INTERGRAPH

A PROPOSED IMPLEMENTATION OF THE RASSP AUTHORIZATION MODEL USING INTERGRAPH DM2

INTERGRAPH
Intergraph Corporation
Huntsville, Alabama 35894-0001

Date: May 31, 1996

TABLE OF CONTENTS

ACRONYMS AND ABBREVIATIONS

CM	Configuration management
DM	Document Management
EF	Enterprise Framework
EFCM	Enterprise Framework Configuration Management
GUI	Graphical user interface
RASSP	Rapid Prototyping of Application-Specific Signal Processors
SQ	Saved query
TO	Transfer ownership
VL	Vault location
WL	Work location
WS	Workspace

LIST OF FIGURES

1. Authorization Model

1.1 RASSP Authorization Model

An authorization is a triplet $\{o_i, r_j, t_k\}$ where o_i is an authorization object in an authorization object hierarchy, r_j is an authorization role in an autorization role hierarchy, and t_k is an authorization type in an authorization type hierarchy.
An *authorization object* is a data object on which an authorization may be specified.
An *authorization role* is a collection of users that have the same set of authorizations on the same set of objects.
An *authorization type* is a type of operation that may be performed on a data object.
The hierarchy for each member of the triplet is organized as a directed acyclic graph described in ìThe Authorization Model for the RASSP System, Version 2î. [Martin Marietta, 1994]

The directed links between the nodes in the hierarchy represent an *implication relationship* between the nodes. The hierarchy for objects and types implies inheritance from parent node to child node. The hierarchy for roles describes a structure where the parent node authorizations are equal to, or greater then the child node. [Martin Marietta, 1994]

An authorization may be *positive*, granting an authorization, or *negative*, revoking an authorization. An explicit or an implicit positive authorization $\{o_i, r_j, t_k\}$ has to exist for an operation of the type t_k to be performed by a user belonging to role r_j on a data object blonging to the authorization object o_i. A positive (or negative) authorization specified on a node n_i in an authorization hierarchy may be overridden by a negative (or positive) authorization on a node n_j that follows n_i in the authorization hierarchy. [Martin Marietta, 1994]

The authorization object hierarchy and the authorization role hierarchy for a project may be customized by a RASSP user/systems administrator. The authorization type hierarchies, however, are not cusomizable by a RASSP user/system administrator. Defining a new authorization type will typically involve adding a new functionality to the system. [Martin Marietta, 1994]

1.2 RASSP Authorization Model in DM2

The RASSP Authorization Model can be implemented using DM2.
An authorization in DM2 performs the same function as the triplet described above, $\{o_i, r_j, t_k\}$ where o_i is an authorization object in an authorization object hierarchy, r_j is an authorization role in an autorization role hierarchy, and t_k is an authorization type in an authorization type hierarchy. DM2 extends the definition of the triplet by adding a *condition* which defines the circumstances that allow the authorization role to perform the authorization type on the authorization object.

An authorization in DM2.0 can then be described as a quadruplet $\{o_i, r_j, t_k, c_n\}$, where c_n is the *authorization condition*.

In DM2 the quadruplet is called *a message access rule*. An authorization object in a message access rule is an object *class* on which an authorization may be specified. An authorization role in a message access rule is a defined *group* or *role* for which the authorization is valid. An authorization type in a message access rule is a *message group* that defines the operations that may be performed on the object class.

A message access rule may be described in the following way: a rule grants permission for the stated group, or role, to send a message from the stated message group to the stated object class under the defined condition.

For example:

Condition:	IS_OWNER
Class:	WorkItem
MessageGrp:	UpdateGrp
Participant:	Engineering Manager

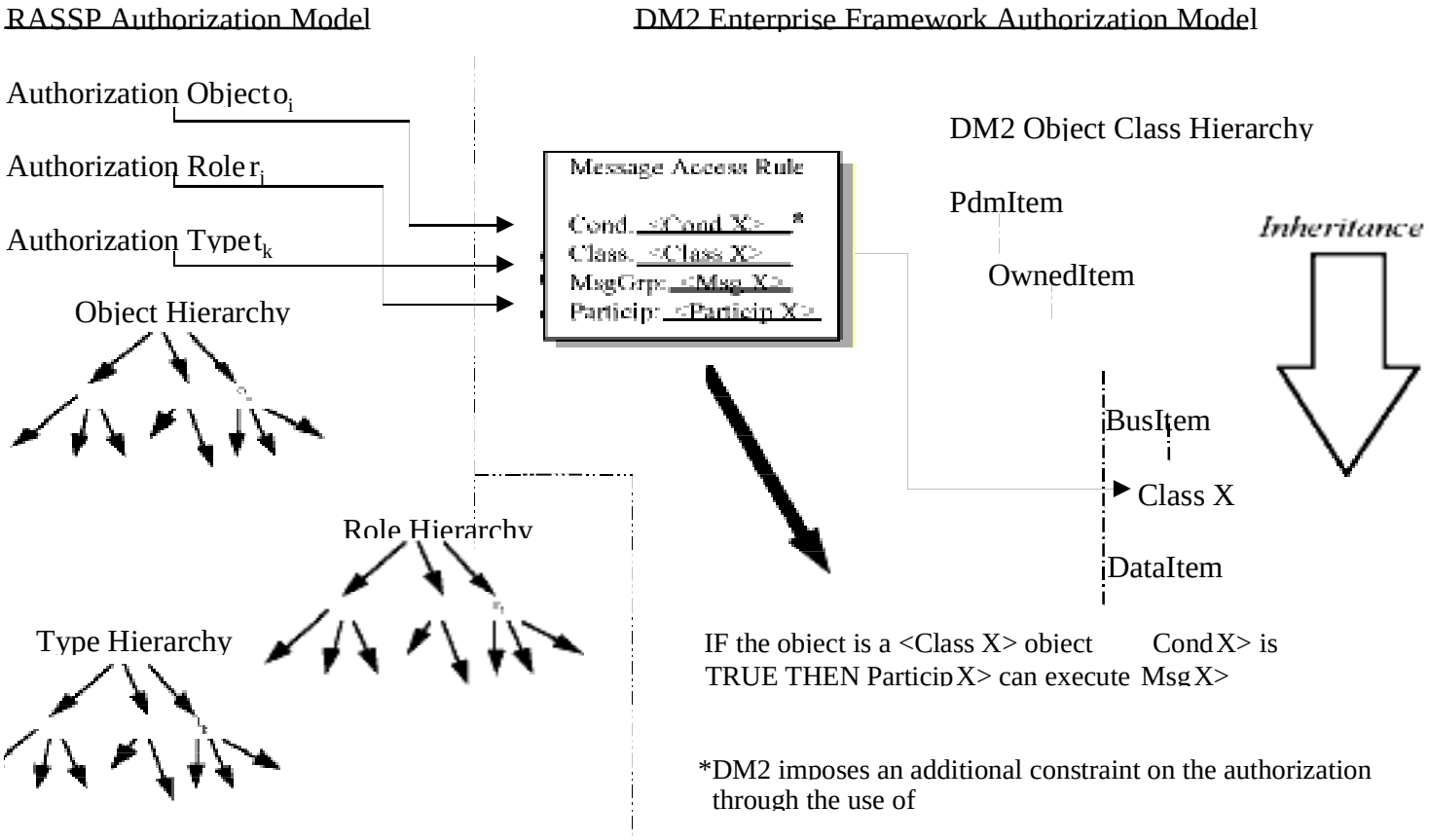
This example states that the Engineering Manager role can send any message in the UpdateGrp message group to any class of items at the WorkItem node level, or below it in the object hierarchy, under the defined condition that the user in the Engineering Manager role owns that WorkItem.

In DM2 rules are only granting, or positive, in nature. An explicit rule must exist for an action to be performed on an object class. If a rule exists, then the actions that the rule grants cannot be overridden or limited by another rule. In DM2 if two or more rules conflict or overlap, then the more permissive rule is always followed.

Inheritance in DM2 is found only in the object class hierarchy. A message group defines only the messages that are availabe through that rule. Likewise, all authorizations granted to a participant must be explicitly defined for that participant. In other words, authorizations for one participant (user, role, or group) who has authority over other participants, are not necessarily a superset of authorizations defined for those under his authority. Similarily, those participants under anothers authority do not necessarily have a subset of authorizations defined for them. (Asterisks ë*í may be used as wildcards in defining object class, participant, or message group.)

In the RASSP Enterprise Framework (EF) implementation of the DM2 authorization, the DM2admin user will be the only user who may create or update rules. The DM2admin will also be responsible for creating new groups or roles, new EF projects, and new object classes on the system.

DM2 Enterprise Framework Implementation of RASSP Authorization



2 Mechanisms to support the Enterprise Framework Authorization Model

2.1 Mechanisms to Manipulate the Authorization Object Hierarchy

2.1.1 Creating an Authorization Object

In DM2 an object class can be created and inserted into the Object Class Hierarchy at any level except at the hierarchy root node. Adding a new object class involves many steps that are complex and will alter the Customized Environment where DM2 is installed. Updates made to the Customized Environment of DM2 should be handled by one person, namely the DM2 System Adminstrator.

2.1.2 Deleting an Autorization Object

In DM2 an object class can be deleted from the Object Class Hierarchy at any level except at the hierarchy root node. Deleting an object class involves many steps that are complex and will alter the Customized Environment where DM2 is installed. Updates made to the Customized Environment of DM2 should be handled by one person, namely the DM2 System Adminstrator.

2.1.3 Adding a child to an Authorization Object

Adding a child to an authorization object is similar to creating a new authorization object. Since an object in DM2 may appear in only one node in the authorization object hierarchy it may be moved, but not copied to another node.

2.1.4 Associating Data files with Authorization Object

In DM2 an object class may be instantiated any number of times to create many specific objects of the same type. Once an object is created and data associated with it, that object may be handed over to configuration management.

2.1.5 Retrieving an Authorization Object

In DM2 a query can be made for a specific object class which will return a list of objects of that class, plus any other class below it in the object hierarchy which is currently defined in the scope of the database.

2.1.6 Retrieving the Children of an Authorizaion Object

In DM2 a query can be made for a specific object class which will return a list of objects of that class, plus any other class below it in the object hierarchy which is currently defined in the scope of the database.

2.2 Mechanisms to Manipulate the Authorization Role Hierarchy

2.2.1 Creating an Authorization Role

In DM2, authorization roles, whether they are defined roles or groups, do not have explicitly defined parent-child relationships with other authorization roles. Such a relationship is expressed through the rules that are created for each specific authorization role. One role may have authority over another, which could be interpreted as a parent-child relationship. Creating an authorization role should be the responsibilty of a specific user, thus controlling the number of roles that are created.

2.2.2 Deleting an Authorization Role

In DM2, deleting a role has no affect on any of its implied children. All users must be removed from the role before it is deleted and the associated rules removed also. Deleting an authorization role should be the responsibilty of a specific user, thus controlling which roles are deleted and when.

2.2.3 Adding a Child to an Authorization Role

In DM2, a parent-child relationship between authorization roles is implied through the rules defined for the specific roles. The rules for each role define the actions available to that role. Some roles may have be granted more authority then others, essentially giving one role authority over an other. This implies a parent-child relationship.

2.2.4 Associating Users with Authorization Roles

DM2 provides an easy ìdrag-and-dropî GUI interface to associate users with authorization roles. By simply choosing a user and dragging that user into the desired role or group creates an association between that user an the role. All the rules defined for the role are defined for the users in that role.

2.2.5 Retrieving an Authorization Role

In DM2 a query can be made for an authorization role. This query will return the authorization being queried if it exists in the database. Since there are no explicitly defined relationships between roles, the need for a root node, which defines the tree to search, need not be given to the query.

2.2.6 Retrieving the Children of an Authorization Role

In DM2 a query cannot be defined to request the children of a specific role. A query for a specific role may be made, or a query for all the roles in the database may be made. But given an authorization role, there is no defining relationship between it and any implied child node.

2.3 Mechanisms to Manipulate the Authorization type hierarchy

2.3.1 Retrieving an Authorization Type

In DM2 a query can be defined for an authorization type. This query will return the authorization type if it exists in the database. Authorization types in DM2 can be access as one specifc message, a message grouping, or all the messages in that exist.

2.3.2 Retrieving the Children of a node in the Authorization Type Hierarchy

In DM2 a query cannot be defined to request the children of a specific authorization type. A query for a specific type may be made, or a query for all the authorization types in the database may be made. But given an authorization type, there is no defining relationship between it and any child type.

2.4 Mechanisms to Grant Authorizations

Since DM2 is granting by design, new authorizations cannot be created to revoke, or limit, existing authorizations.

2.4.1 Granting Authorizations

In DM2 authorizations can be defined using Message Access Rules. Given an authorization object <Class W>, and authorization role <Participant X>, and authorization type <Message Group Y>, and a condition under which the authorization is granted <Condition Z>, a Message Access Rule can be written to grant authorization as described below--

IF the object is a <Class W> object, AND <Condition Z> is TRUE THEN
    <Participant X> can execute <Message Group Y> commands.

2.4.2 Revoking Authorizations

In DM2 the only way to revoke an authorization is to remove that authorization, or message access rule, from the database. This will remove the granting authorization.

References

[Cattell, 1991] Cattell, R.G.G., Object Data Management, Massachusetts: Addison Wesley, 1991.
[Martin Marietta, 1994] Martin Marietta RASSP Team, Martin Marietta Laboratories, ìThe Configuration Management Model for the RASSP Systemî, Moorestown, New Jersey, 1994.