

# Application of URNs to Authorization and some other exotic uses

Victoriano Giralt

Central Computing Facility  
University of Málaga

Zagreb, January 31st, 2006



# Contents

- 1 Use cases of URNs in Entitlements
  - Expenses Authorization Control
  - Mobile phone number usage control
- 2 URNs for adding hierarchies
  - Object classification
  - Classifications use cases
- 3 URN handling problems
- 4 The future



# Outline

- 1 Use cases of URNs in Entitlements
  - Expenses Authorization Control
  - Mobile phone number usage control
- 2 URNs for adding hierarchies
  - Object classification
  - Classifications use cases
- 3 URN handling problems
- 4 The future



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
**entitlement**:appAccess:SolicitudGasto:*LEVEL*

Asigns access rights to the designated application:

- **Function**
- Usage
- Advantages

**entitlement**

the URN describes a right for a user or role



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:LEVEL

Asigns access rights to the designated application:

- **Function**
- Usage
- Advantages

## applAccess

kind of right, access to an application in this case.



# Expenses Authorization Control

(state: production)

```
irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:appAccess:SolicitudGasto:LEVEL
```

Asigns access rights to the designated application:

- **Function**
- Usage
- Advantages

**SolicitudGasto**

application the right is granted on.



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:*LEVEL*

Asigns access rights to the designated application:

- **Function**
- Usage
- Advantages

## *LEVEL*

granted access level, application specific:  
RUG, ROU, RGE



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- **Usage**
- Advantages

## LDAP search

The application does a standard directory search to find out if the user that has been authenticated has the right to use it and the access level that has been granted to her.





# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- **Usage**
- Advantages

## Query via web service

The application queries a web service with user and application identifier as inputs and obtains the access level or the absence of the right to use.



# Expenses Authorization Control

(state: production)

```
irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:LEVEL
```

- Function
- Usage
- Advantages

## Future: PAPI

We are preparing a migration path for our applications, such that, once the user has been authenticated by PAPI, the assertion will carry application specific AuthZ information derived from the entitlements stored in the user's entry in the directory.



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage
- **Advantages**

## Unique authorization point

All of an object's authorizations, both explicit and implicit, are centrally kept in a directory entry.



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage
- **Advantages**

## A sole authorization model

URN allow us to express all authorization in a common form, with application specific semantics.



# Expenses Authorization Control

(state: production)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:applAccess:SolicitudGasto:LEVEL

- Function
- Usage
- Advantages

Agent-Function-Qualifier

*Who can do What on Which object*



# Mobile phone number usage control

(a more complicated case)

```
irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile:VALUE
```

## ● User to application

- The problem
- Examples
- BUT ...

### Personal management of permissions

The user grants permissions on his data to applications.

May we use *entitlements*?

Is it unorthodox?

A new `irisUserPrivateAttribute`?



# Mobile phone number usage control

(a more complicated case)

```
irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile:VALUE
```

- User to application
- **The problem**
- Examples
- BUT ...

## Attribute access control

Different applications may want to use an attribute, the user can decide if she permits the use of the attribute or not, for the ends of each of them.



# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile:VALUE

- User to application
- The problem
- **Examples**
- BUT ...

## mobile

This attribute can be used for several applications, like:

- + changing forgotten passwords
- + sending marks
- + sending notices





# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile: **VALUE**

- User to application
- The problem
- **Examples**
- BUT ...

**VALUE = passwordChange**

The user allows the SMS gateway to use his mobile phone number for the password change function. From another point of view, the user authorizes the use of his mobile phone number for starting a password change.



# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile: **VALUE**

- User to application
- The problem
- **Examples**
- BUT ...

**VALUE = marks**

The user authorizes the use of her mobile phone number for accessing her marks and for sending them to such number.



# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile: **VALUE**

- User to application
- The problem
- **Examples**
- BUT ...

**VALUE = maySpam**

The user allows the use of his mobile phone number for sending notices from the University.



# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile: *VALUE*

- User to application
- The problem
- Examples
- **BUT ...**

## irisUserEntitlement

Holds permissions granted to the object (user).



# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile: *VALUE*

- User to application
- The problem
- Examples
- **BUT ...**

## irisUserPrivateAttribute

Holds access permissions that the object  
(user) grants on her attributes.



# Mobile phone number usage control

(a more complicated case)

irisUserEntitlement = urn:mace:rediris.es:uma.es:  
entitlement:attrAccess:mobile: *VALUE*

- User to application
- The problem
- Examples
- **BUT ...**

A new irisUserPrivateAttribute?

Should we migrate to a URN based model? Would it work?



# Outline

- 1 Use cases of URNs in Entitlements
  - Expenses Authorization Control
  - Mobile phone number usage control
- 2 URNs for adding hierarchies
  - Object classification
  - Classifications use cases
- 3 URN handling problems
- 4 The future



# Object classification

## of hierarchies and sparse trees

We have had the opportunity of designing our enterprise directory from scratch very recently, learning from many others' successes and mistakes.

- **Shallow trees**
- Hierarchies
- Virtual views

### Few one level branches

Real world usage has shown us that, storing objects inside a flat structure, with a few branches for storing similar object types, just one level beneath the organization root, is more practical, requiring fewer administration.





# Object classification

## of hierarchies and sparse trees

We have had the opportunity of designing our enterprise directory from scratch very recently, learning from many others' successes and mistakes.

- Shallow trees
- **Hierarchies**
- Virtual views

### Organizations DO have hierarchies

Regardless of internal directory structure, there is an organizational hierarchy for many of the objects stored in it.

Therefore, there is a need for *presenting* entries in a hierarchical form.



# Object classification

## of hierarchies and sparse trees

We have had the opportunity of designing our enterprise directory from scratch very recently, learning from many others' successes and mistakes.

- Shallow trees
- Hierarchies
- **Virtual views**

### Several hierarchies for the same set

Often, the same type of objects has to be presented with a different structure. This is difficult to solve with traditional approaches. It is quite easy to do using classification codes stored as URNs.



# Use cases for classifications

different views of the same tree

Classifications describe hierarchies using variable length codes, adding a new code element for each new level. Code elements size is fixed for each classification.

- **Classifications branch**
- Classification root
- Classification entry

```
dn: dc=classif,dc=uma,dc=es
```

```
objectClass: top  
objectClass: organizationalUnit  
objectClass: dcObject  
dc: classif  
ou: classif
```



# Use cases for classifications

different views of the same tree

Classifications describe hierarchies using variable length codes, adding a new code element for each new level. Code elements size is fixed for each classification.

- Classifications branch
- **Classification root**
- Classification entry

```
dn: dc=umaLoc-1.0,dc=classif,  
dc=uma,dc=es
```

```
objectClass: top  
objectClass: organizationalUnit  
objectClass: dcObject  
dc: umaLoc-1.0  
ou: umaLoc-1.0
```



# Use cases for classifications

different views of the same tree

Classifications describe hierarchies using variable length codes, adding a new code element for each new level. Code elements size is fixed for each classification.

- Classifications branch
- Classification root
- **Classification entry**

```
dn: copaCode=a01b01c01d03e05,  
dc=umaLoc-1.0,dc=classif,  
dc=uma,dc=es
```

```
objectClass: top  
objectClass: copaArea  
copaName: Director's office  
copaCode: a01b01c01d03e05  
description: The office of the director of  
the Polytechnic School
```

# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
**umaLoc**:1.0:a01b01c01d03e05

- **Classification**
- Version
- Code

## Name of the classification

This allows to know which classification the code belongs to.

umaLoc: Geographical location

umaOrg: Organizational roles

levels have fewer nodes



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
**umaOrg**:1.0:a1b1c1d1e1

- **Classification**

- Version
- Code

## Name of the classification

This allows to know which classification the code belongs to.

umaLoc: Geographical location

umaOrg: Organizational roles

levels have fewer nodes



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:a01b01c01d03e05

- Classification
- **Version**
- Code

## Classification version

The versioning information is important in order to know that object entries are up to date when presenting them using one precise classification.





# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:**a01b01c01d03e05**

- Classification
- Version
- **Code**

## Entry's classification code

The code places the entry in an exact location in the University premisses.



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:a01b01c01d03e05

- Classification
- Version
- **Code**

## Entry's classification code

a01:  
Campus "El Ejido"



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:**a01b01**c01d03e05

- Classification
- Version
- **Code**

## Entry's classification code

a01b01:  
Campus "El Ejido"  
Polytechnic School



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:**a01b01c01**d03e05

- Classification
- Version
- **Code**

## Entry's classification code

a01b01c01:

Campus "El Ejido"

Polytechnic School

Administration Building



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:a01b01c01d03e05

- Classification
- Version
- **Code**

## Entry's classification code

a01b01c01d03:

Campus "El Ejido"

Polytechnic School

Administration Building

Third floor



# Use cases for classifications

different views of the same tree

A person's entry can hold as many classification codes as needed in order to place her in different hierarchies.

irisClassifCode = urn:mace:rediris.es:uma.es:classif:  
umaLoc:1.0:**a01b01c01d03e05**

- Classification
- Version
- **Code**

## Entry's classification code

a01b01c01d03e05:  
Campus "El Ejido"  
Polytechnic School  
Administration Building  
Third floor  
Director's office

# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- **Main entry**
- Virtual view

## Available views defined at root

The root entry holds an attribute with virtual views that can be overlaid on the directory.

```
copaMainNav: dc=umaLoc-vv1,  
              dc=vvviews,dc=uma,dc=es  
copaMainNav: dc=umaOrg-vv1,  
              dc=vvviews,dc=uma,dc=es
```



# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

Virtual view entries have attributes that hold all information that a program needs for doing searches that present the objects according to the desired hierarchy.





# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

Search base for retrieving a classification.

copaClassifBase:

dc=umaLoc-1.0,dc=classif,

dc=uma,dc=es



# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

URN prefix of the classification codes.

copaPrefix:

urn:mace:rediris.es:uma.es:

classif:umaLoc:1.0



# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

Object class for the classification elements.

```
copaAreaObjectClassName:  
  copaArea
```



# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

Attribute of the classification entries that holds the codes.

copaCodeAttr:  
copaCode

Example value:

a01b01c01d03e05

# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

Attribute of the classification entries that holds the printable name of the code.

copaPrintAttr:

copaName

Example value:

Director's office

# Use cases for classifications

different views of the same tree

Once the persons' entries hold various classification codes it is possible to overlay different hierarchical views over an otherwise shallow directory. This view can be navigated using data stored in the directory.

- Main entry
- **Virtual view**

## Information for presenting the view

Attribute of the object entries that holds the classification codes.

```
copaCodeResourceAttr:  
  irisClassifCode
```



# Outline

- 1 Use cases of URNs in Entitlements
  - Expenses Authorization Control
  - Mobile phone number usage control
- 2 URNs for adding hierarchies
  - Object classification
  - Classifications use cases
- 3 URN handling problems**
- 4 The future



# On URN handling problems

or, more precisely, their absence

URNs usage problems are more perceived than real

- **Searching for URNs**
- Entitlement processing
- URN processing

URN = text string

**When properly indexed**, LDAP shines for its speed in substring searching; regardless of length. (We have benchmarks to back this).





# On URN handling problems

or, more precisely, their absence

URNs usage problems are more perceived than real

- Searching for URNs
- **Entitlement processing**
- URN processing

**Entitlement = multivalued attribute**

Processing is not more complex than with any other multivalued attributes.



# On URN handling problems

or, more precisely, their absence

URNs usage problems are more perceived than real

- Searching for URNs
- Entitlement processing
- **URN processing**

## URN = text string

Searching for information inside a URN is just string processing, which most programming languages in use can easily accomplish.



# Outline

- 1 Use cases of URNs in Entitlements
  - Expenses Authorization Control
  - Mobile phone number usage control
- 2 URNs for adding hierarchies
  - Object classification
  - Classifications use cases
- 3 URN handling problems
- 4 The future



# The future is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications  
produced by teams that are far from middleware.



# The future is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications  
produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- Our quest for a solution



# The future

is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications  
produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- Our quest for a solution



# The future

is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- Our quest for a solution

## The access, **NOT** the directory

The directory can't know if the application is using the credentials it should use.

Then, applications could use information they are not authorized to.

# The future

is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- Our quest for a solution

## Credentials control

Applications SHOULD NOT have access to user credentials.  
Why? They may abuse them.  
We have already done that.





# The future

is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- **Our quest for a solution**

## Web services

As an interface between applications and the directory. Attribute access policy enforcing can be verified.  
Good for in-house applications.  
Difficult for third party applications.

# The future

is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- **Our quest for a solution**

## Kerberos

Can do AuthN.

Can do AuthZ?

There are kerberized third party applications, but not many.



# The future

is uncertain

We are building an AAI based on the ideas presented here.  
We have tried hard to apply AAI concepts to applications produced by teams that are far from middleware.  
And many doubts have arisen

- Rule out LDAP?
- **Our quest for a solution**

## Web AAI

Easily applied to web applications with source.

Can be ported to web servers to avoid application modification.

Non web applications?

