Application of XML Tools for Enterprise-Wide RBAC Implementation Tasks

Ramaswamy Chandramouli

National Institute of Standards & Technology (USA)

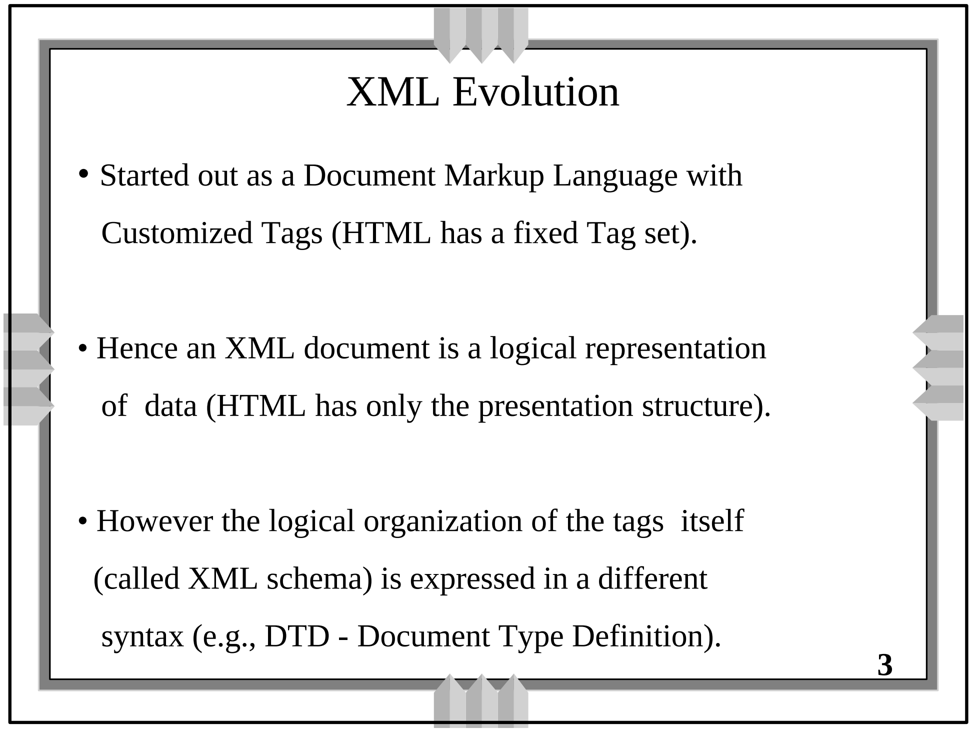
Major Topics

- Brief Introduction to XML and its APIs
- Representing an RBAC model in an XML schema
- Use of XML Tools for three Enterprise-RBAC tasks

  (1) Implementing RBAC in a DB Server

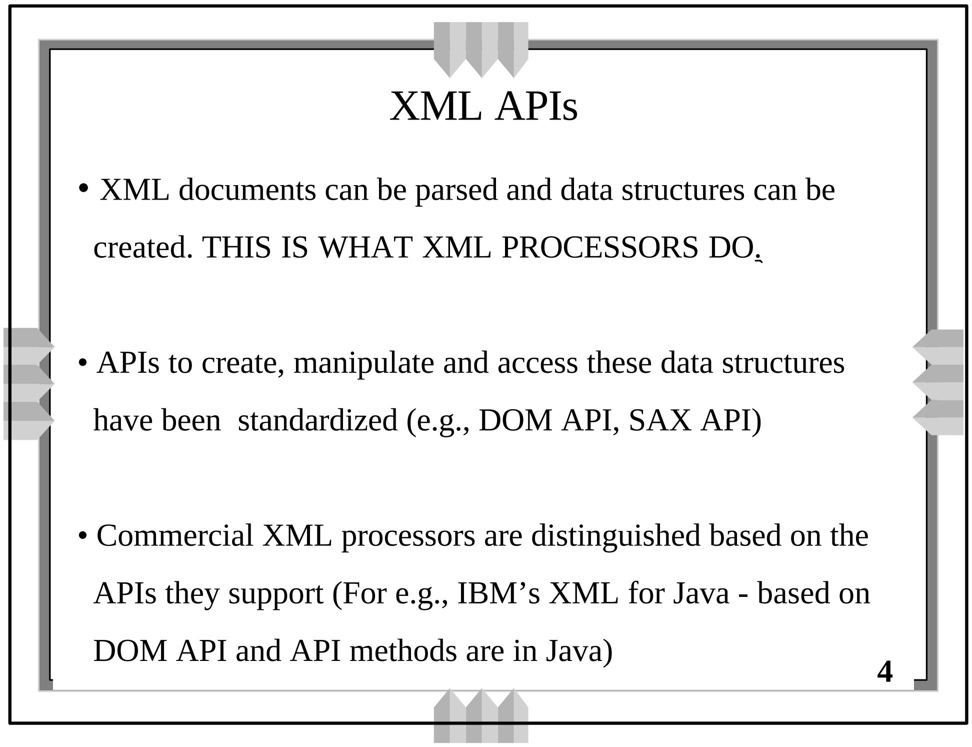
  (2) Implementing Identical RBAC Models

  (3) Implementing Access Control Service based on other Models using RBAC data



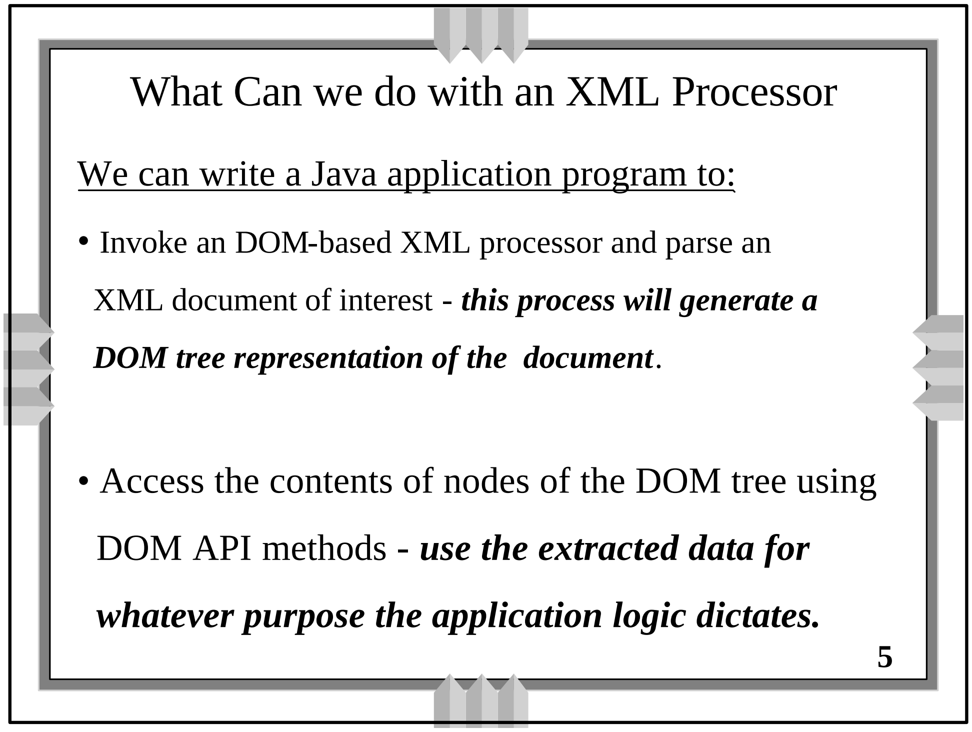
XML Evolution

- Started out as a Document Markup Language with Customized Tags (HTML has a fixed Tag set).
- Hence an XML document is a logical representation of data (HTML has only the presentation structure).
- However the logical organization of the tags itself (called XML schema) is expressed in a different syntax (e.g., DTD - Document Type Definition).



XML APIs

- XML documents can be parsed and data structures can be created. THIS IS WHAT XML PROCESSORS DO.
- APIs to create, manipulate and access these data structures have been standardized (e.g., DOM API, SAX API)
- Commercial XML processors are distinguished based on the APIs they support (For e.g., IBM's XML for Java - based on DOM API and API methods are in Java)

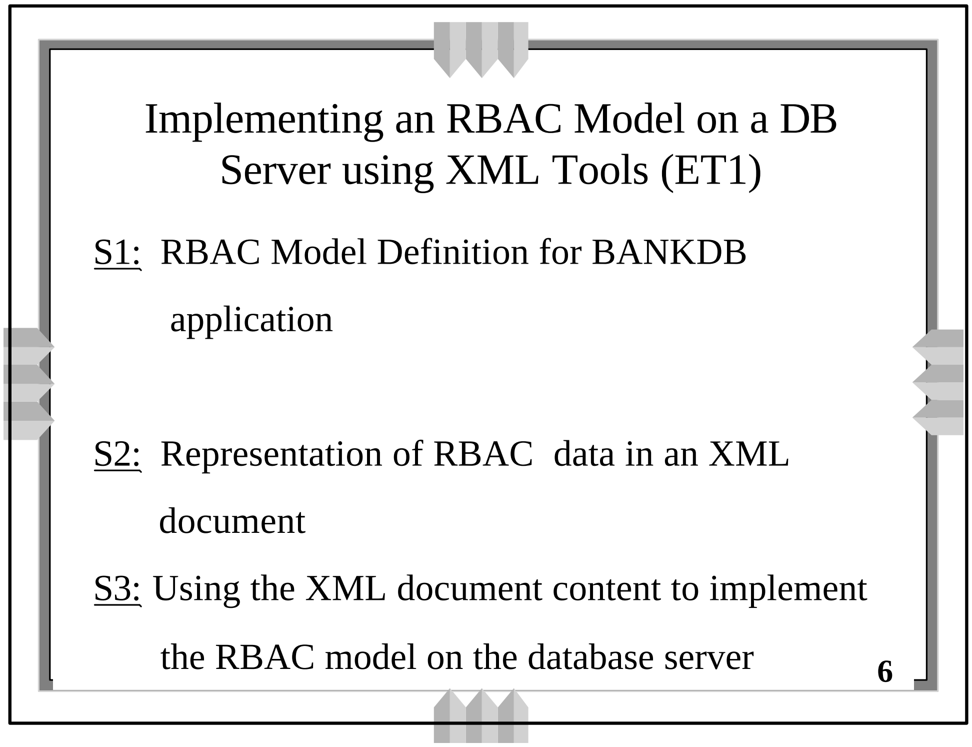


What Can we do with an XML Processor

<u>We can write a Java application program to:</u>

- Invoke an DOM-based XML processor and parse an XML document of interest - ***this process will generate a DOM tree representation of the document***.

- Access the contents of nodes of the DOM tree using DOM API methods - ***use the extracted data for whatever purpose the application logic dictates.***

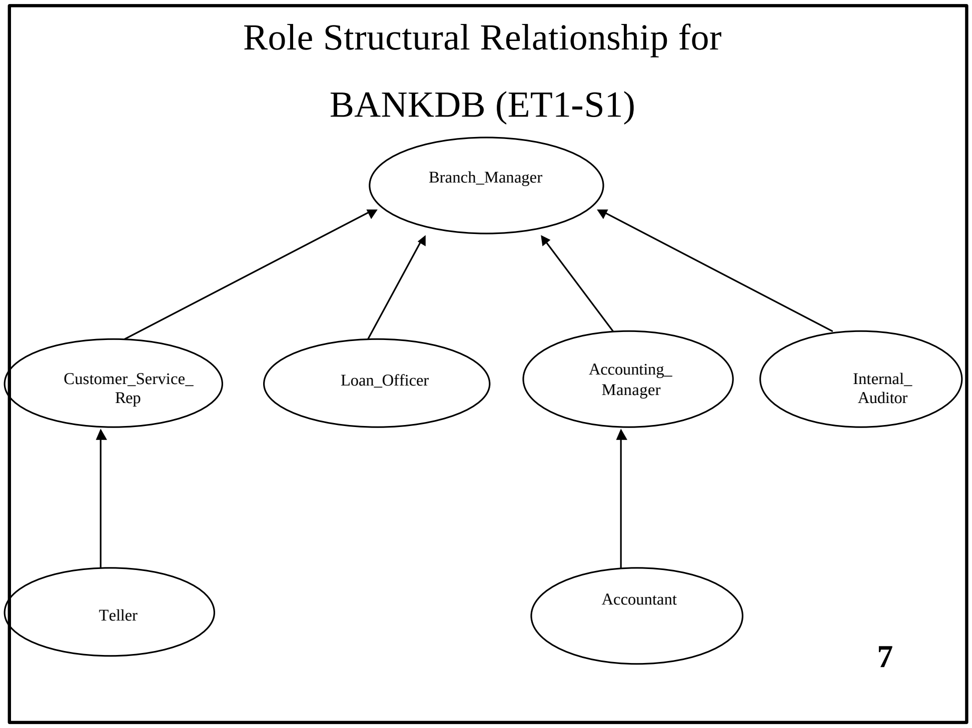


Implementing an RBAC Model on a DB Server using XML Tools (ET1)

<u>S1:</u> RBAC Model Definition for BANKDB application

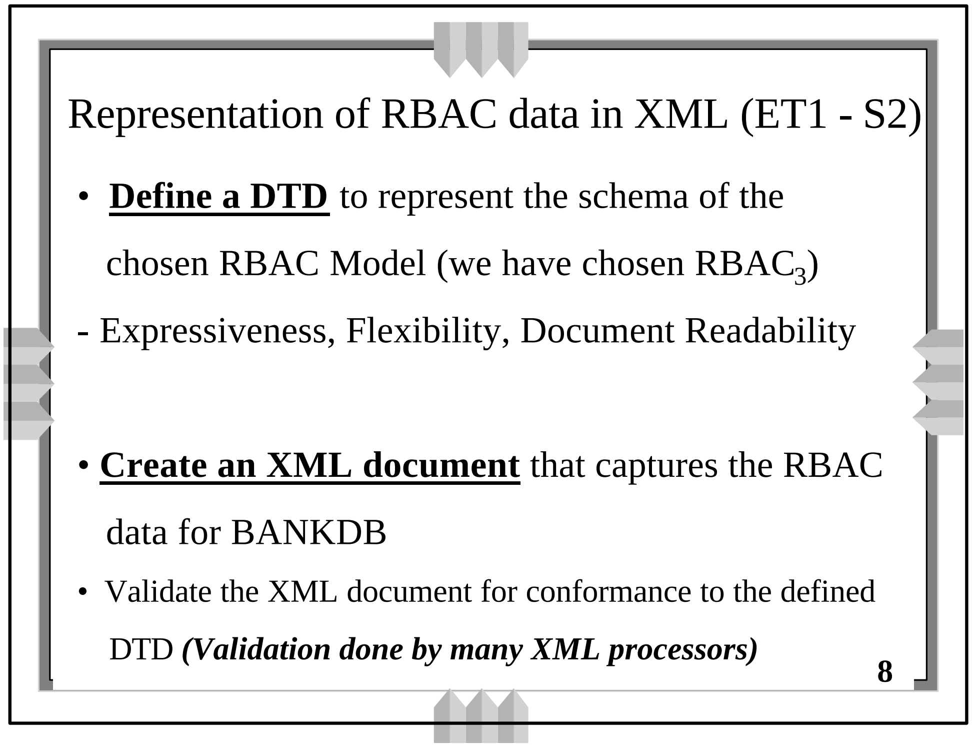
<u>S2:</u> Representation of RBAC data in an XML document

<u>S3:</u> Using the XML document content to implement the RBAC model on the database server



Role Structural Relationship for BANKDB (ET1-S1)

7



Representation of RBAC data in XML (ET1 - S2)

- **<u>Define a DTD</u>** to represent the schema of the chosen RBAC Model (we have chosen RBAC$_3$)

- Expressiveness, Flexibility, Document Readability

- **<u>Create an XML document</u>** that captures the RBAC data for BANKDB

- Validate the XML document for conformance to the defined DTD ***(Validation done by many XML processors)***

8

RBAC.dtd

(Schema for $RBAC_3$ Model)

```
<!ELEMENT Role_Graph  (Application , (role )* )*>

<!ELEMENT Application  (DB_Name , Server )>
<!ELEMENT DB_Name  (#PCDATA )>
<!ELEMENT Server  (#PCDATA )>

<!ELEMENT role (Name , Cardinality? , (Parent_Role?)* , (Child_Role?)* ,
                              (SSD_Role?)* , (DSD_Role?)* )>
<!ELEMENT Name  (#PCDATA )>
<!ELEMENT Cardinality  (#PCDATA )>
<!ELEMENT Parent_Role  (#PCDATA )>
<!ELEMENT Child_Role  (#PCDATA )>
<!ELEMENT SSD_Role  (#PCDATA )>
<!ELEMENT DSD_Role  (#PCDATA )>
```

9

BANKDB_RBAC.XML (fragment)

(RBAC data for Banking Application)

```
<?xml version="1.0" ?>
<!DOCTYPE Role_Graph SYSTEM  "RBAC.dtd">
<Role_Graph>
     <Application>
        <DB_Name>Bank Corporate Database</DB_Name>
        <Server>Solaris</Server>
     </Application>

     <role>
       <Name>Branch_Manager</Name>
       <Cardinality>1</Cardinality>
       <Child_Role>Customer_Service_Rep</Child_Role>
       <Child_Role>Loan_Officer</Child_Role>
      <Child_Role>Accounting_Manager</Child_Role>
      <Child_Role>Internal_Auditor</Child_Role>
     </role>
……………………….
</Role_Graph>
```

10

RBAC_E.dtd

(A more Expressive Schema for RBAC$_3$ Model)

```
<!ELEMENT Role_Graph  (Application, (role )*)>

<!ELEMENT Application (DB_Name,Server)>
<!ELEMENT DB_Name (#PCDATA)>
<!ELEMENT Server (#PCDATA)>

<!ELEMENT role  (Parent_Roles? , Child_Roles? , SSD_Roles? ,
DSD_Roles? )>
<!ATTLIST role  name        CDATA     #REQUIRED
                cardinality CDATA     #IMPLIED
                a-dtype     NMTOKENS  'cardinality int' >
<!ELEMENT Parent_Roles  (role )*>
<!ELEMENT Child_Roles  (role )*>
<!ELEMENT SSD_Roles  (role )*>
<!ELEMENT DSD_Roles  (role )*>
```

1
1

BANKDB_RBAC.XML (fragment)
(Conforming to RBAC_E.dtd)

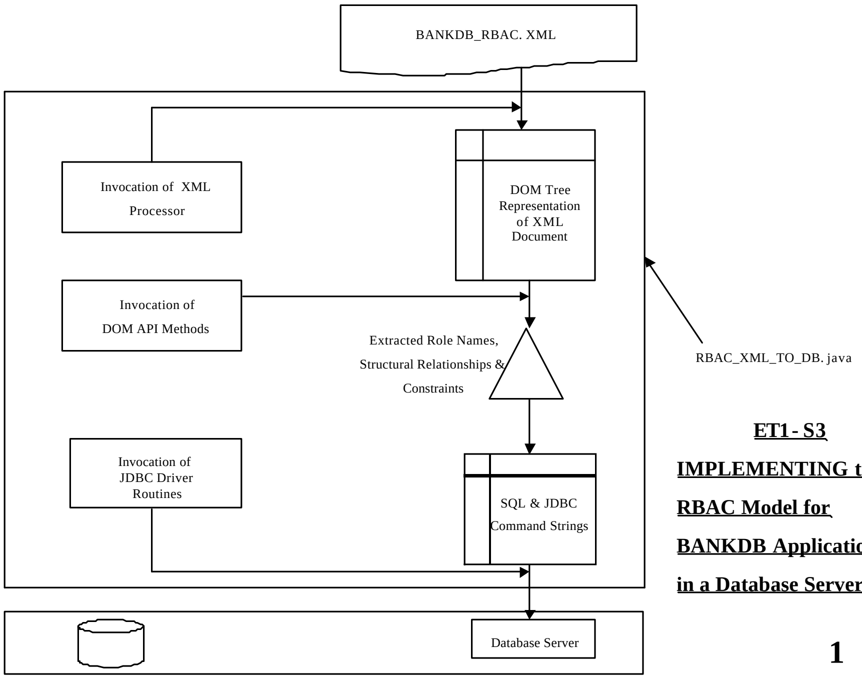
```
<?xml version="1.0" ?>
 <!DOCTYPE Role_Graph SYSTEM "RBAC_E.dtd">
 <Role_Graph>
   <Application>
     <DB_Name>Bank Corporate Database</DB_Name>
     <Server>Solaris</Server>
   </Application>

   <role name="Branch_Manager" cardinality="1">
    <Child_Roles>
        <role name="Customer_Service_Rep"/>
        <role name="Loan_Officer"/>
        <role name="Accounting_Manager"/>
        <role name="Internal_Auditor"/>
    </Child_Roles>
   </role>
```

1
2

BANKDB_RBAC.XML (fragment) (contd …) (Conforming to RBAC_E.dtd)

```
<role name="Customer_Service_Rep">
    <Parent_Roles>
        <role name="Branch_Manager"/>
    </Parent_Roles>
    <Child_Roles>
        <role name="Teller"/>
    </Child_Roles>
    <SSD_Roles>
        <role name="Accounting_Manager"/>
        <role name="Internal_Auditor"/>
    </SSD_Roles>
    <DSD_Roles>
        <role name="Loan_Officer"/>
    </DSD_Roles>
  </role>
   ……..(other role element definitions) …..
</Role_Graph>
```

1
3


BANKDB_RBAC. XML
Invocation of XML Processor
DOM Tree Representation of XML Document
Invocation of DOM API Methods
Extracted Role Names, Structural Relationships & Constraints
RBAC_XML_TO_DB. java
Invocation of JDBC Driver Routines
SQL & JDBC Command Strings
Database Server

ET1- S3

IMPLEMENTING the RBAC Model for BANKDB Application in a Database Server

1
4

Implementing an RBAC Model with Identical Data on two Servers (ET2)

S1: Retrieve access control data from DBServerA and generate an XML document that conforms to our RBAC schema RBAC.dtd

S2: Use the XML document generated above to implement RBAC model on DBServerB

(same as ET1)

15

Generating an XML document conforming to RBAC.dtd (ET2 - S1) "Validating Generation"

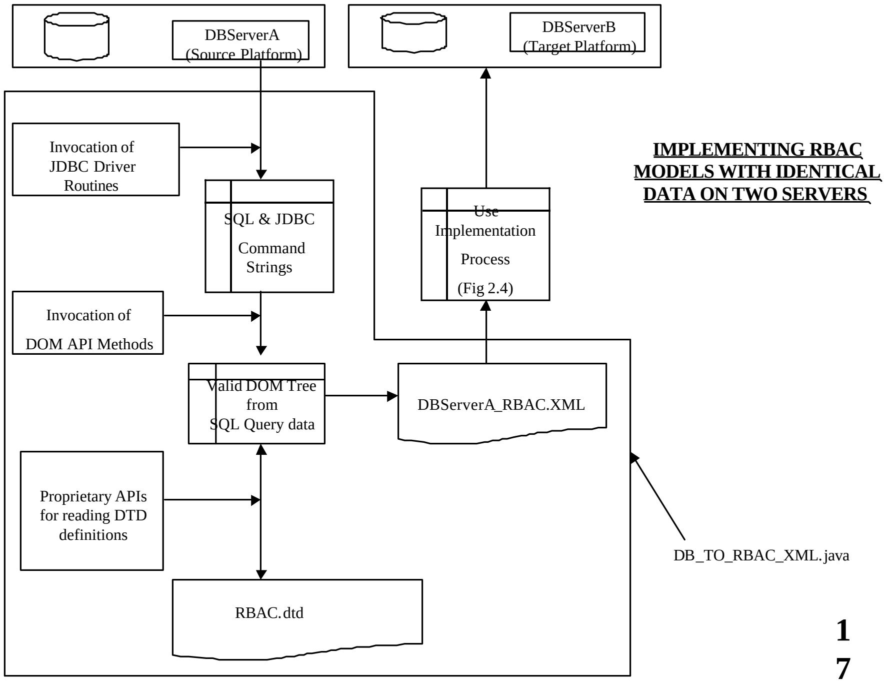
(1) Retrieve Role Names, Structural relationships and constraints by reading DBServerA.

(2) Build a DOM tree representation using that data by simultaneously reading the DTD definitions.

(3) Generate the XML document file from the DOM tree data in memory.

16

IMPLEMENTING RBAC MODELS WITH IDENTICAL DATA ON TWO SERVERS



DBServerA (Source Platform)

DBServerB (Target Platform)

Invocation of JDBC Driver Routines

SQL & JDBC Command Strings

Use Implementation Process (Fig 2.4)

Invocation of DOM API Methods

Valid DOM Tree from SQL Query data

DBServerA_RBAC.XML

Proprietary APIs for reading DTD definitions

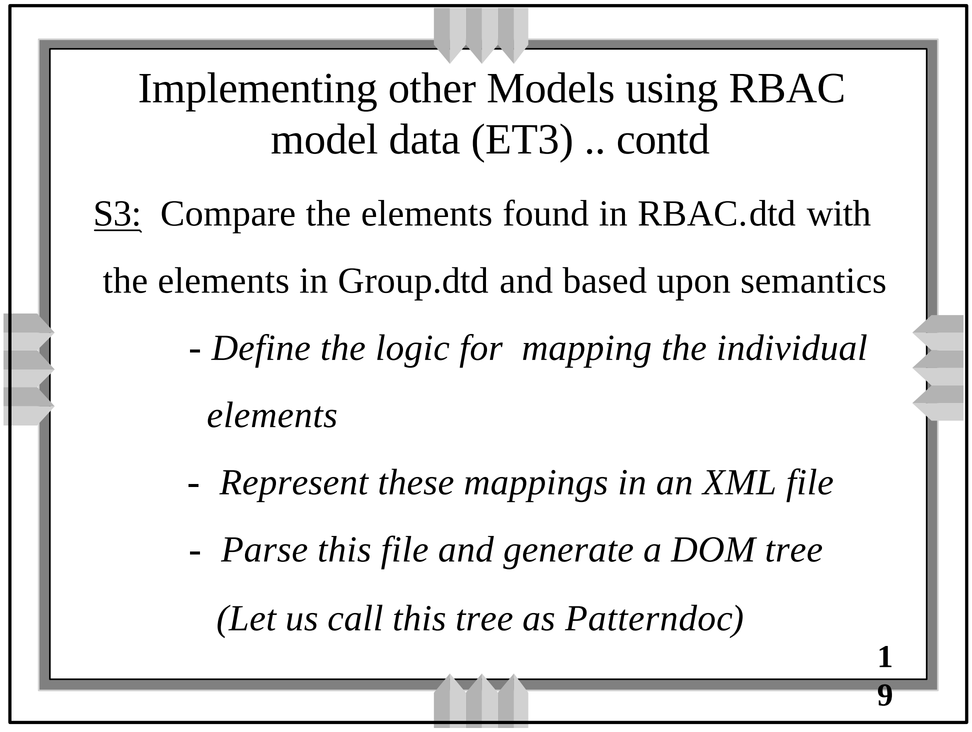
RBAC.dtd

DB_TO_RBAC_XML.java

Implementing other Models using RBAC model data (ET3)

S1: Parse the XML document containing RBAC data and generate a DOM tree representation (Let us call this tree object as *Sourcedoc*)

S2: Make a copy of this DOM tree representation This will be the initial target DOM tree (Let us call this tree object as *Targetdoc*)



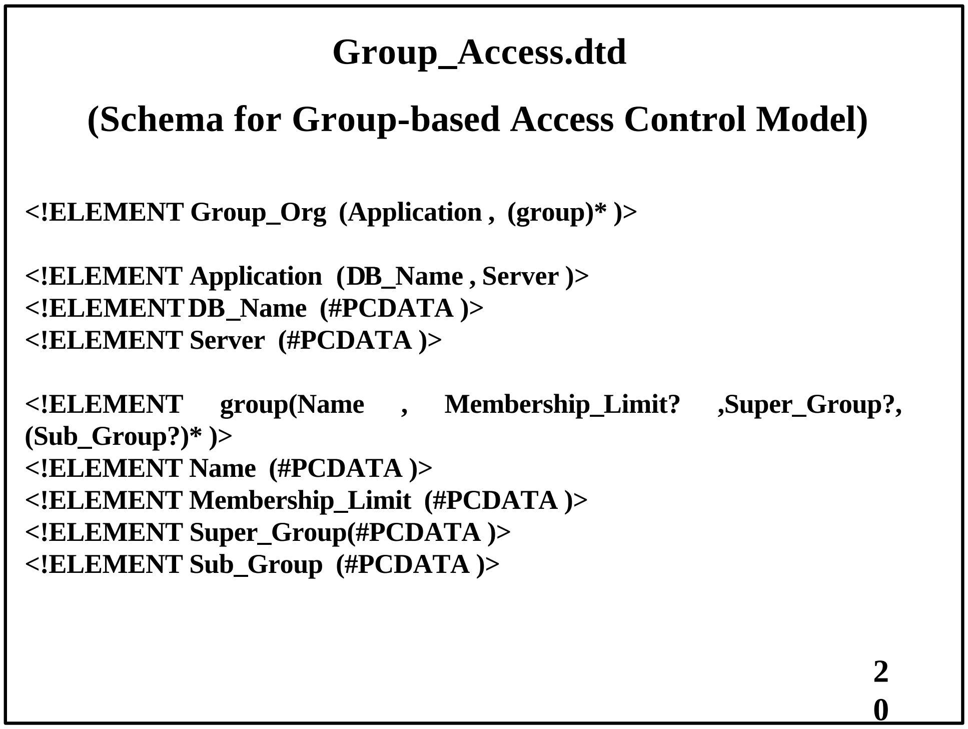
Implementing other Models using RBAC model data (ET3) .. contd

S3: Compare the elements found in RBAC.dtd with the elements in Group.dtd and based upon semantics

- *Define the logic for mapping the individual elements*
- *Represent these mappings in an XML file*
- *Parse this file and generate a DOM tree*

(Let us call this tree as Patterndoc)

19

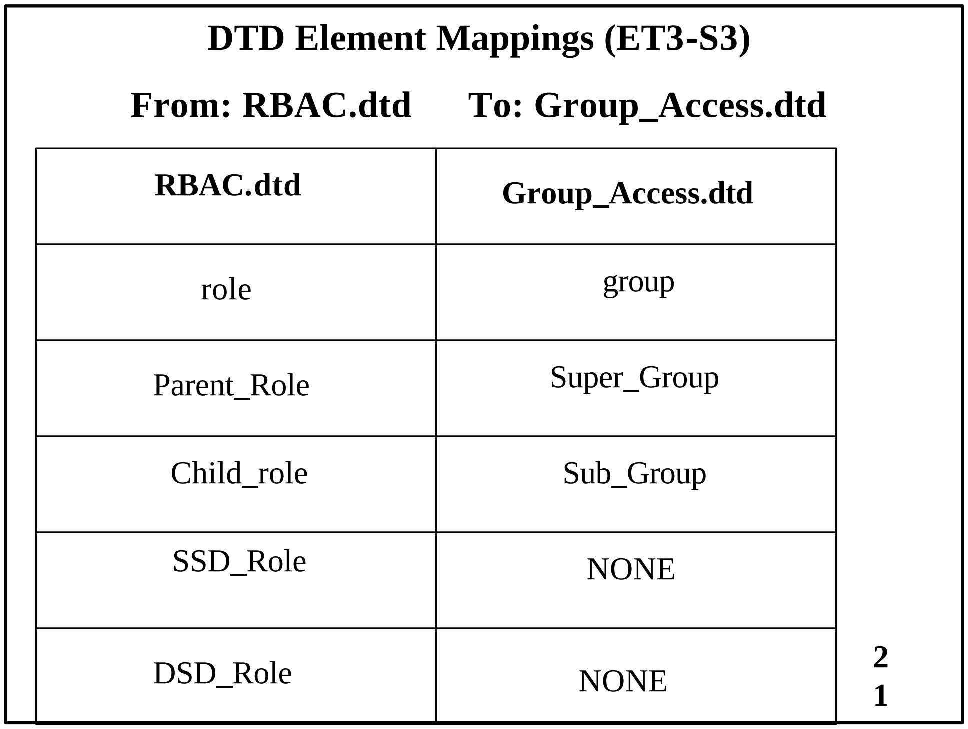


Group_Access.dtd

(Schema for Group-based Access Control Model)

```
<!ELEMENT Group_Org  (Application ,  (group)* )>

<!ELEMENT Application  (DB_Name , Server )>
<!ELEMENT DB_Name  (#PCDATA )>
<!ELEMENT Server  (#PCDATA )>

<!ELEMENT    group(Name    ,    Membership_Limit?    ,Super_Group?,
(Sub_Group?)* )>
<!ELEMENT Name  (#PCDATA )>
<!ELEMENT Membership_Limit  (#PCDATA )>
<!ELEMENT Super_Group(#PCDATA )>
<!ELEMENT Sub_Group  (#PCDATA )>
```

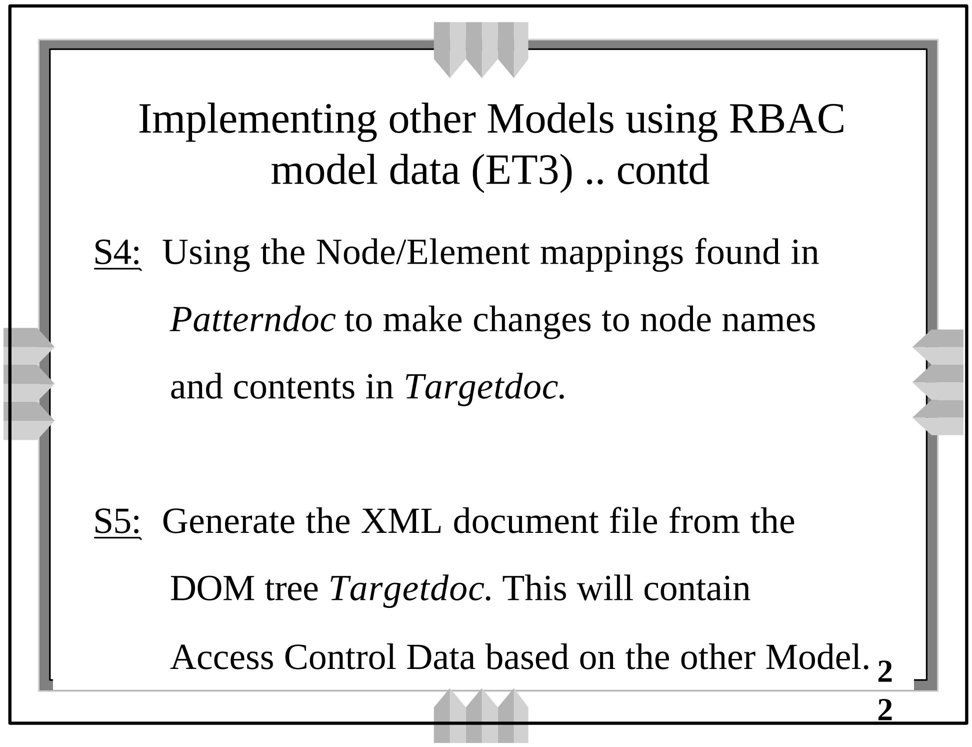
20



DTD Element Mappings (ET3-S3)

From: RBAC.dtd To: Group_Access.dtd

RBAC.dtd	**Group_Access.dtd**
role	group
Parent_Role	Super_Group
Child_role	Sub_Group
SSD_Role	NONE
DSD_Role	NONE

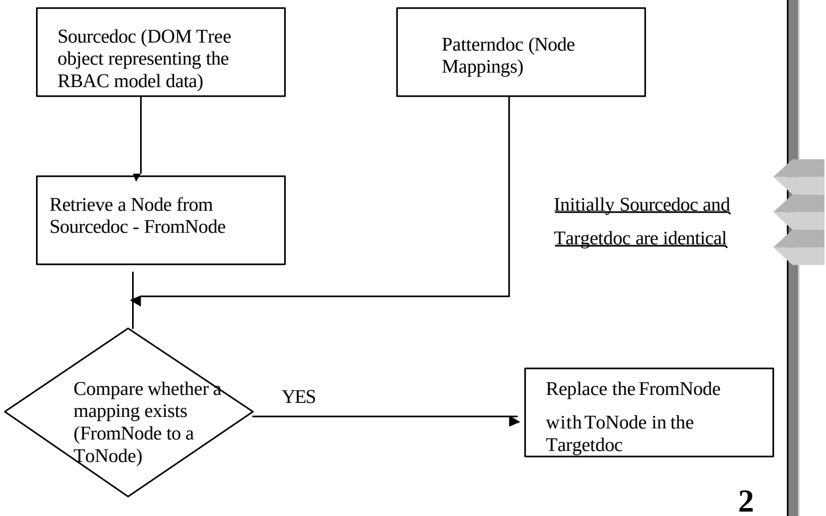


Implementing other Models using RBAC model data (ET3) .. contd

S4: Using the Node/Element mappings found in *Patterndoc* to make changes to node names and contents in *Targetdoc.*

S5: Generate the XML document file from the DOM tree *Targetdoc.* This will contain Access Control Data based on the other Model.

DOM Tree Conversion (ET3-S4)



Sourcedoc (DOM Tree object representing the RBAC model data)

Patterndoc (Node Mappings)

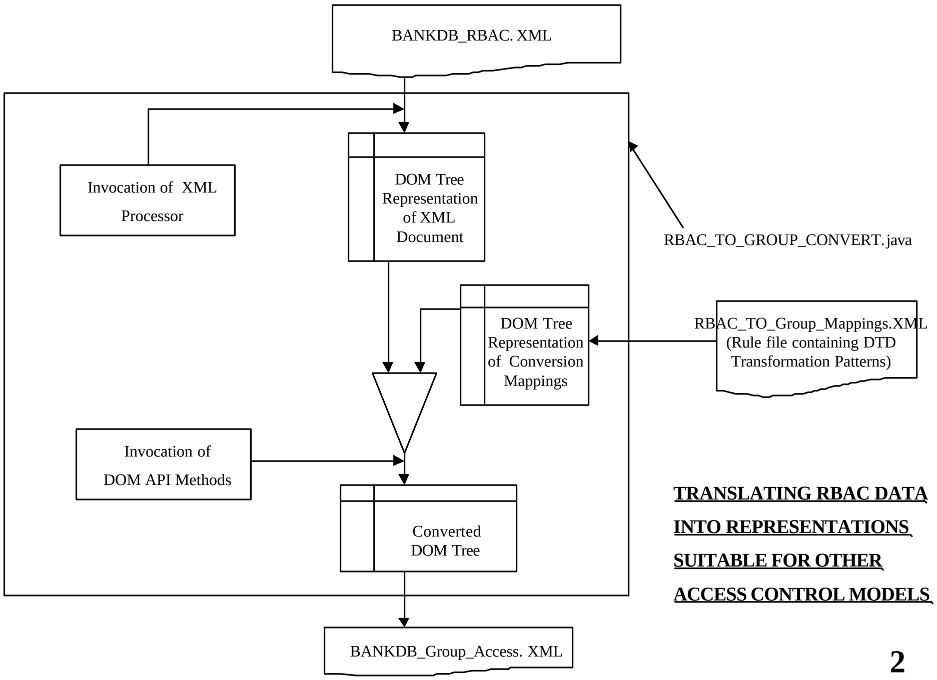
Retrieve a Node from Sourcedoc - FromNode

Compare whether a mapping exists (FromNode to a ToNode)

YES

Replace the FromNode with ToNode in the Targetdoc

Initially Sourcedoc and Targetdoc are identical



BANKDB_RBAC. XML

Invocation of XML Processor

DOM Tree Representation of XML Document

RBAC_TO_GROUP_CONVERT.java

DOM Tree Representation of Conversion Mappings

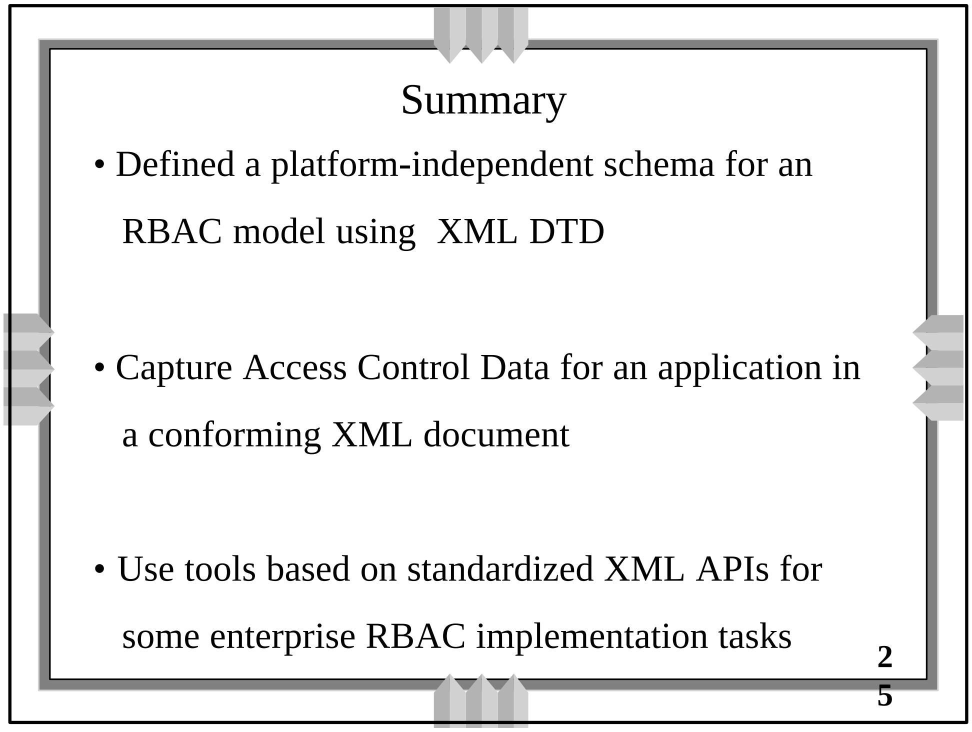
RBAC_TO_Group_Mappings.XML (Rule file containing DTD Transformation Patterns)

Invocation of DOM API Methods

Converted DOM Tree

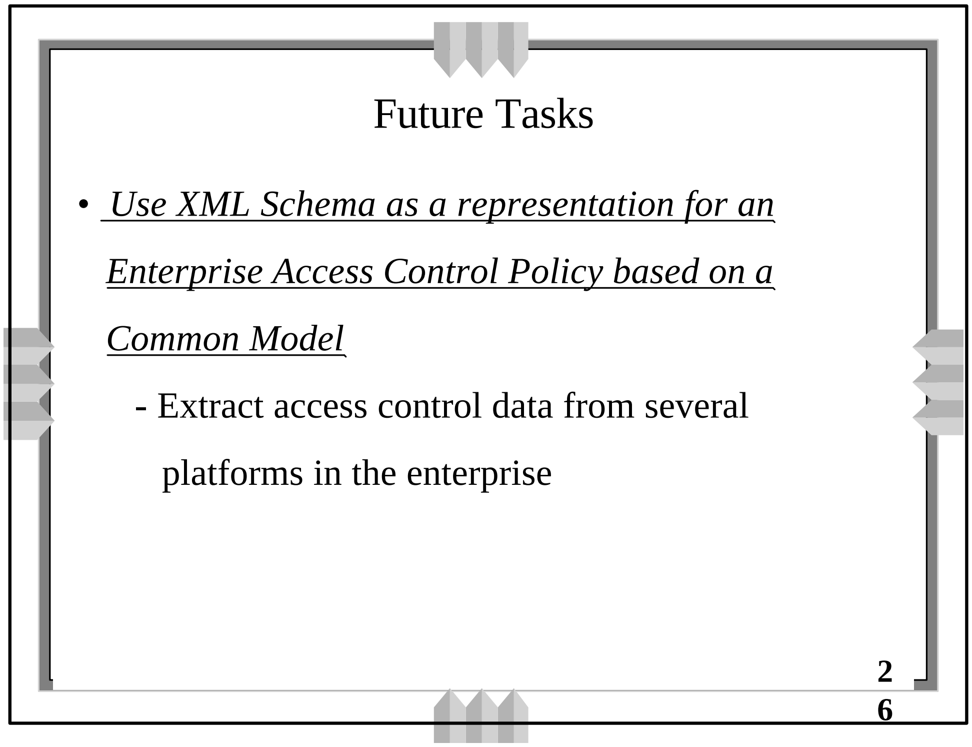
BANKDB_Group_Access. XML

TRANSLATING RBAC DATA INTO REPRESENTATIONS SUITABLE FOR OTHER ACCESS CONTROL MODELS



Summary

- Defined a platform-independent schema for an RBAC model using XML DTD
- Capture Access Control Data for an application in a conforming XML document
- Use tools based on standardized XML APIs for some enterprise RBAC implementation tasks



Future Tasks

- *Use XML Schema as a representation for an Enterprise Access Control Policy based on a Common Model*
  - Extract access control data from several platforms in the enterprise

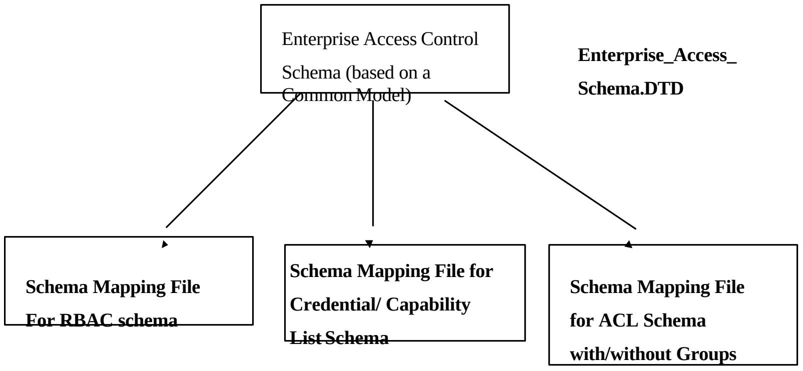
Future Tasks .. contd

- *If you have an XML schema representing the Enterprise Access Control Policy based on a Common Model*

  - You can create create appropriate schema mapping files and use them with Enterprise schema for access control implementation on various native platforms.

27

Implementing Access Control on Native Platforms from an Enterprise Schema



Enterprise Access Control Schema (based on a Common Model)

Enterprise_Access_ Schema.DTD

Schema Mapping File For RBAC schema

Schema Mapping File for Credential/ Capability List Schema

Schema Mapping File for ACL Schema with/without Groups

28