ENOVIA MatrixOne Fundamentals

Agenda

- ENOVIA Components
- Business Objects
- Person, Role, Group
- Files and File Format
- File Check In and Check Out
- Attribute, Type, Relationship, Policy
- IconMail
- Automating Processes
- Vaults and Stores

ENOVIA MatrixOne

INTELIZIGN

ENOVIA MatrixOne System is:

- A general-purpose information management tool
- Designed for multi-discipline teams
- In heterogeneous computer environments

ENOVIA MatrixOne provides:

- Communication mechanisms
- Process management facilities
- File storage and access controls
- Application integration

ENOVIA Components

ENOVIA MatrixOne system is a client/server software suite consisting of the following components

- **Business Modeler** - Application that is used to model the types of objects used in our business along with the attributes, process rules and persons associated with those objects.
- **Matrix Navigator** - Application that is used to create specific instances of the objects that were defined in the Business Modeler.
- **System Manager** - Application that is used to perform activities related to the configuration and maintenance of database storage locations.
- **Matrix Query Language(MQL)** – A command driven interface to all the above components. Helps to set up and test a Matrix database quickly and efficiently.

ENOVIA Components cont...

The following components are also provided as part of ENOVIA System

- **Collaboration Server**

  When access to the database is provided through web browsers, a web or application server as well as one of the Matrix collaboration servers is required. The collaboration server is available in RMI or EJB varieties. The collaboration server includes several servlets used to communicate with the Matrix collaboration kernel.

- **Web Navigator**

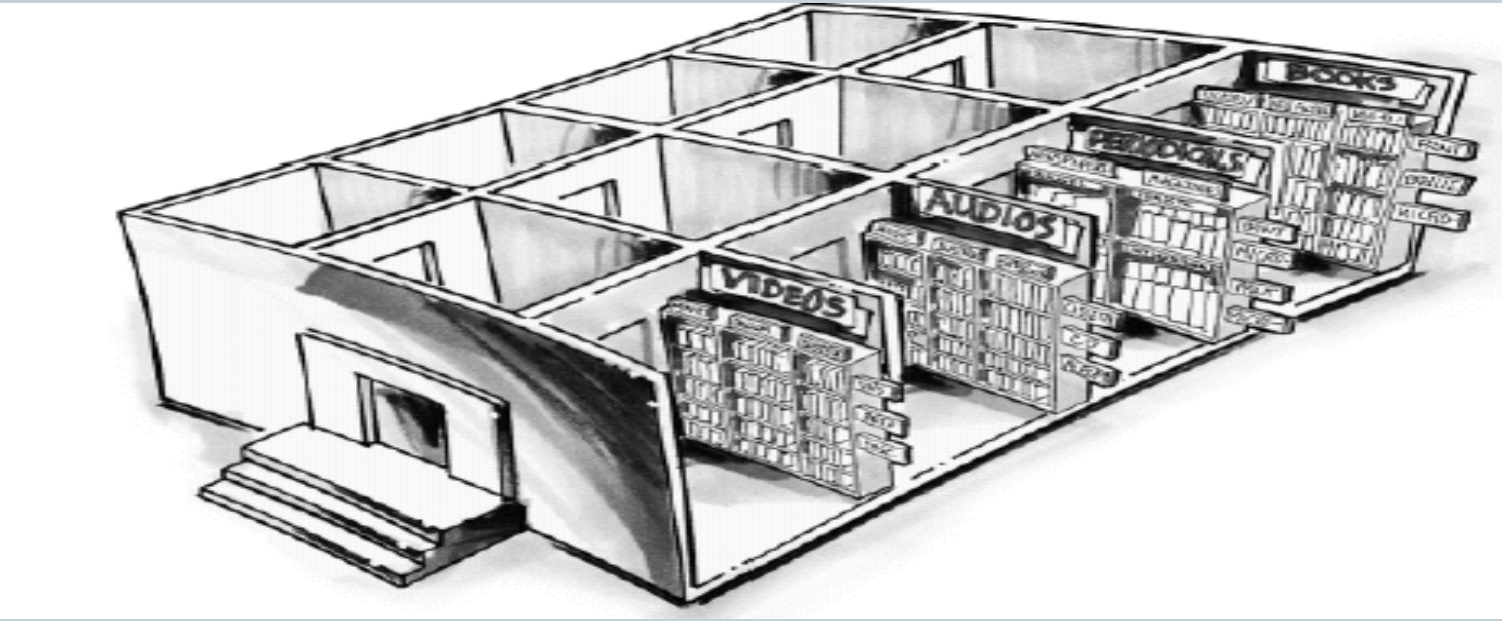
  Web Navigators are used to access the Matrix database from any Web browser.

Business Objects

Everything is a Business Object in ENOVIA!!!

ENOVIA manages an object's documents and related information.

Business Object is an item within Matrix that is used to contain, control and manipulate information.



Business Object - Basics

Type

Name

Revision

Owner

Vault

Description

Policy

Date Created (Originated)

Date Modified (Modified)

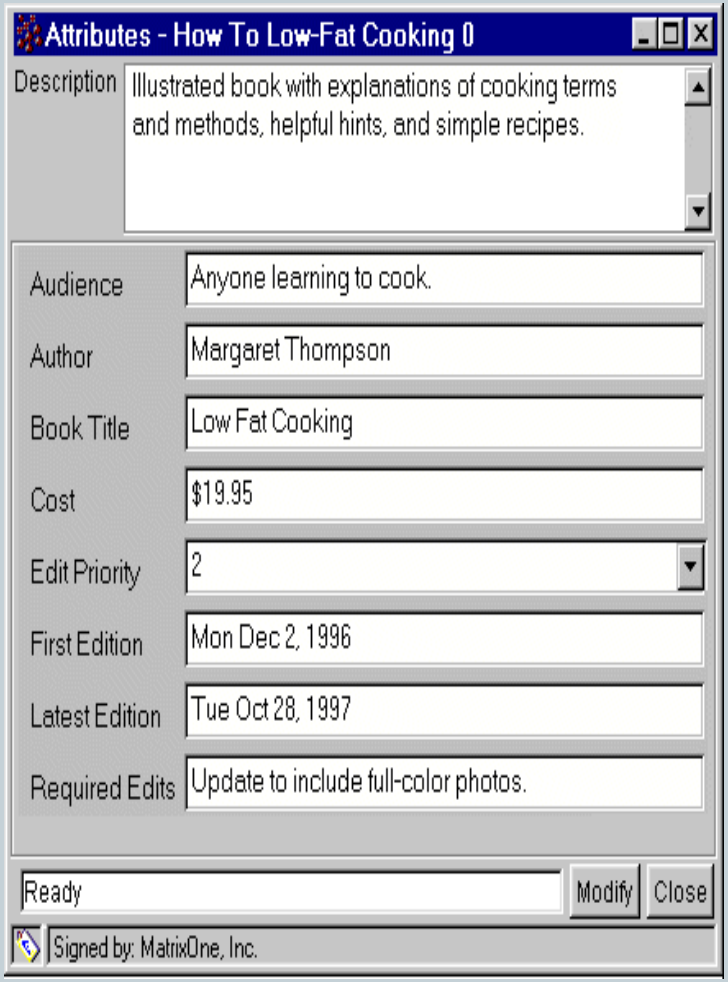
Grantee

Current State

Object Attributes

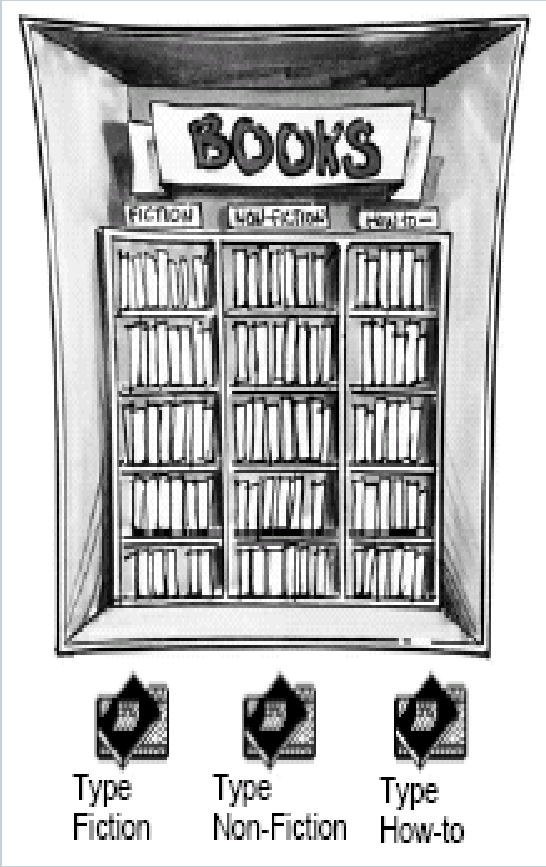
- **Attributes further define an object within Matrix.**

- An attribute is any characteristic that you can assign to an object or to its relationship with other objects.



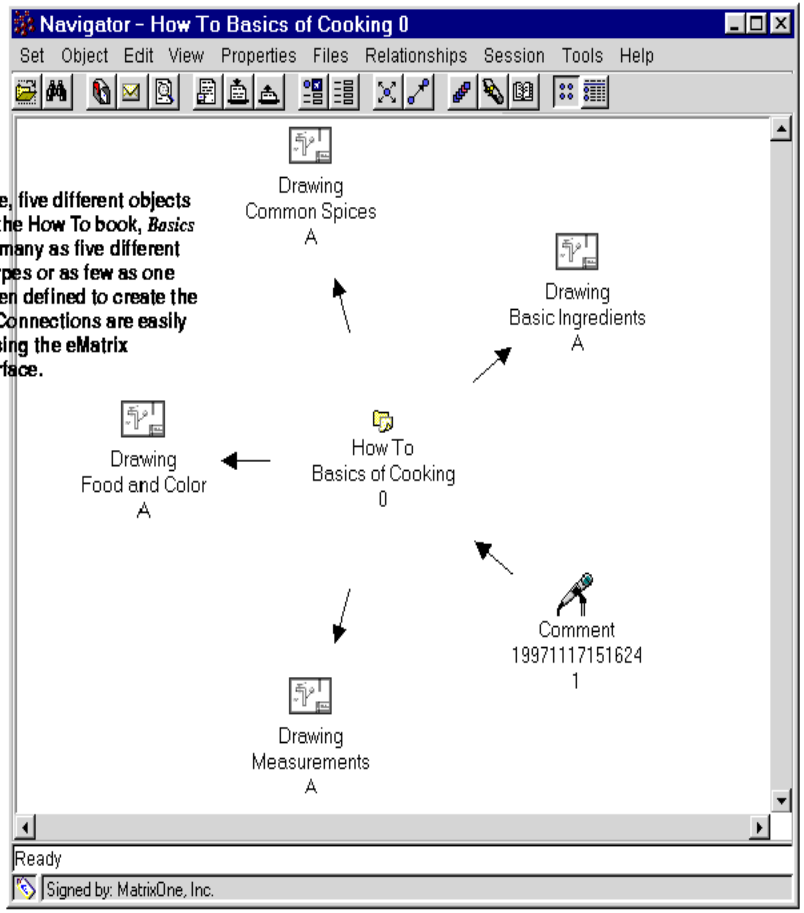
Business Object Types

- A type defines a business object.
- Each business object has a name and is defined as a specific type having specific characteristics (attributes). A type defines a kind of business object and the collection of attributes it can have.
- A type can have any number of sub-types, creating a hierarchy.
- Each business object in Matrix is uniquely identified by its type, name, and revision (TNR).



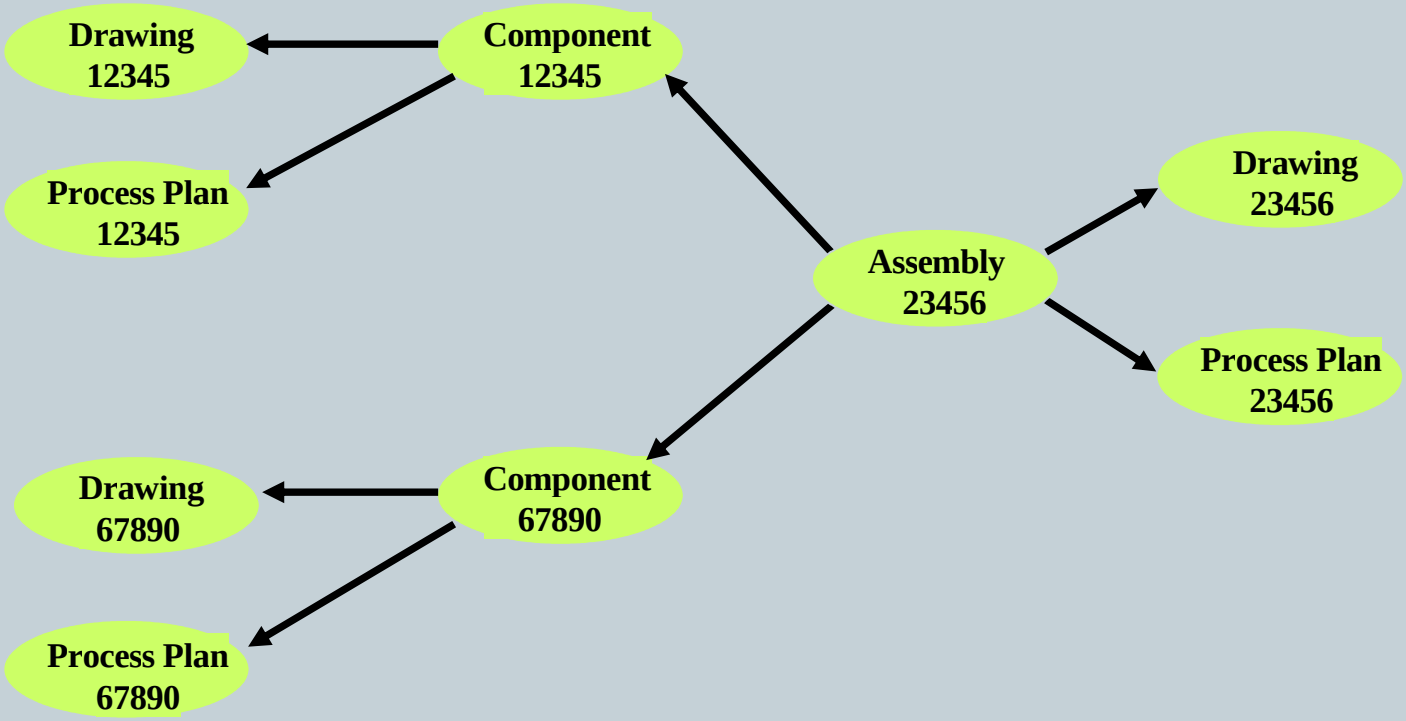
Object Relationships

- **Matrix shows you how objects are organized and related.**
- A relationship is a type of connection made between associated business objects. These connections enable you to see how one business object relates to other objects.
- A Matrix database typically will have many different types of relationships, each specifying the types of objects it can connect. The definitions determine the meanings of each end of the relationship as well as any attributes it may have.

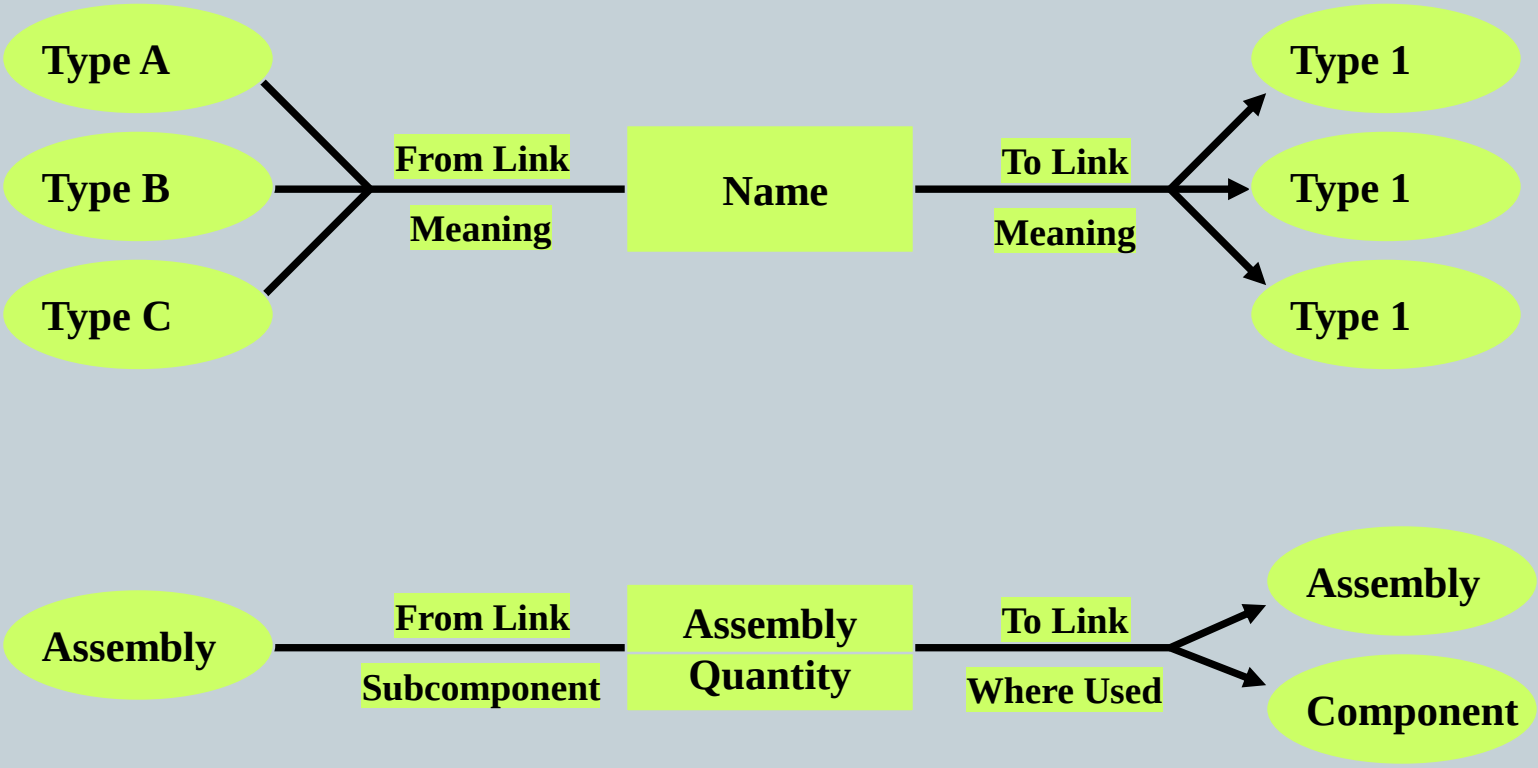


eMatrix Navigator Browser in Star mode, with Icon option selected.

Objects can be Connected with Relationships

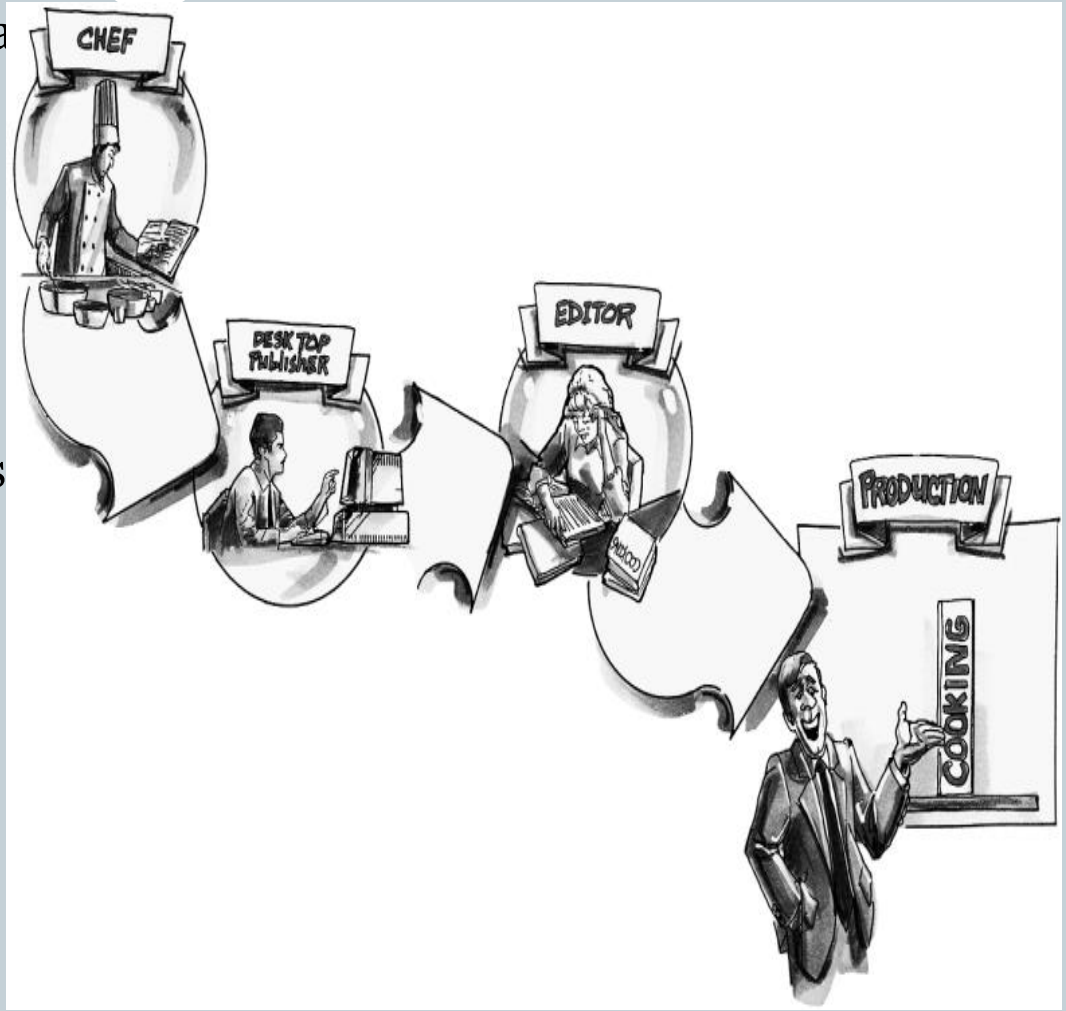


Relationships are Objects Too



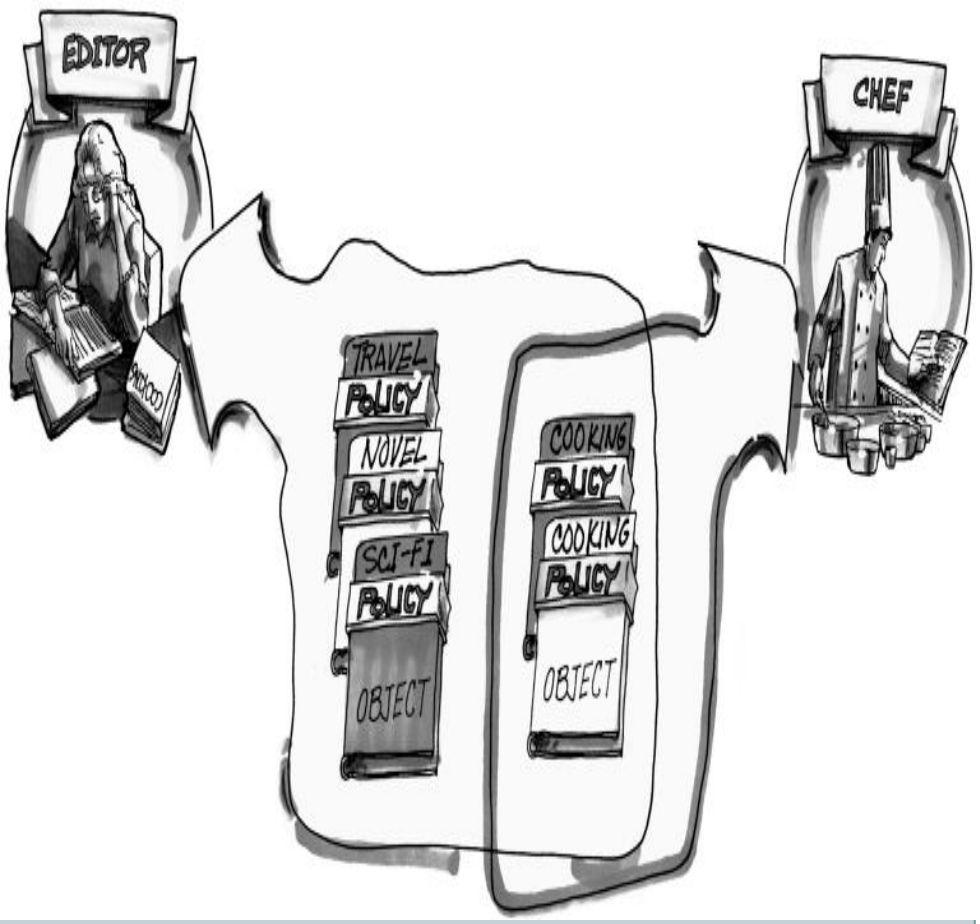
Object Lifecycle

- **Each business object has a lifecycle.**
- A lifecycle is a series of states through which a business object passes during its existence.
- When a new business object is created, its policy defines a lifecycle—the states through which it will pass from inception to completion.
- The policy governs all activities during each lifecycle state. The policy also defines the conditions that must be met for the promotion.

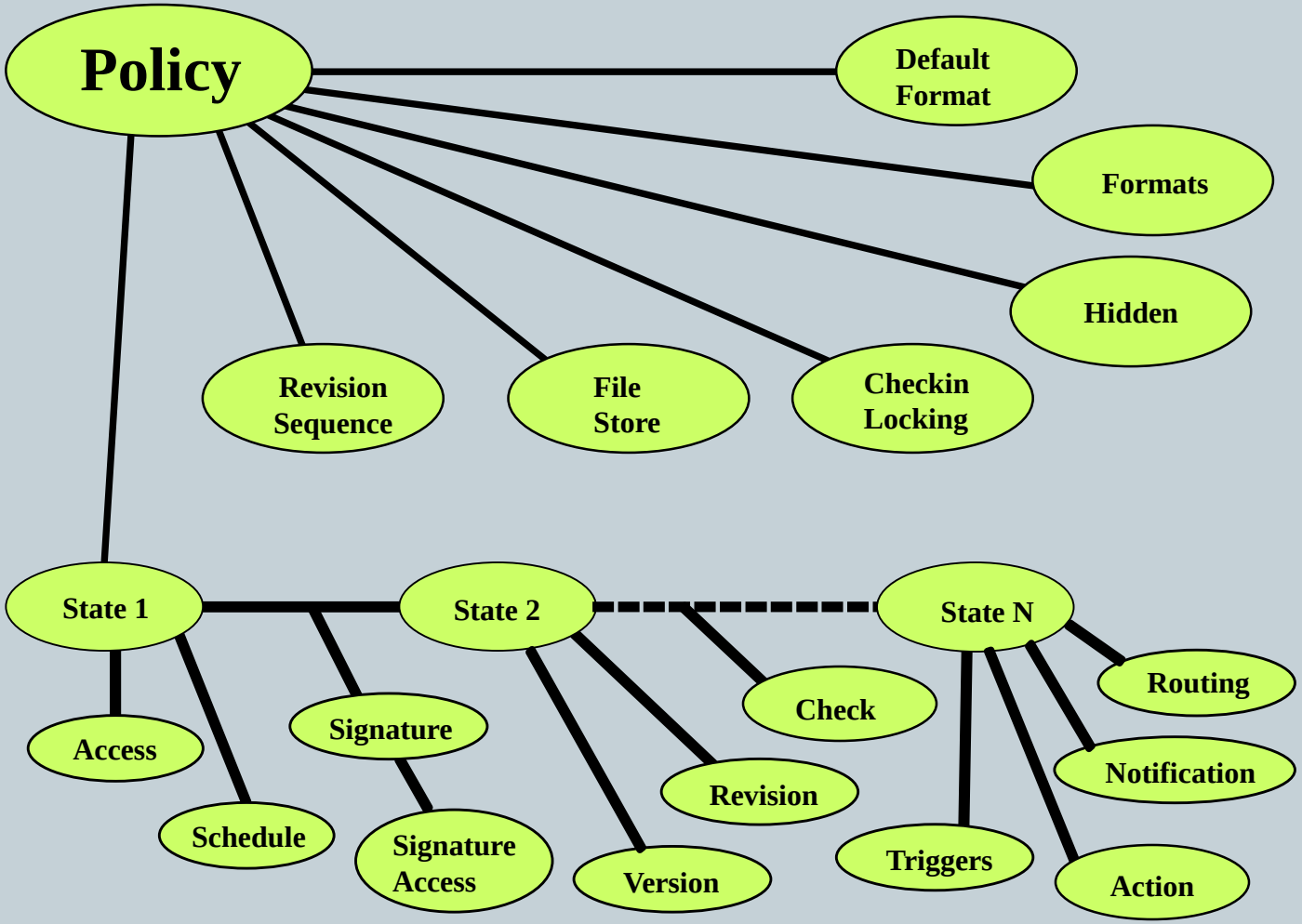


Policies

- **A business object is controlled by its policy.**
- A policy is a set of rules that governs the behavior of a business object—its lifecycle, access privileges, revisioning schemes, the file formats associated with the object, and where/how checked-in files are stored in the database.
- Within the policy, various types of access are defined for the roles people have in an organization.
- A policy may govern more than one type of object, and there may be more than one possible policy for any one object type. But specific business objects follow the lifecycle and access rules of just one policy at any one time.



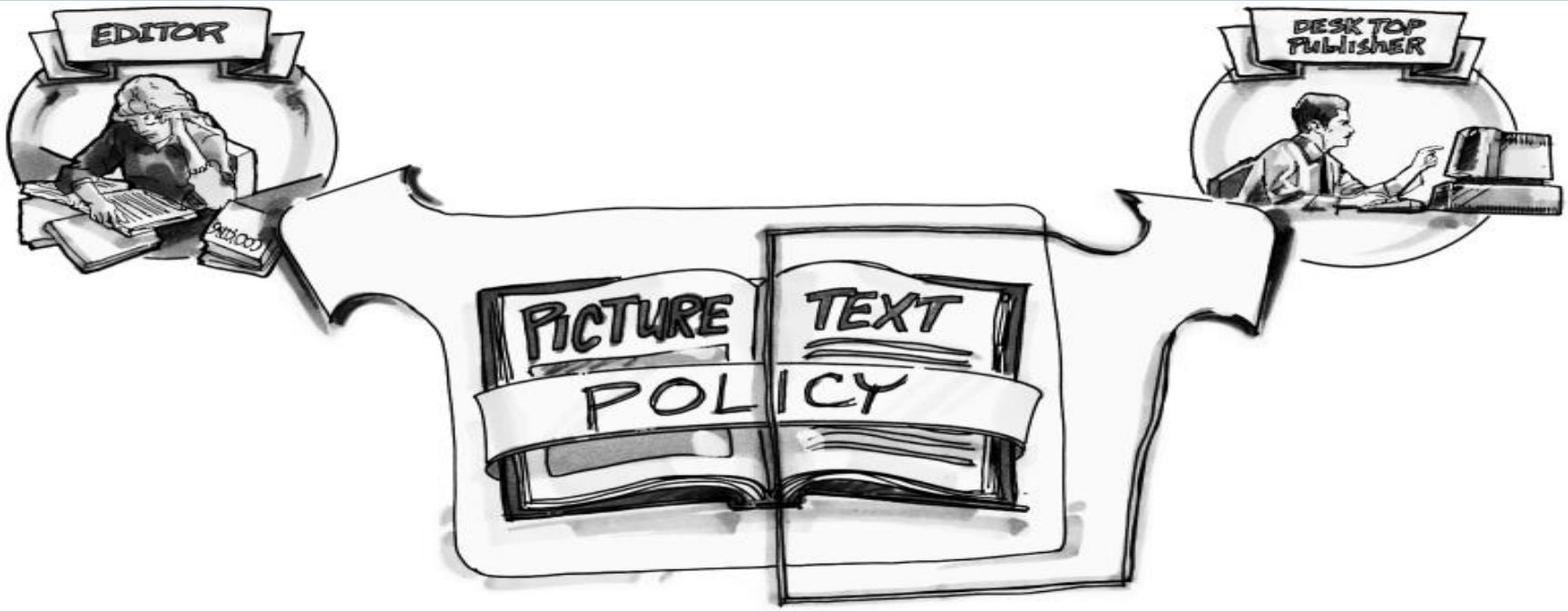
The Policy is Central to Matrix Behavior



Policies and Lifecycle

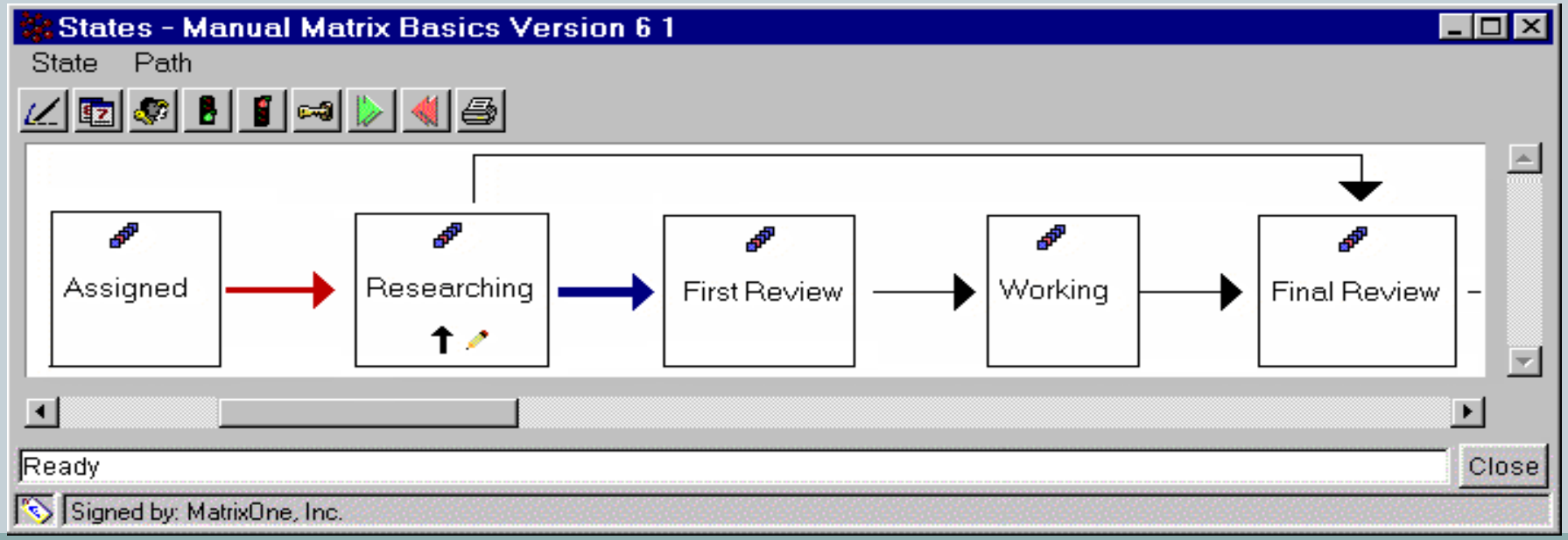
- **Privileges may vary during an object's lifecycle.**

- A policy governs the various states through which the object will pass during its lifecycle, the people who have access at each state, and the conditions required for changing from one state to another.



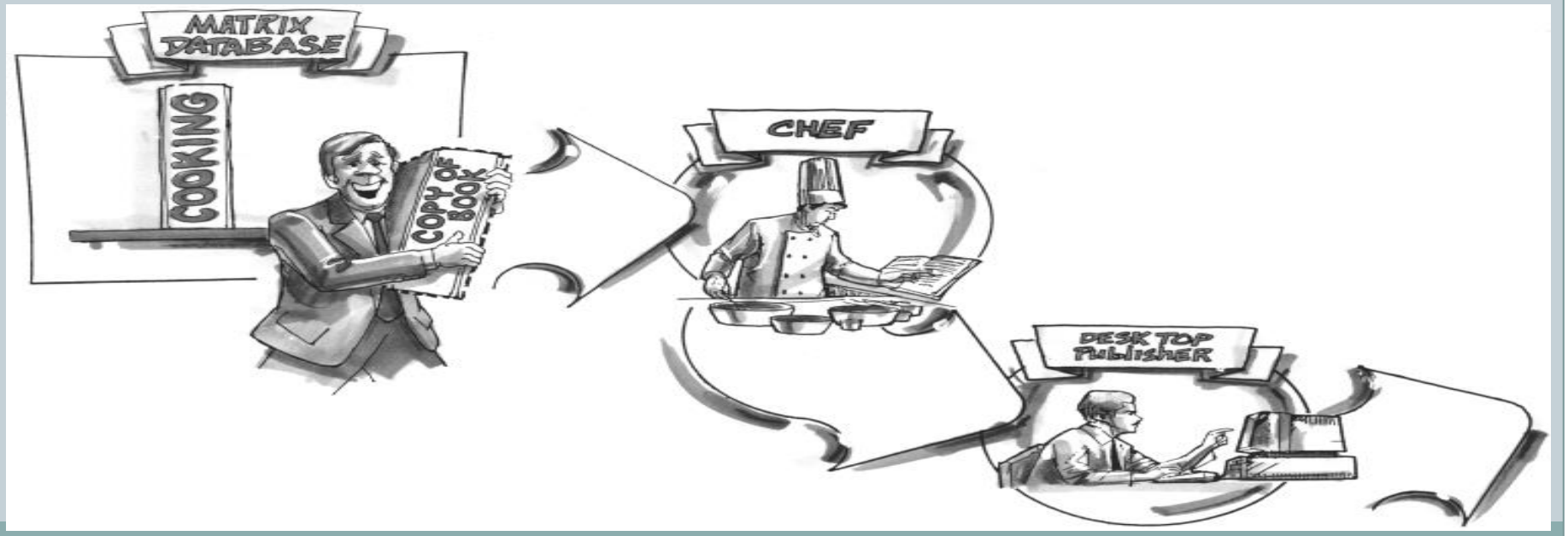
States and Signatures

- **Approval allows promotion of an object to the next state.**
- Within Matrix, the lifecycle that has been defined by an object's policy is displayed in the State browser.
- Signatures provide a means to authorize the promotion of an object, and to which state (branch) it will move.



States and Ownership

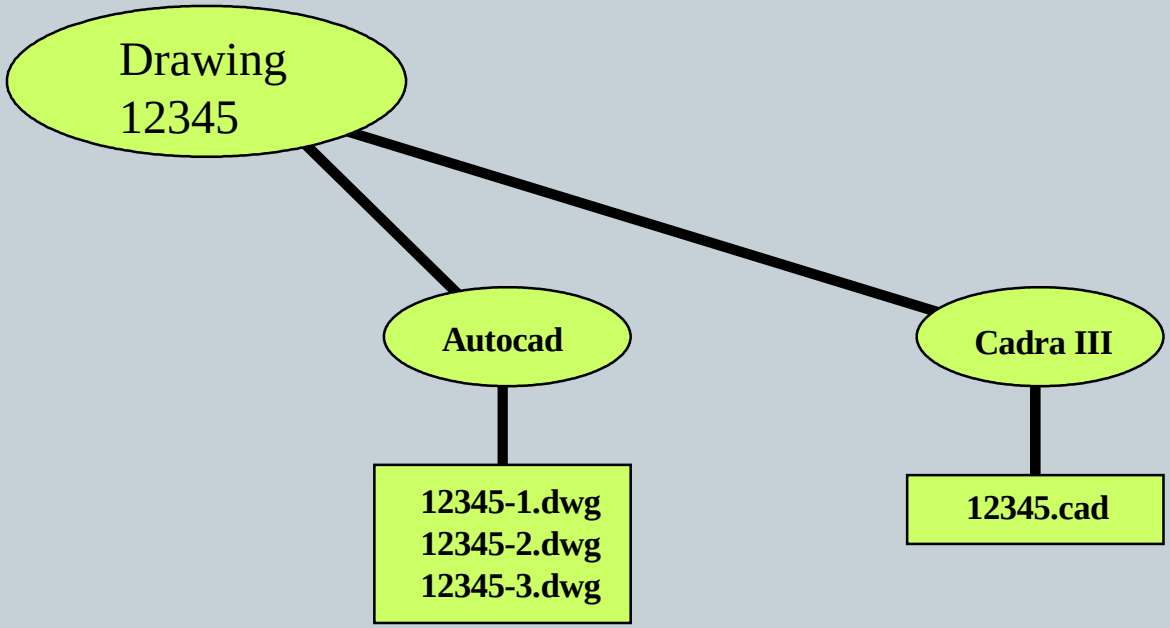
- **Ownership of a business object may change in each state.**
- At each state in a lifecycle, the object's policy defines the actions that can take place.
- The policy also specifies the person or groups of people who can perform each action.
- Ownership and access privileges of the object may change in each state, as specified by the policy.
- Matrix retains the history of actions that take place on the business object.



States and Revisions

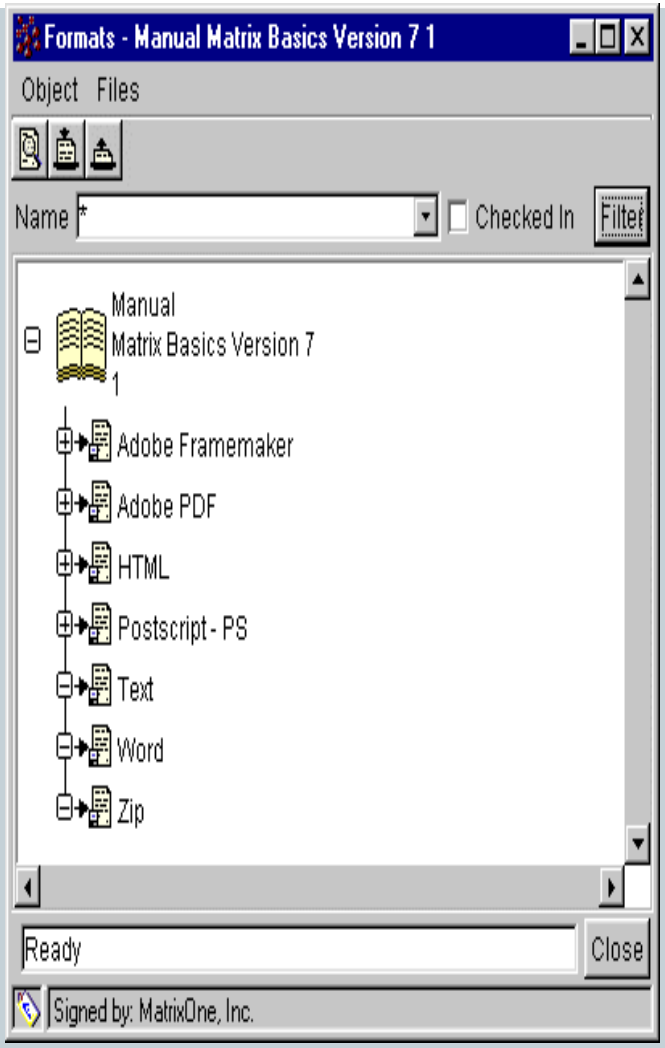
- **Revisions are usually associated with a change in state.**
- A revision of a business object is a special kind of copy of an object. An edition of a book is a revision. The revised object may have all of the same attribute values and the same policy—or these may vary.
- When an object is revised, its type and name remain the same, however, the revision label changes to identify the new object.
- The policy specifies the scheme for labeling revisions with letters and/or numbers. For example, revision labels might appear such as: AA, AB, AC or 110, 111, 112 or 1st Print, 2nd Print, 3rd Print.

Files are Stored in an Object's Formats



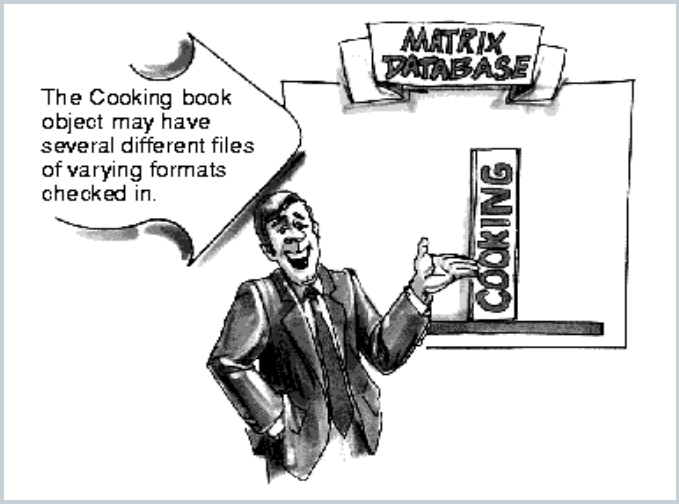
Files and Formats

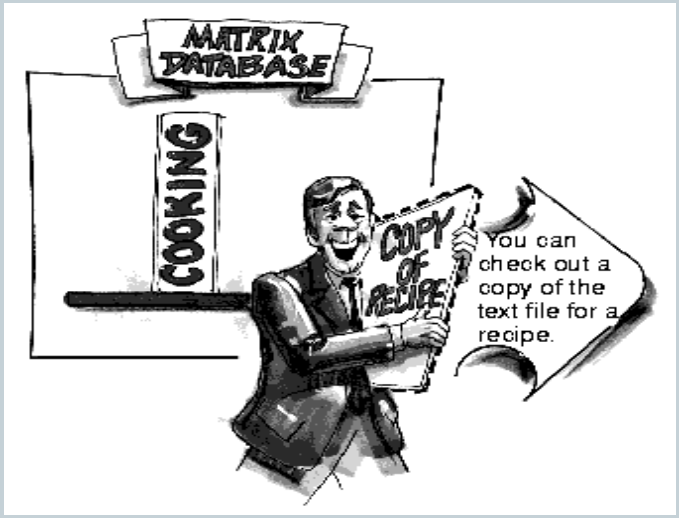
- **Business objects may have associated files.**
- A Matrix business object may be used to store application files that have been generated by virtually any software application. For example, a CAD file may be placed in a Drawing object in Matrix.
- A file *format specifies the type of file* (such as ASCII, HTML, etc.) that may be associated with an object.
- Types of file formats available for a specific business object are defined by the object's policy.
- Formats define:
  1. The application software and version ("FrameMaker 5.5")
  2. Default extension for the filename (.fm)
  3. The programs to launch when the edit, view, or print requests are received by Matrix.
- A single object may have files in various formats.



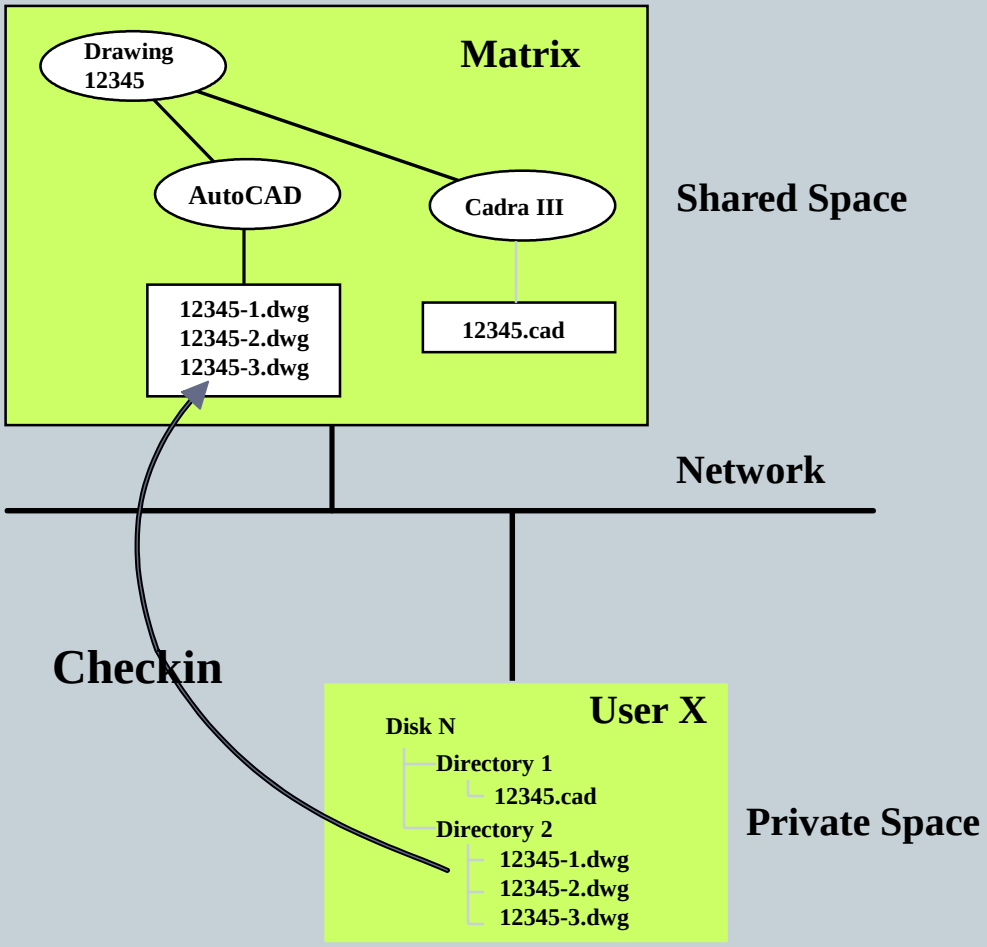
File Check In and Check Out

- File Check In is the process of placing a file into a business object from your local workstation.
- File Check Out is the process of retrieving a copy of a file contained in a business object to a workstation.
- A master copy of the file is thus under Matrix control at all times which ensures that the file is not deleted from the database unless explicitly deleted.
- During Check Out a file saved in Matrix can be locked so that no one can update the copy when one person is working on it.
- After the work is completed on the Checked-Out file, the file is again checked back into the database in order to update Matrix with the new version of the file

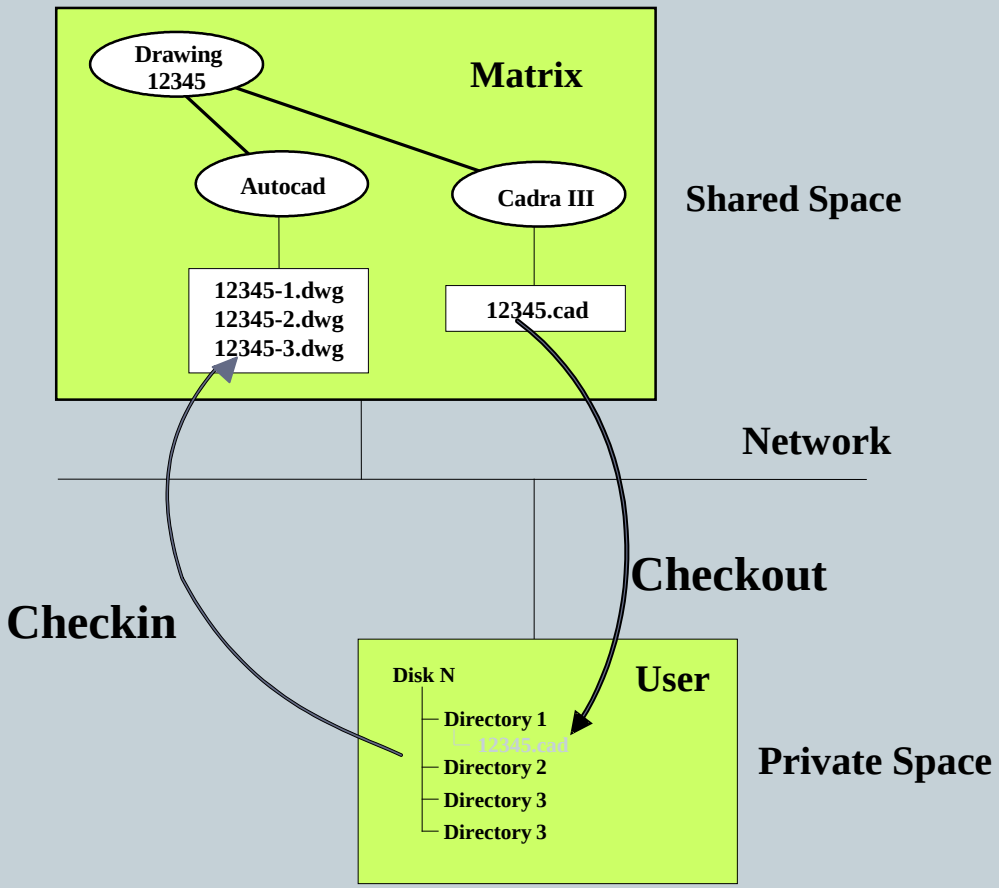




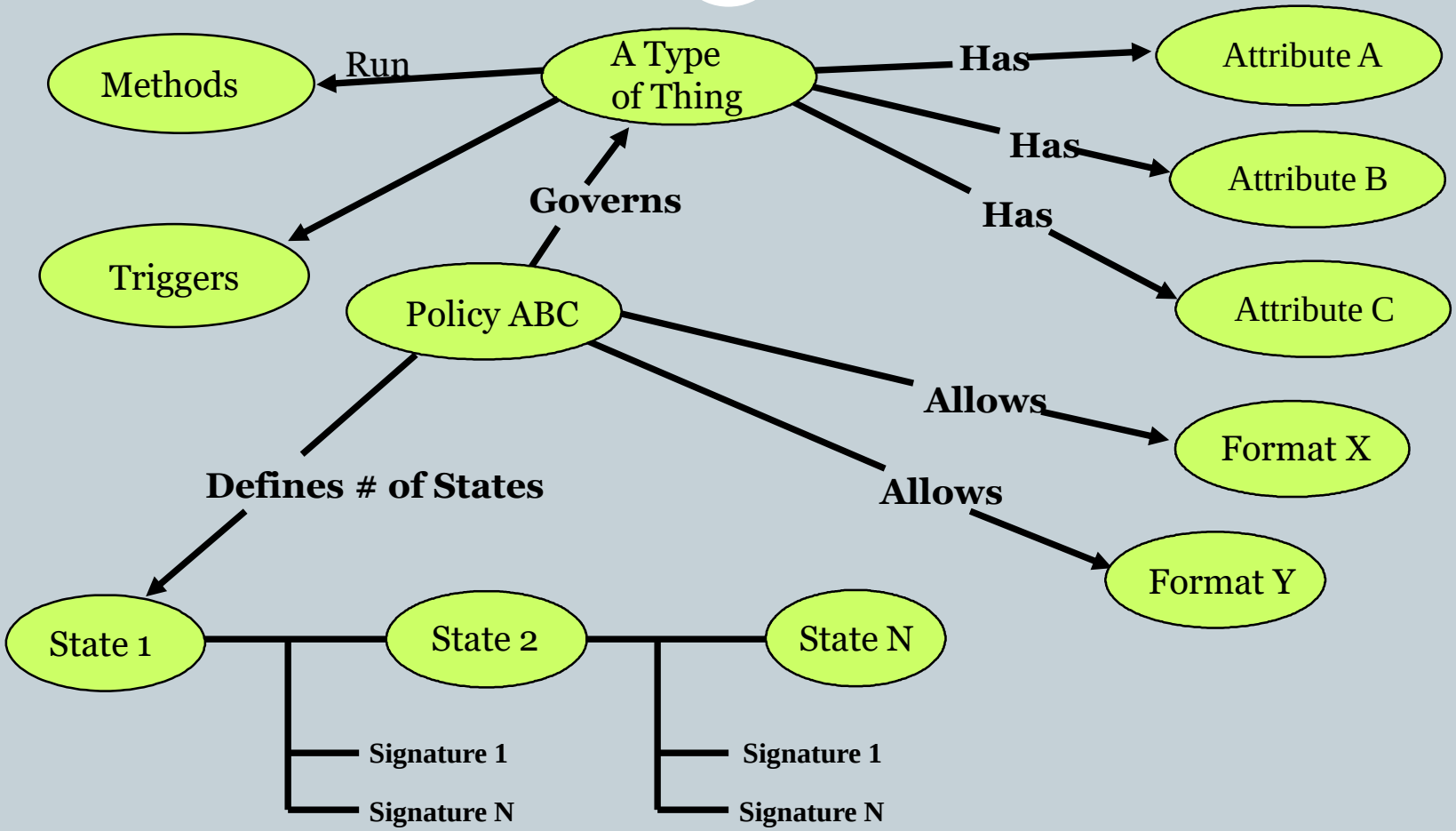
Files are put into an Object by Checkin



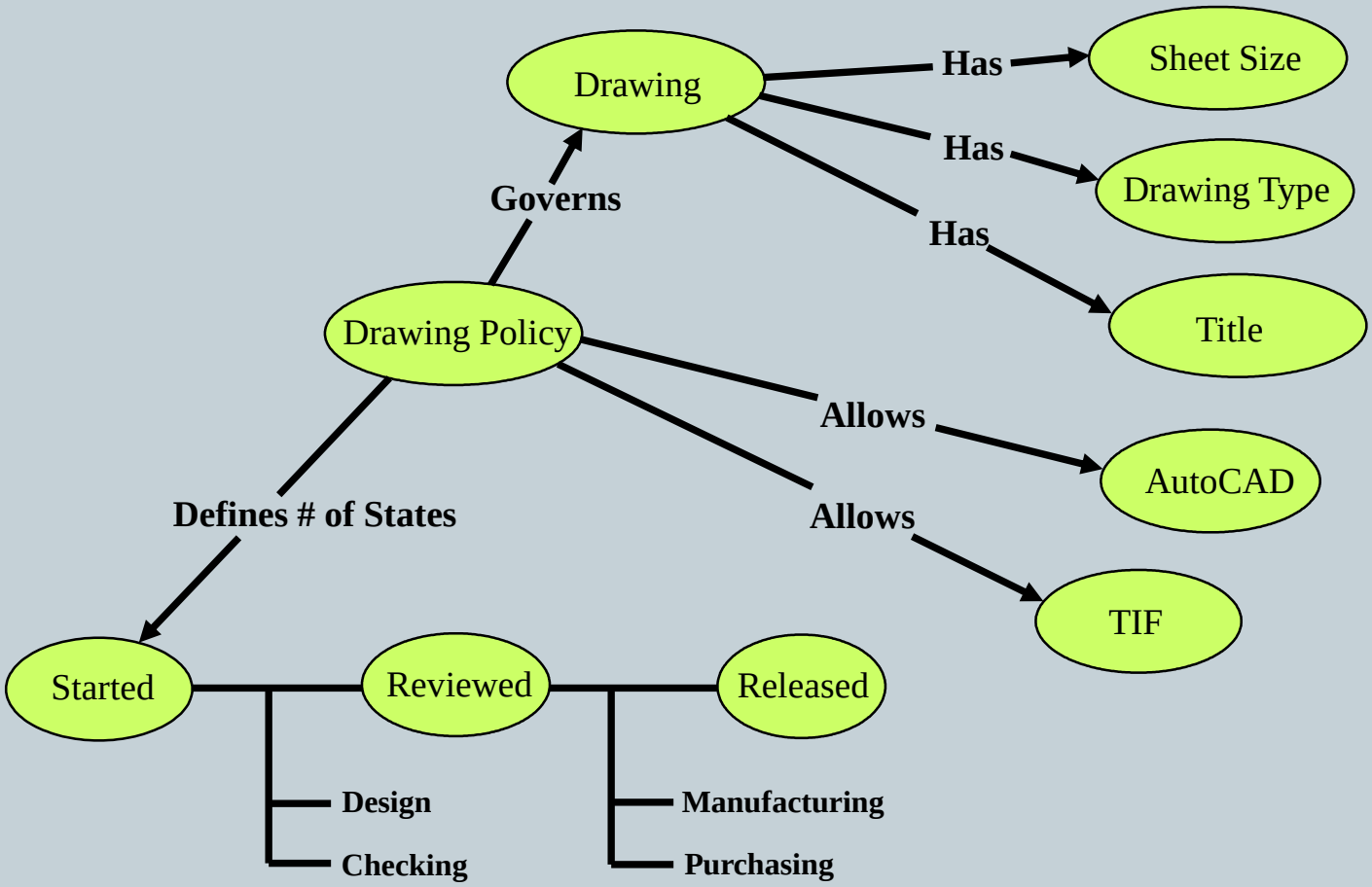
And obtained from one by Checkout



The Basic Matrix Schema



A Typical Object Definition



Persons and Roles

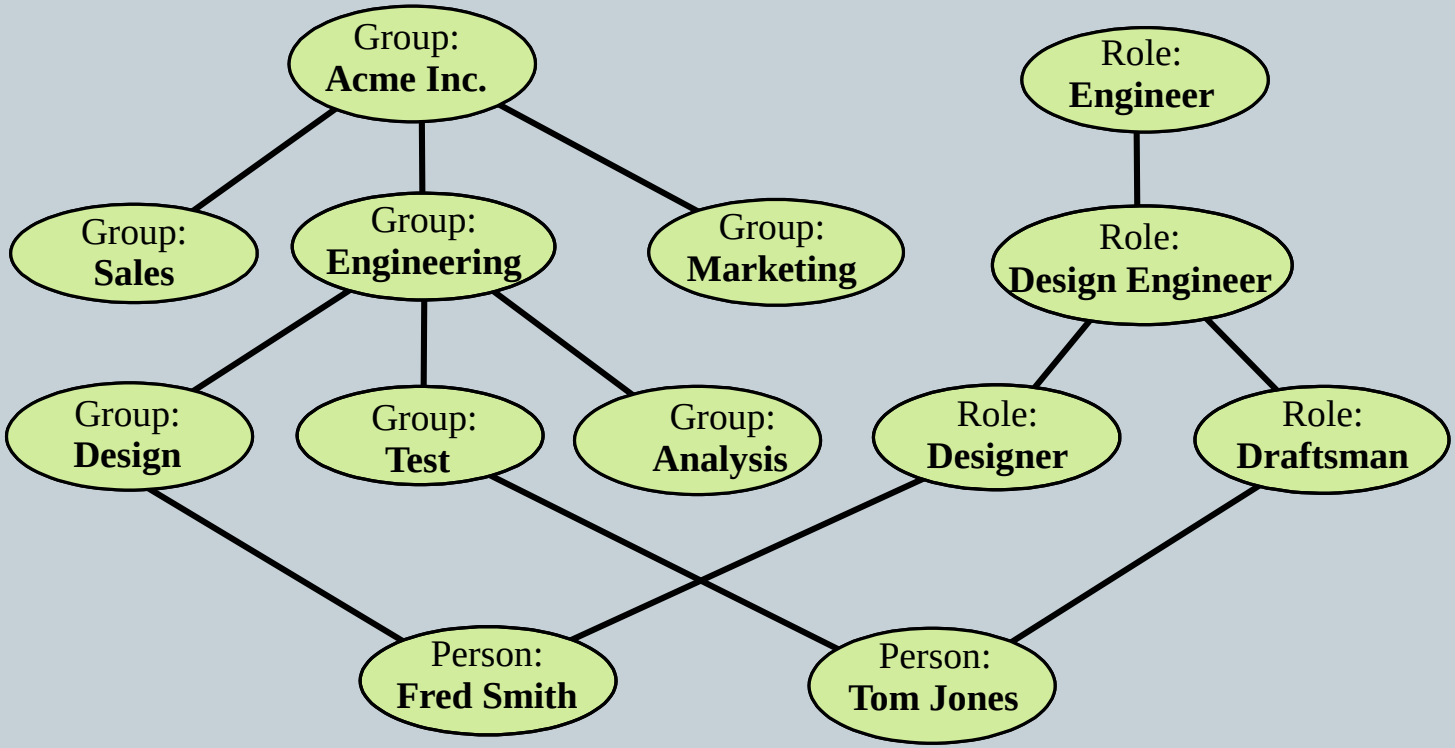
- You are identified as an Matrix user when you are defined as a person within Matrix. Your person definition enables you to own and access business objects contained within Matrix.
- Your person definition also identifies the role that you play in the organization—your job function.
- It is possible for a person to have more than one role within Matrix. For example, you might be a Manager and an Editor.
- Depending on your role, you have access in defined ways to specific types of business objects. Your role also might restrict you from accessing business objects that you do not need.

Groups of People

- **A group shares information.**

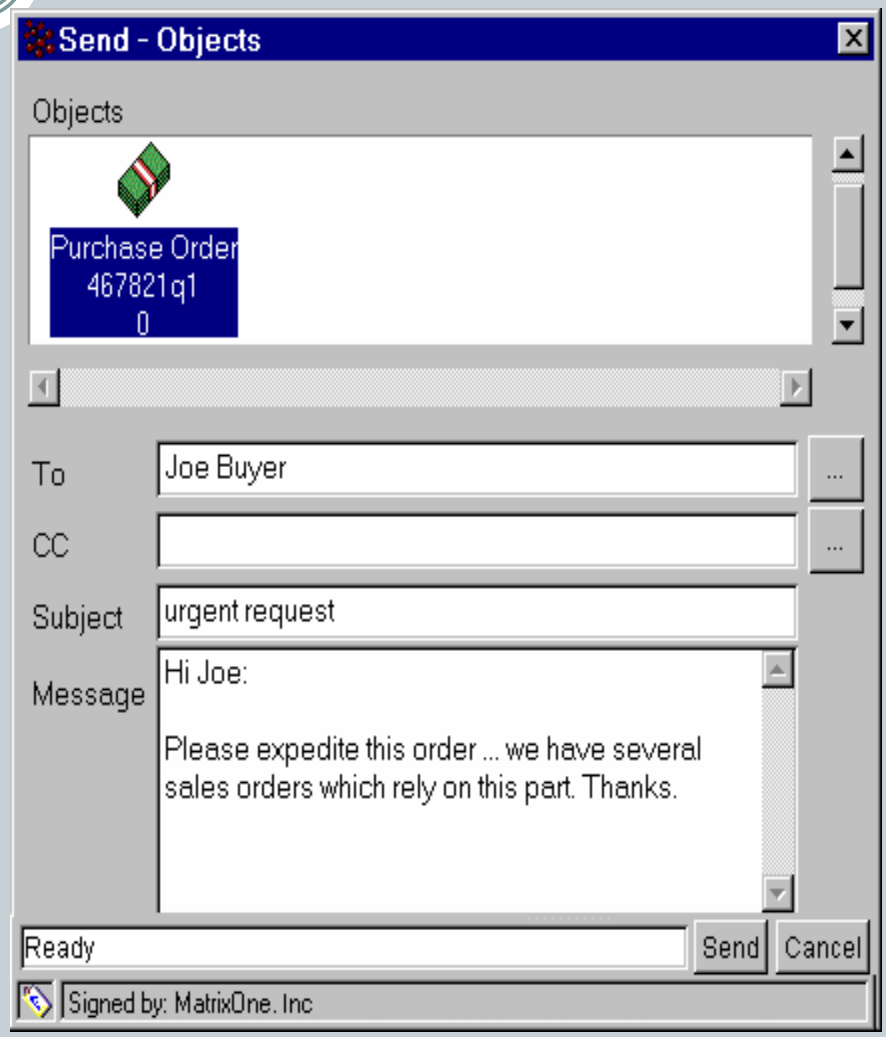
- A group identifies a set of people who are members of the same organization. They may share access to business objects for a common reason, such as a particular project or functional skills. Within a group, people of different talents and abilities may act in different roles.

Sample Organization Model



IconMail

- IconMail makes it easy to access information. It is an internal mail system.
- You can send/receive information to/from other Matrix users involved in work during a business object's lifecycle.
- IconMail is actually an object with an attached message.
- This provides access to *all of the* information for that object. IconMail makes it easy to access all related objects and associated files.



Automating Processes

- **Programmatic process implementation is much less error prone.**
- Matrix provides internal access to the programs written to drive your processes.
- Program objects, which contain executable code, may be created or imported. Programs may be explicitly executed as a method against an object, to perform any database transaction or even an external process, such as updating a report.
- Program objects are used as the building blocks for event triggers, business wizards, and workflow processes.

Automating Processes cont...

INTELIZIGN

- **Event triggers** are programs which are executed based on the occurrence of specific database events (like the modification of an attribute value, or the connection of an object).

- An event trigger may consist of any combination of three different types of programs: a **check program**, which tests that the state of the database is what the other programs expect, and can even block the event from occurring; an **override program**, which can perform additional checking or actually replace the event that triggered it; and an **action program**, which executes after the event occurs.

Automating Processes cont...

INTELIZIGN

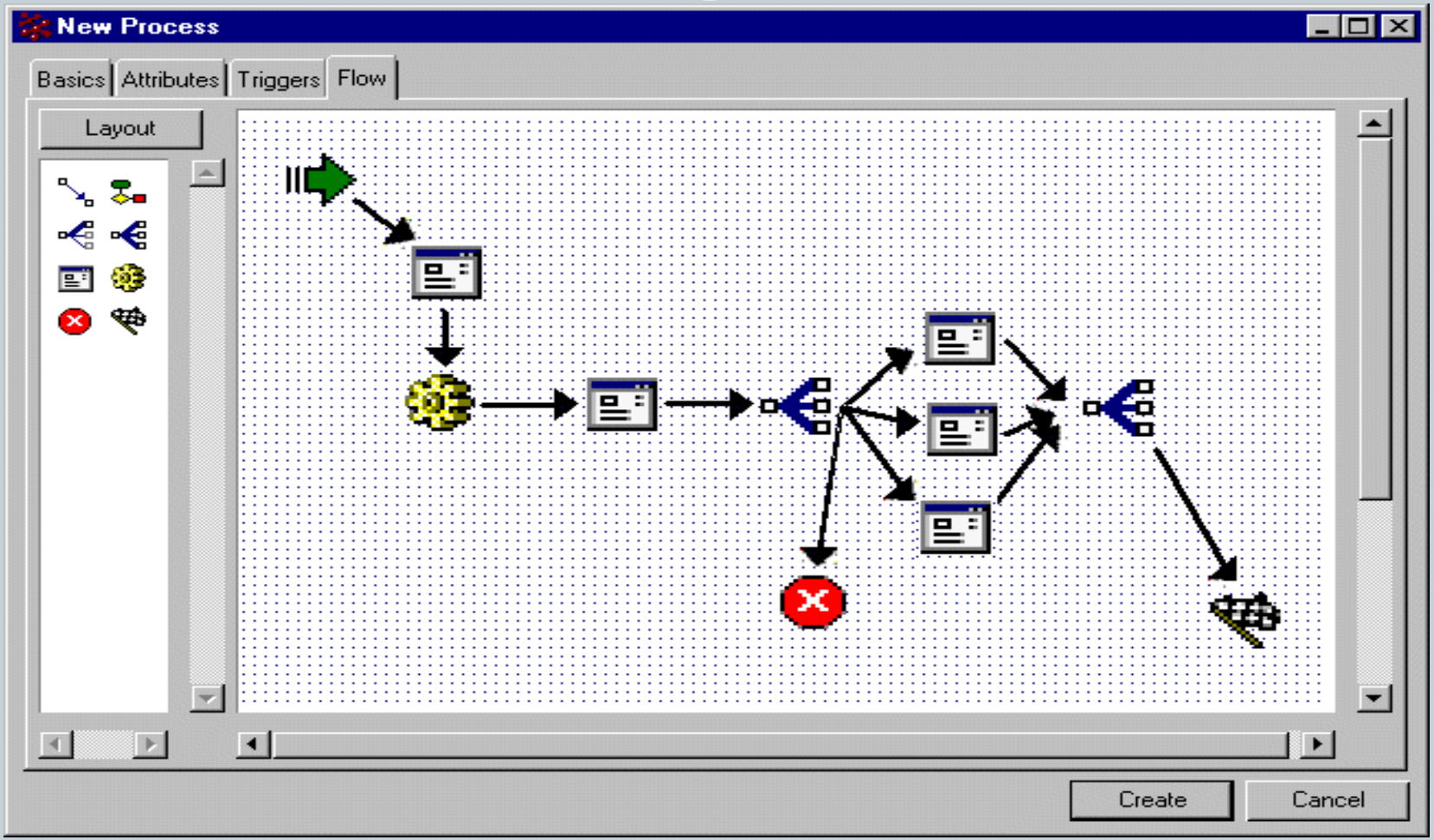
- ***Business wizards*** *may be written that automate routine processes with a user-friendly* interface similar to a windows installation routine.

- Wizards consist of several program objects that display any number of frames to provide easy, step-by-step instructions for the user.

Automating Processes cont…

- ***Workflow processes*** *can be defined that map out the activities involved in completing a* business function.
- These process definitions may be comprised of automated activities, which can be run without user intervention, and interactive activities which are sent via IconMail, to the users responsible for completing them.
- These “nodes” of the process can be linked together with “and” connectors that allow multiple activities to be performed simultaneously, or “or” connectors that provide flexibility and branching within the process.
- When one task (or set of tasks) is completed, the process automatically initiates the next task by sending a IconMail, or by running the auto activity program that is next in the process definition.

Automating Processes cont...



Vaults and Stores

INTELIZIGN

- Vault is a storage location for a group of business objects.
- A vault is a storage location residing in the underlying database. Vaults allow the designer of the Matrix database to construct one or more logical storage locations within the database.
- Store is a storage location for application files associated with business objects.
- ENOVIA offers different types of stores, each providing a different degree of control over the file.

INTELIZIGN

Thank You!