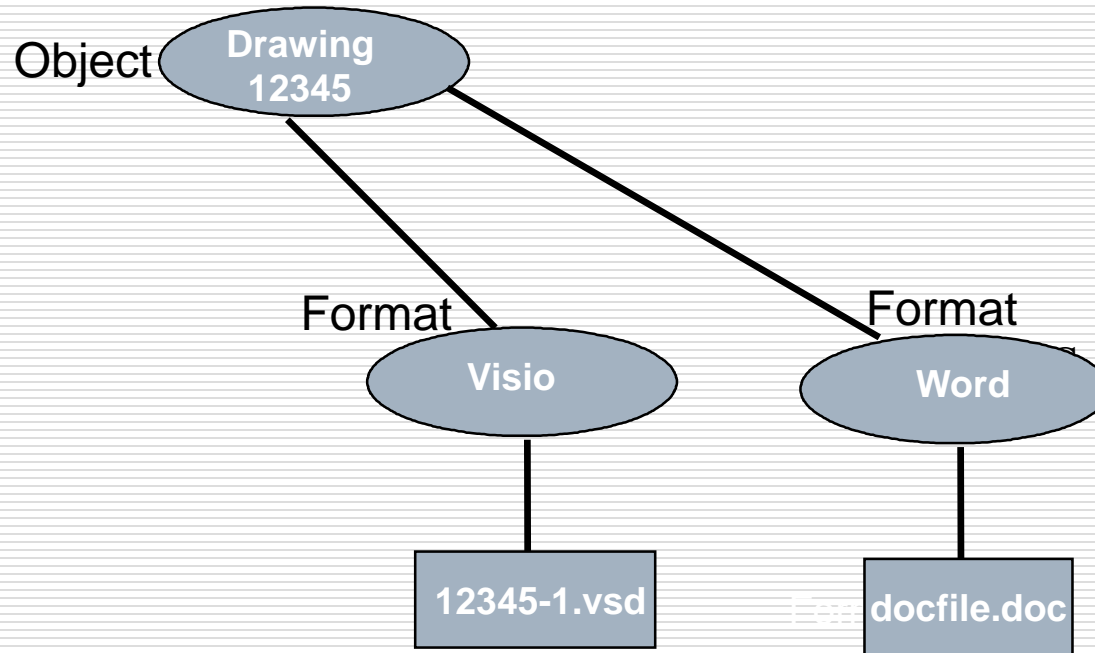
Format



- A Format defines what application to use when a user wishes to view, edit or print a file that is checked in to a Business Object
- Formats are controlled by Policies
- Files are associated with a Format by “Check-In”
- Files are “taken out” of a Format by “Check-Out”

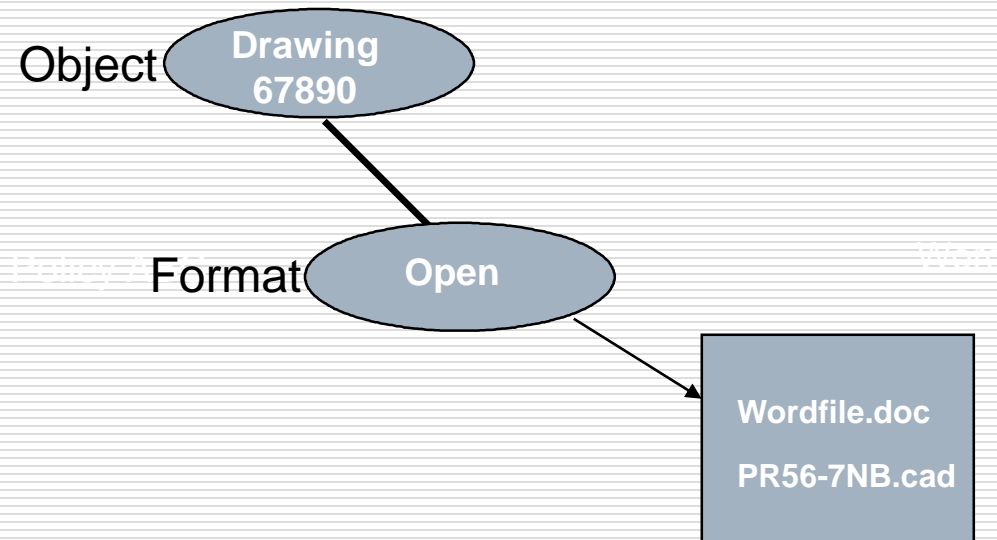
Format Y

Format Example



- Drawing 12345 has Visio and Word file Formats
- Formats (ex. vsd, doc, xls, gif etc.) are defined prior to their use
- Formats that are allowed for the Business Objects must be specified when a Policy is defined along with the "default" Format for that Policy

Open Format



- Drawing 67890 has an “Open” file Format
- It is not necessary to specify a particular Program to execute on View, Edit or Print when defining an Open Format
- The suffix of the checked-in file determines through the Windows System Registry which application to invoke

Part-11502.X15

Creating Format

New Format

Name Icon

Version

Description

File Suffix

File Creator

File Type

View Command

Edit Command

Print Command

☐ Hidden

Create Cancel

Name : Specify the name of Format

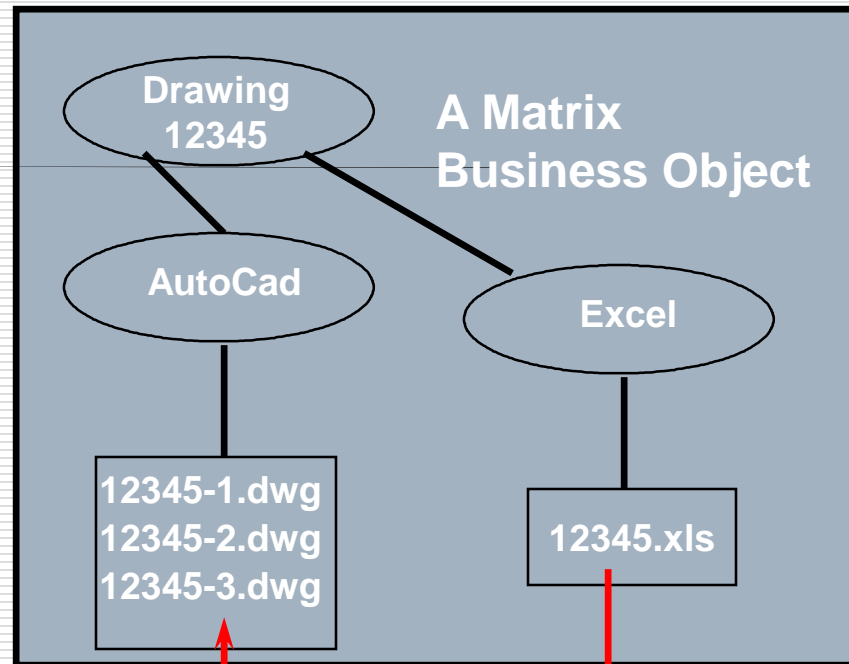
Version: Specify version of the doc.

File Suffix: Specify suffix of the file in this row.

File Type: Specify type of the file(.doc or .xls like this)

File/Directory Check Out

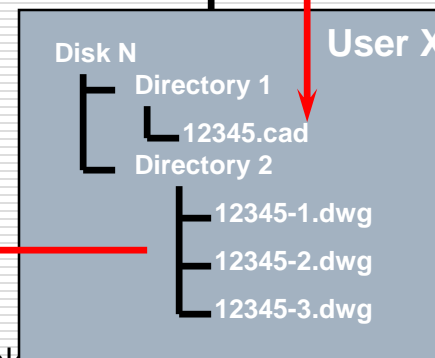
Shared Space



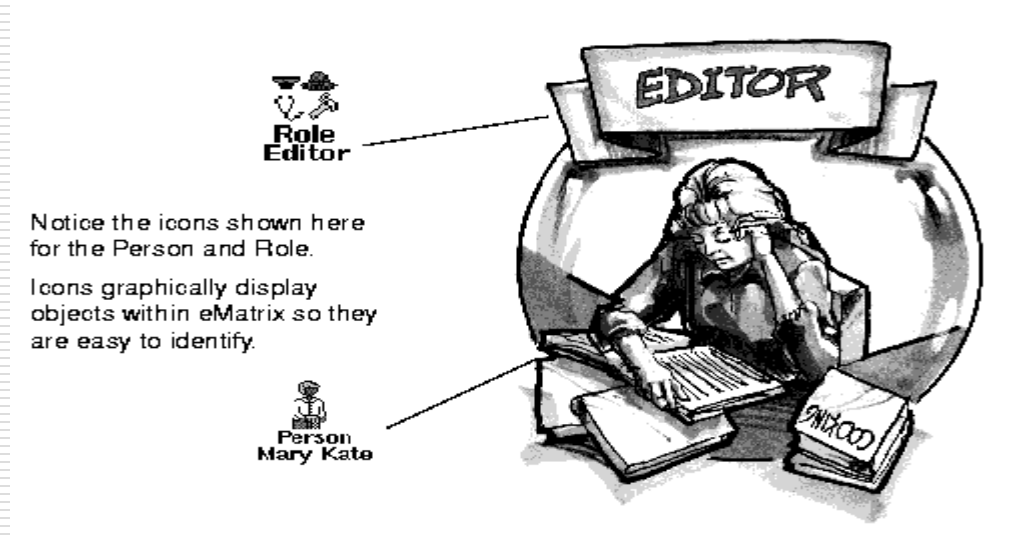
Network

Check Out

Check in

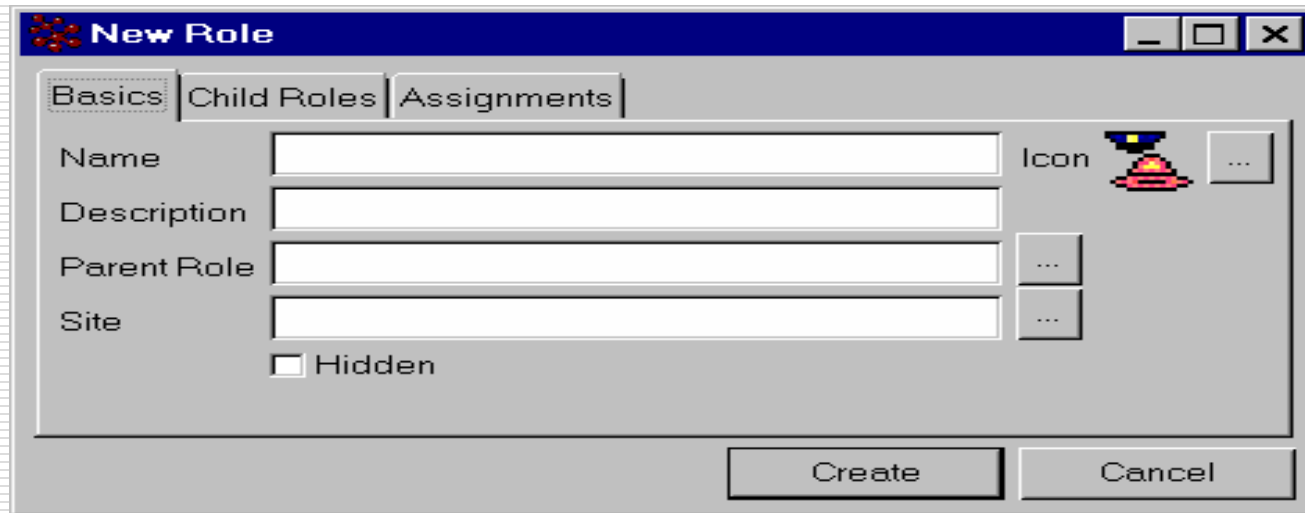


Person and Role




- Person definition identifies the role played by an individual in an organization
- A person can have more than one role within Matrix
- A role an individual has within Matrix decides the business objects he can own and access

Creating Role



The screenshot shows a 'New Role' dialog box with the following fields and controls:

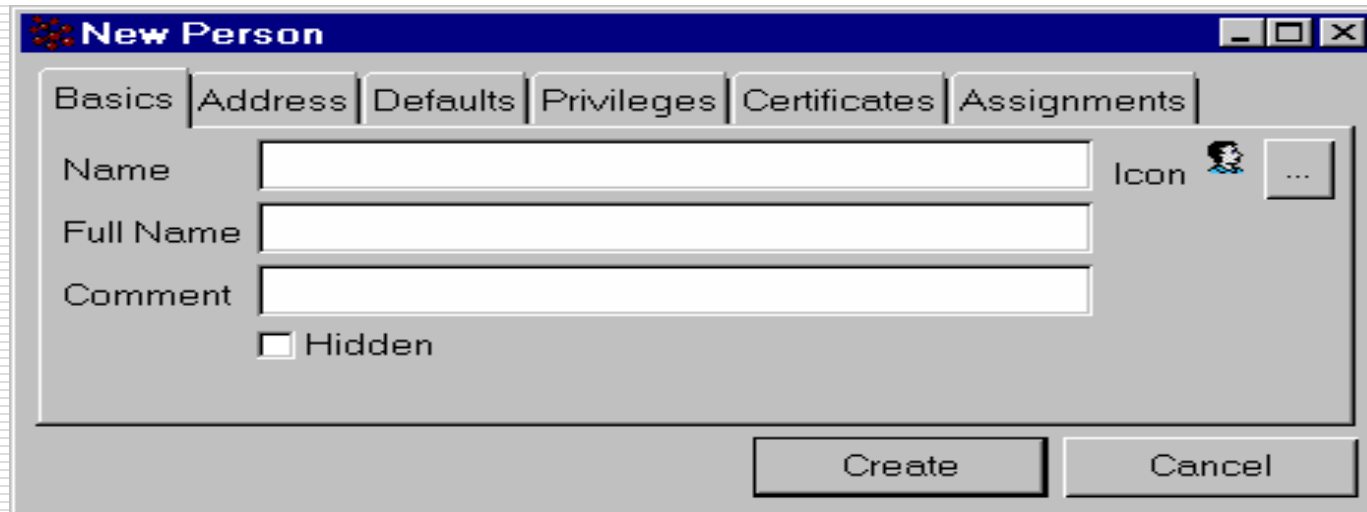
- Basics** (selected tab):
 - Name: Text input field
 - Description: Text input field
 - Parent Role: Text input field
 - Site: Text input field
 - Hidden: ☐ checkbox
 - Icon:  and a button with three dots
- Child Roles**: Tab label
- Assignments**: Tab label
- Create**: Button
- Cancel**: Button

Basics : Specify the name ,description and Parent Role of Role

Child Roles: Specify child Roles.


Assignments: Specify some Assignments for this Role

Creating Person



New Person

Basics | Address | Defaults | Privileges | Certificates | Assignments

Name Icon  ...

Full Name

Comment

☐ Hidden

Create Cancel

Basics : Specify the name ,Full Name and Comment of a Person

Address: Specify the address.

Default: Specify Site and Vault.

Privileges : Specify the access privilege of a Person

Certificates : Add files

Assignment : Add Group and Role assignment to a Person.

Person - Privileges

The screenshot shows a Windows-style dialog box titled "Edit - Person no priv". It has a tabbed interface with the following tabs: Basics, Address, Defaults, Privileges (selected), Certificates, and Assignments. The "Privileges" tab contains the following fields and controls:

- Access:** Two user icons with ellipsis buttons, followed by the text "Administration Access".
- Password:** A text input field, followed by three radio buttons: "No Password", "Disable Password", and "Password Change required".
- Type:** A row of checkboxes: "Application User" (checked), "Full User" (checked), "Business Adm" (unchecked), "System Adm" (unchecked), "Inactive" (unchecked), and "Trusted" (unchecked).

At the bottom right of the dialog are "Edit" and "Cancel" buttons.

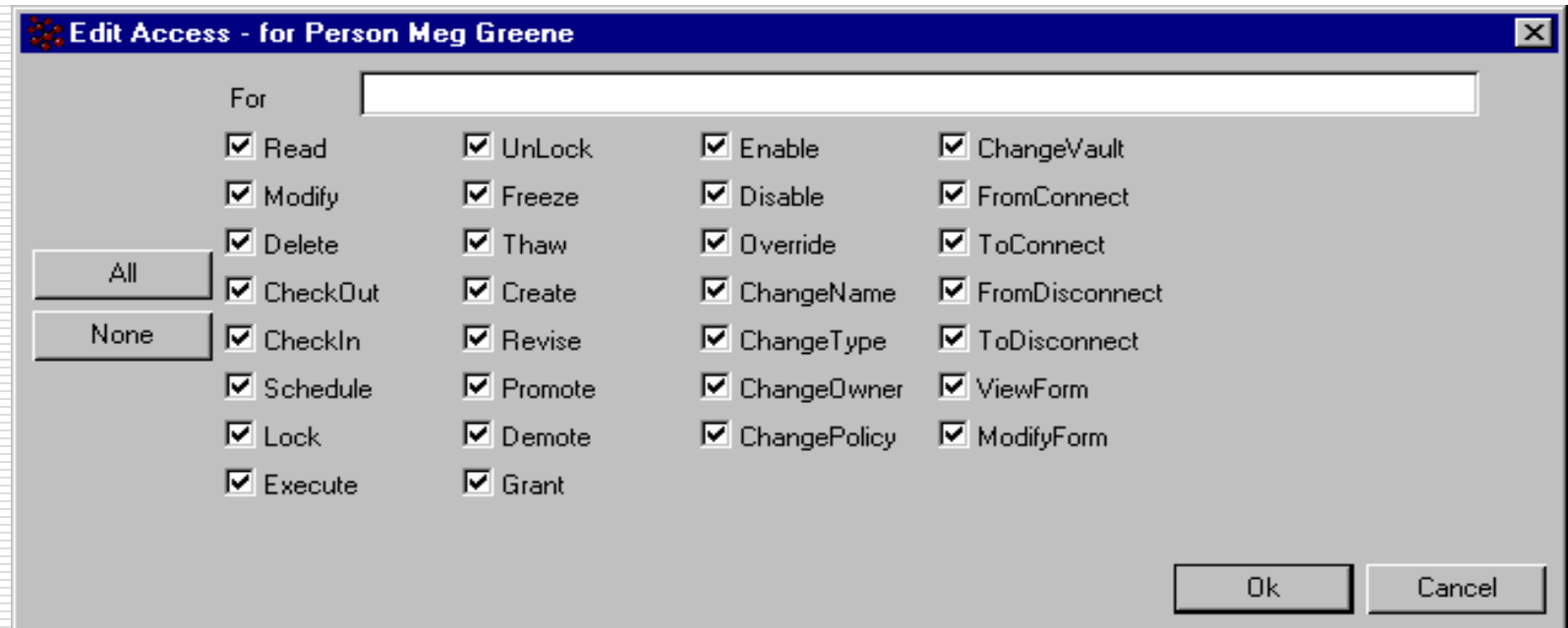
Privileges : Specify the access privilege of a Person

Access : Specify user level access

Password: set the password here.

Type : Specify the user can access application or business or system.

Person - Privileges



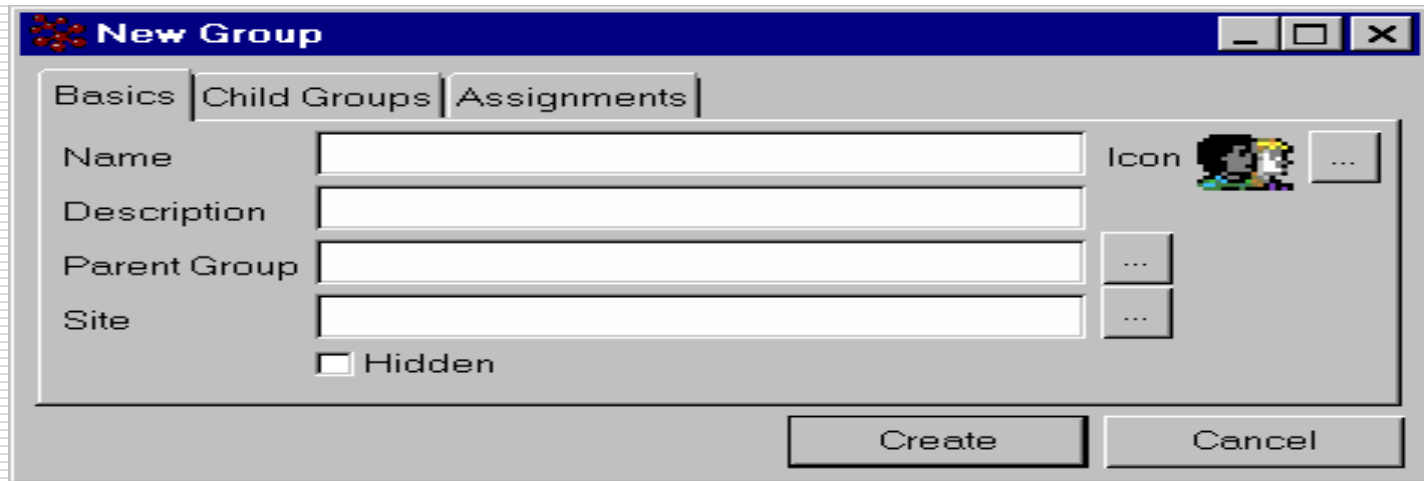
User can edit Access Privileges

Groups



- Identifies a set of people who are members of the same organization
- A group shares information
- A group may share access to business objects
- Creating a group saves from listing every user in a policy definition
- It is easier to build and maintain a group than to specify individual users in all policy definitions

Creating Group



The screenshot shows a 'New Group' dialog box with the following fields and controls:

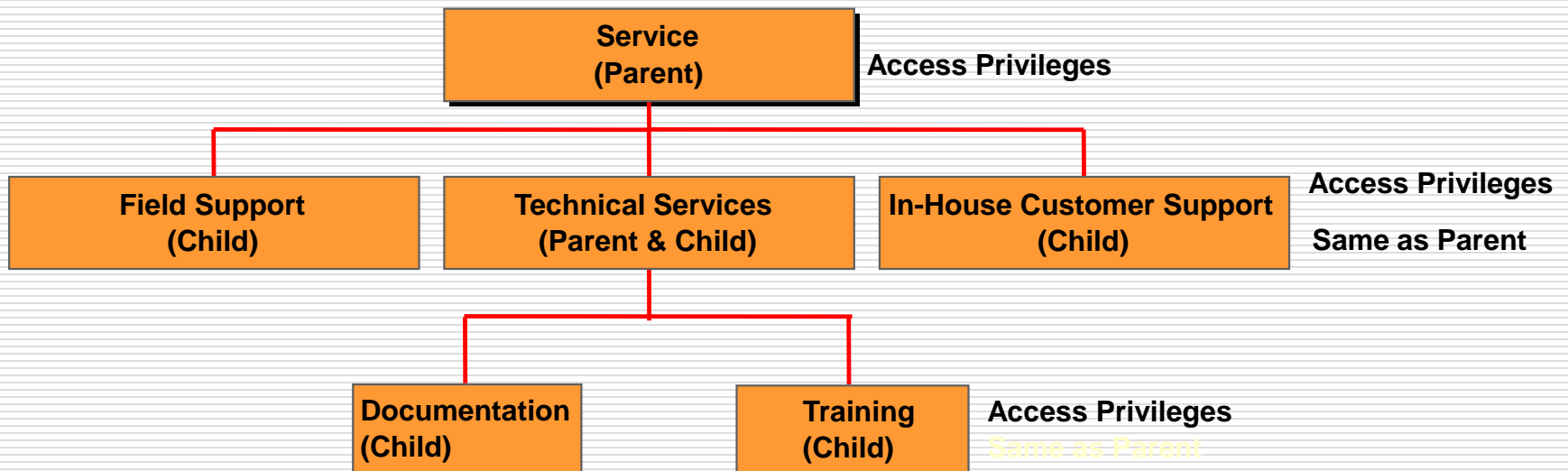
- Basics** (selected tab):
 - Name: Text input field
 - Description: Text input field
 - Parent Group: Text input field with a dropdown arrow
 - Site: Text input field with a dropdown arrow
 - ☐ Hidden
- Child Groups**: Tab label
- Assignments**: Tab label
- Icon**: Image selection field with a small icon and a dropdown arrow
- Create**: Button
- Cancel**: Button

Basics : Specify the Name ,Description , Parent Group and Site for a Group.

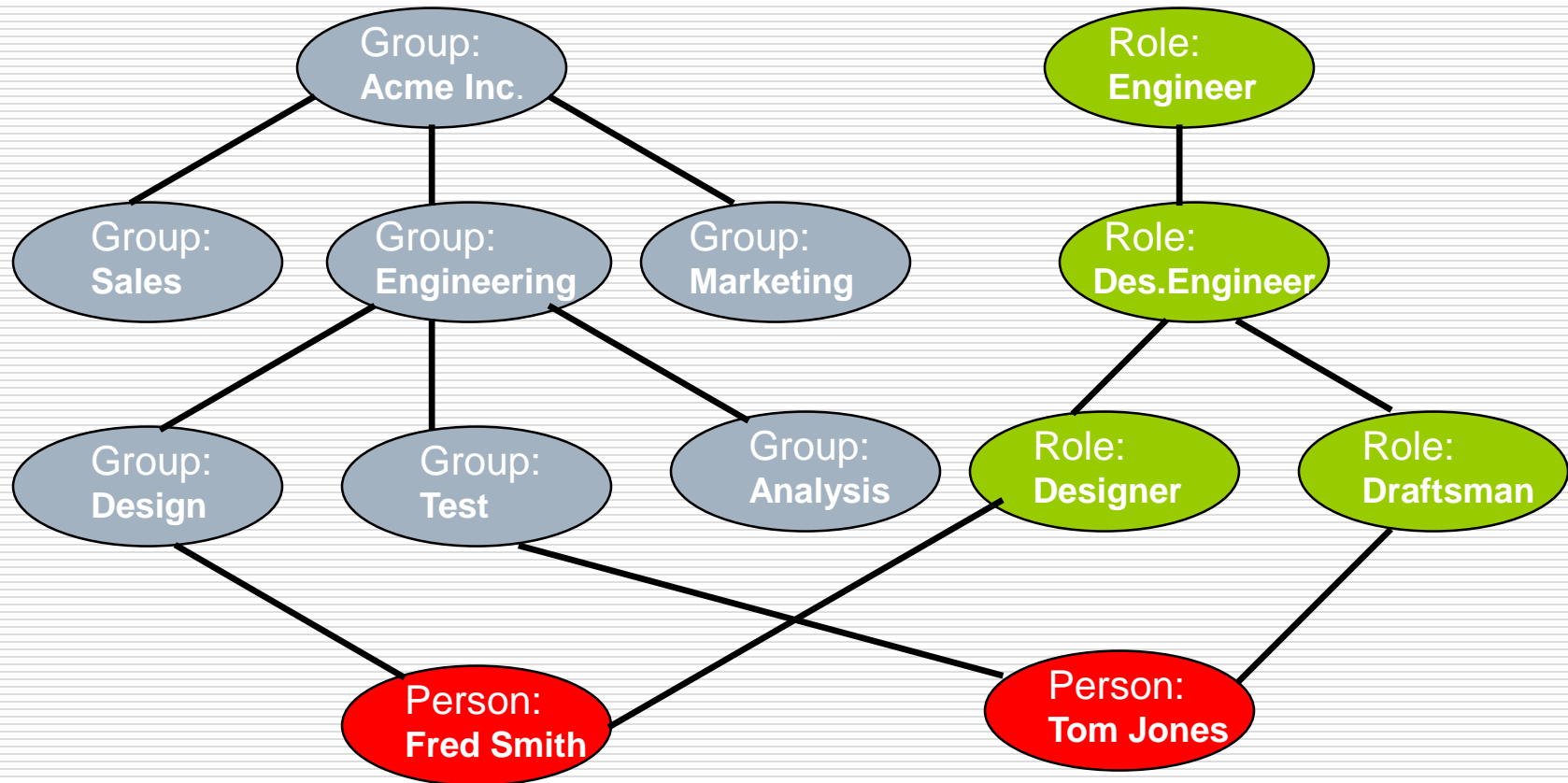
Child Groups: Specify some Group as a Child Group.

Assignment : Add Group and Role assignment to a Person.

Group Hierarchy



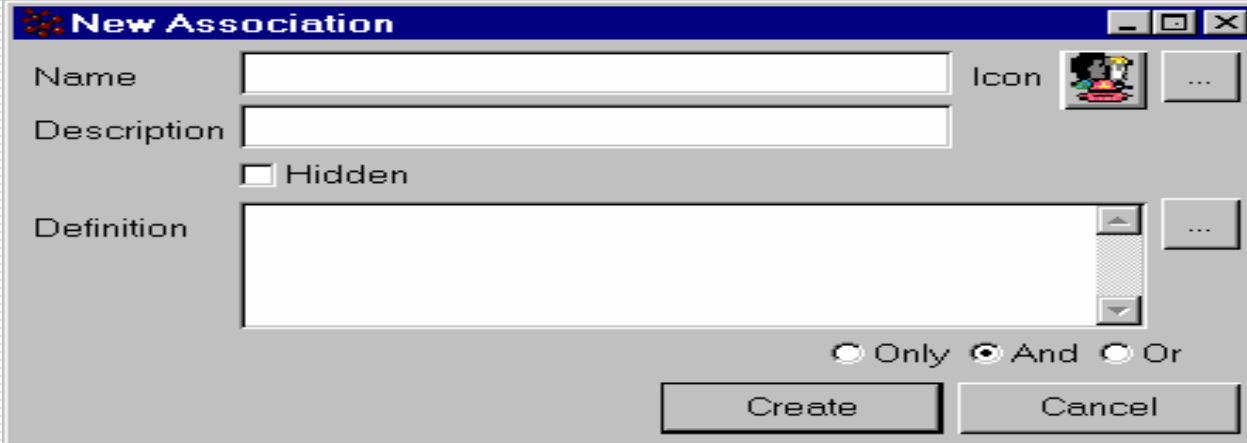
Relationship Between Groups, Roles and Persons




Associations

- An association is a grouping of individuals based on common information or aspects that cannot be defined by a Group, Role or Person

Creating Association



New Association

Name Icon  ...

Description ☐ Hidden

Definition ...

☐ Only ☒ And ☐ Or

Create Cancel

Name : Specify the Name for Association

Description: Specify some description for Association.

Definition : Add Group and Role assignment to a Person.

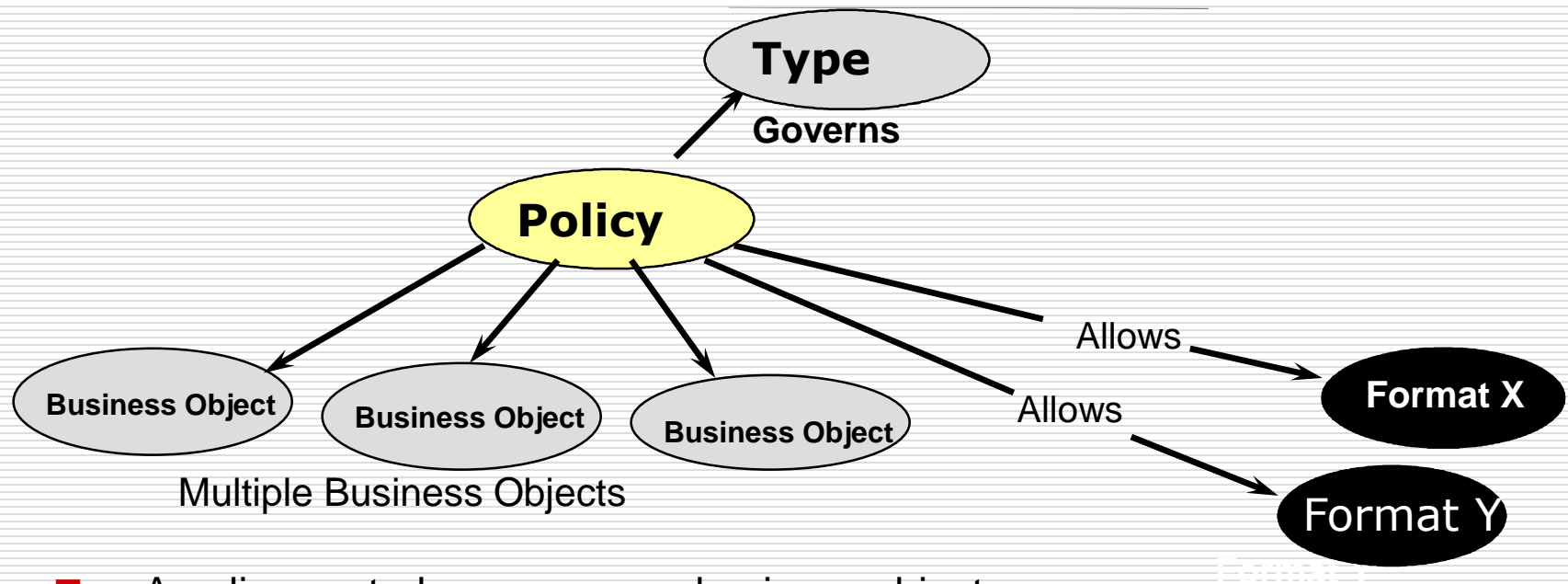
Agenda

Business Modeling Interface

Matrix Schema, Its Object and Relationship

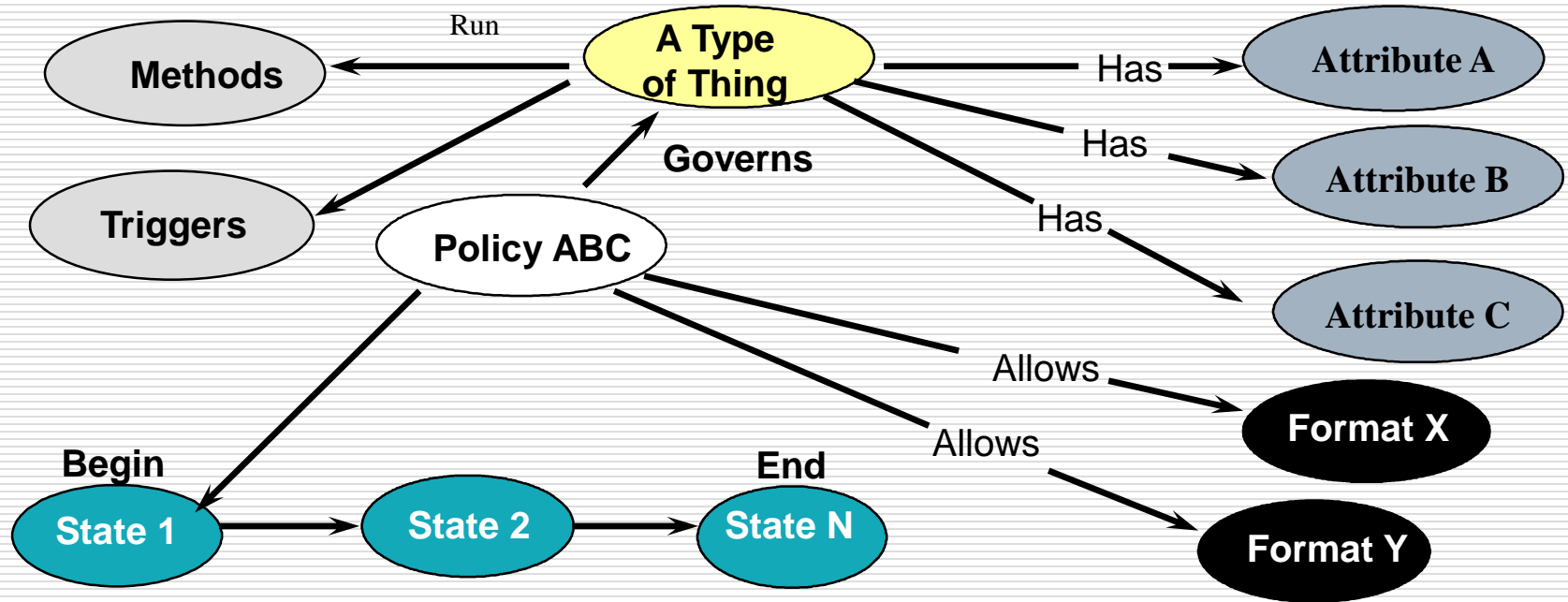
Design and Implement a Matrix Schema

Policy



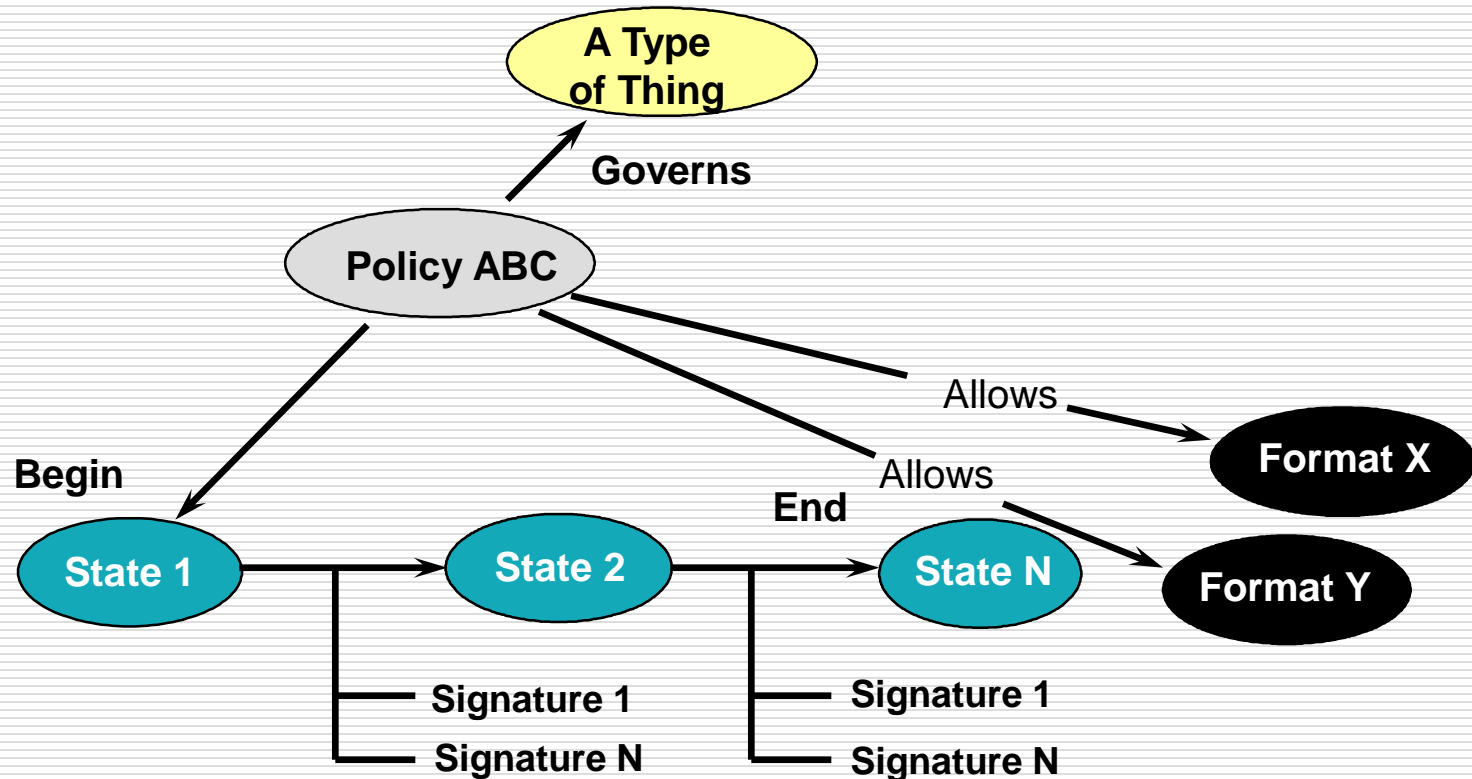
- A policy controls one or more business objects
- It contains the rules that govern access, approvals, life cycle, versioning, revisioning, file formats, stores, notification messages and routing messages

Policy States



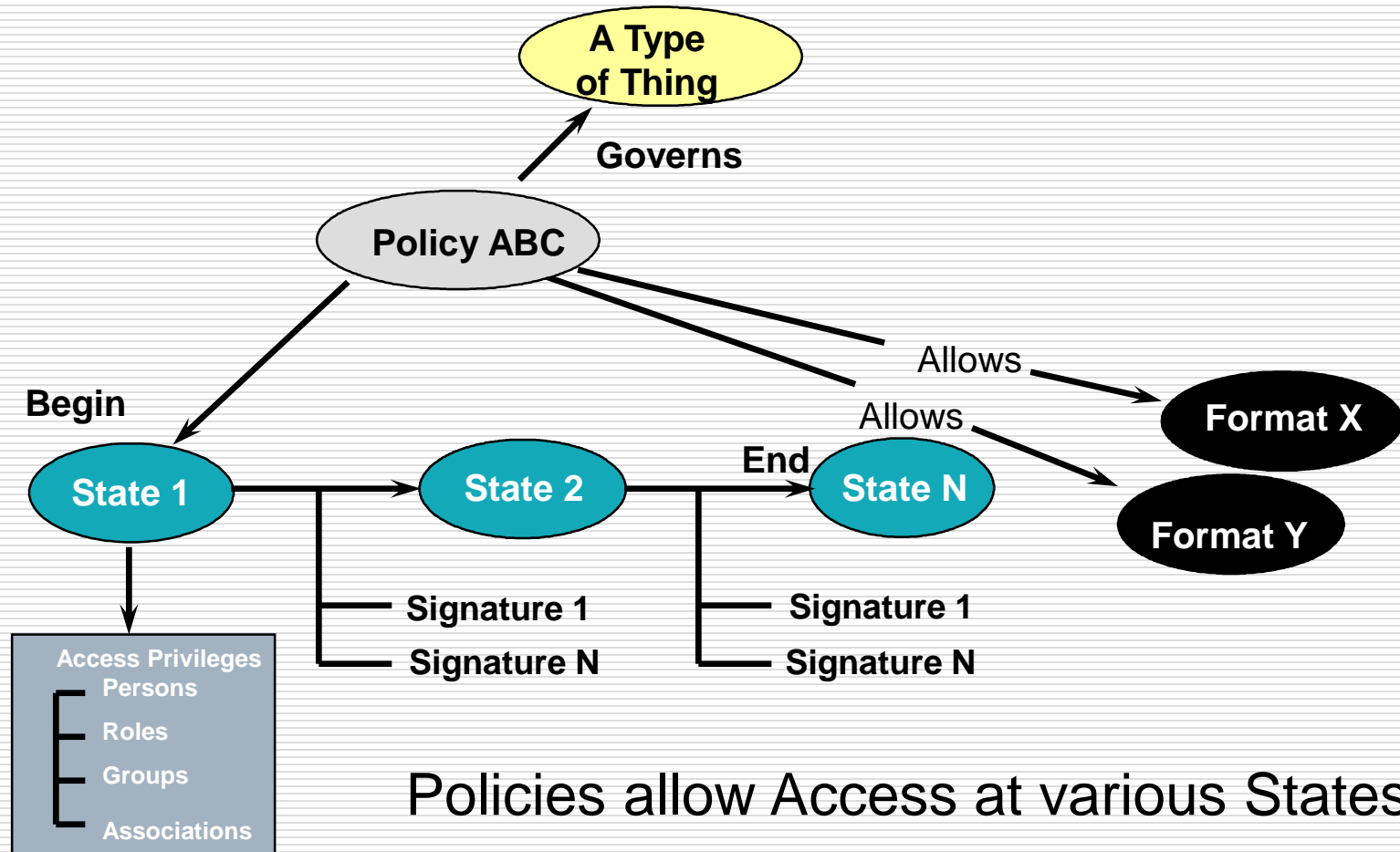
- Policy contains states that define the life cycle of the Business Object
- States have a definite beginning and ending
- The number of states depends on the business process at a particular company

Policy States



Signatures required to “PROMOTE” Business Object from State 1 to State 2

Policy States



Creating Policy

The screenshot shows a 'New Policy' dialog box with the following elements:

- Tabbed Interface:** Four tabs are visible: 'Basics' (selected), 'GovernedTypes', 'AllowedFormats', and 'States'.
- Fields:**
 - Name:** A text input field.
 - Description:** A text input field.
 - Sequence:** A text input field.
 - Store:** A text input field.
 - Hidden:** A checkbox.
 - Icon:** A field containing a folder icon and a browse button (...).
- Buttons:** 'Create' and 'Cancel' buttons at the bottom right.

Basics : Specify the Name, Description ,Sequence and Store

Governed Types: Specify some Types.

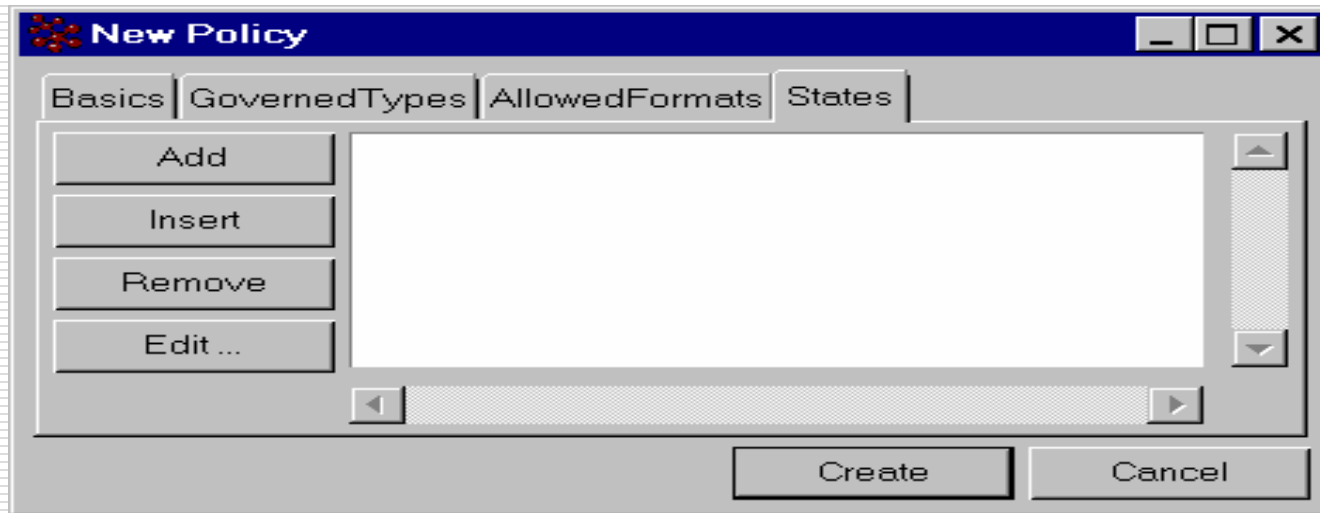
Allowed Formats : Add file format

States :Specify Various states for a Policy

Policy Information

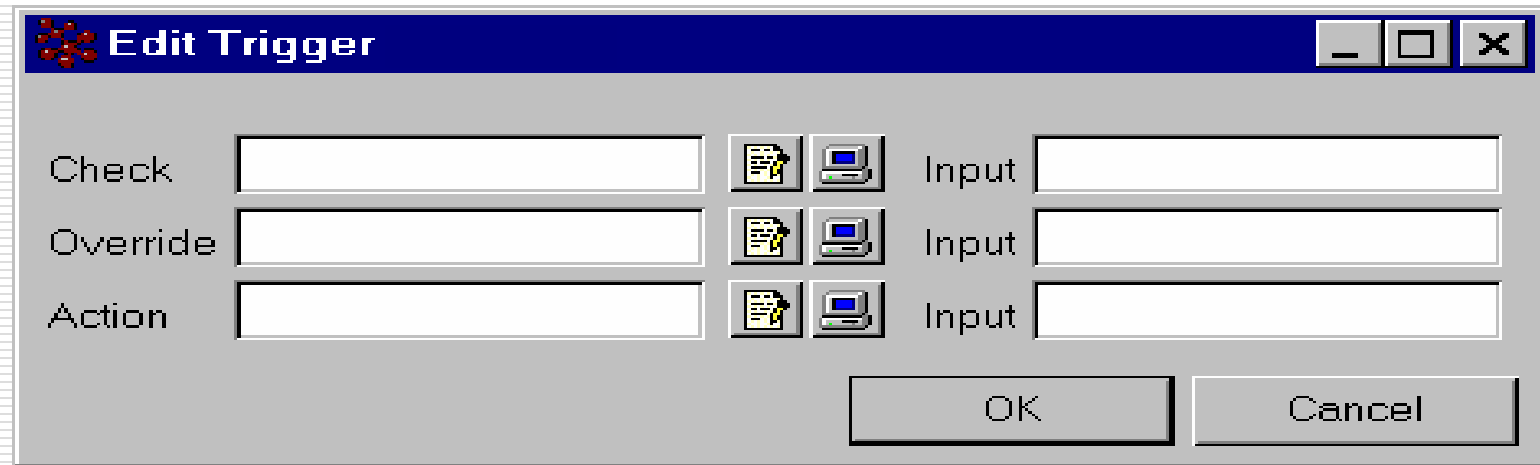
- Parts of a Policy
 - General Information about the Policy
 - States information about the Policy
- General Information
 - Controls the existence of the object instance
 - Provides information about the existence of the Business Object, the types of objects the Policy will govern, the types of allowed formats, the format automatically assigned to each object, where and how checked in files are managed and how revisions are labeled
- States Information
 - Controls the object's creation and what can be done to it after it has been created
 - Provides information about the states or stages in the life cycle of an object
 - The states define who has access to the object, what type of access is allowed, whether the object can be revised and the conditions required for changing state







Creating Policy States



State level user can add state inside Policy and user can insert, edit and remove after creating a state

Edit Trigger in State



Trigger Type	Text Input	Load Icon	Save Icon	Input
Check	<input type="text"/>			<input type="text"/>
Override	<input type="text"/>			<input type="text"/>
Action	<input type="text"/>			<input type="text"/>

OK Cancel

Check : Trigger program will be executed before the event occurs

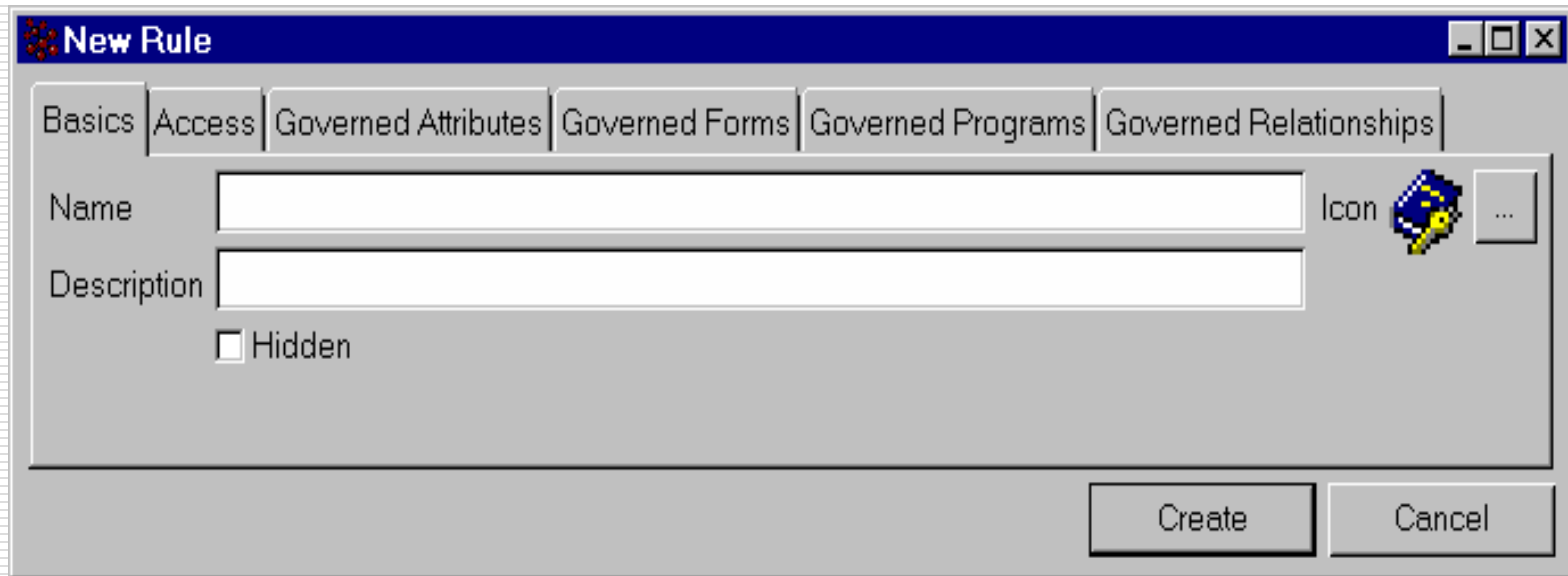
Override: Trigger program can replace the event transaction .

Action: Trigger program will be executed after the event occurs.

Rule

- Rules are administrative objects that define specific privileges for various Matrix users
- Rules define owner, public and user privileges as well as the Programs, Attributes, Forms and Relationships that they govern

Creating Rule




New Rule

Basics | Access | Governed Attributes | Governed Forms | Governed Programs | Governed Relationships

Name

Description

☐ Hidden

Icon 

Rule can prevent user access for Attribute, Forms, Programs and Relationships

Rule Access

- Public Access
- Owner Access
- User Access

Access

- **Public Access**
 - Refers to everyone in the Matrix database
 - Access limits should be defined when defining Public access
- **Owner Access**
 - Owner is the person who initially created the object in the database
 - Owner access refers to the specific person who is the current owner of the object instance
- **User Access**
 - User access enables to name a specific group, role, person or association who will have access to the object
 - User access provides the greatest control in restricting who will or will not have access to an object

Questions



Thank You !!!

Chennai