A Directory Based Repository for OSI Management Information Knowledge

Olivier Festor, Nizar Ben Youssef

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

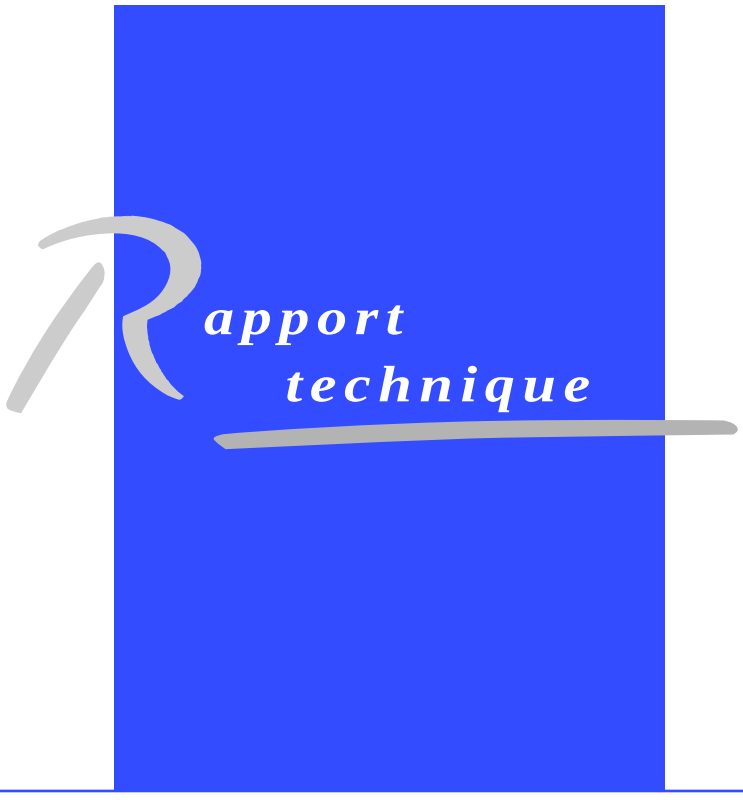
A Directory Based Repository for OSI Management Information Knowledge

Olivier Festor, Nizar Ben Youssef

N° 0264

July 16, 2002

THÈME 1



ISSN 0249-0803

A Directory Based Repository for OSI Management Information Knowledge

Olivier Festor, Nizar Ben Youssef

Thème 1 — Réseaux et systèmes
Projet RÉSÈDAS

Rapport technique n° 0264 — July 16, 2002 — pages

Abstract: This document presents an approach to store GDMO (Guidelines for the Definition of Managed Objects) specifications within a LDAP (Lightweigth Directory Access Protocol) directory server. It defines a set of LDAP object classes and attributes enabling the representation of any GDMO model within the directory. It also presents the implementation provided within the latest distribution of the MODERES-Java toolkit together with a Java API that facilitates the access to the stored GDMO specifications.

Key-words: GDMO, CMIS, Management Information, LDAP, Directories

Ce travail a été réalisé dans le cadre de l'action ANTARES (Architectures et Nouvelles Technologies pour l'Administration des Réseaux et Service) du GIE DYADE.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Utilisation d'un service d'annuaire pour le partage d'informations de gestion OSI

Résumé : Ce rapport présente une solution pour le stockage des modèles de l'information spécifiés dans le langage GDMO (Guidelines for Definition of Managed Objects) dans un serveur d'annuaire LDAP (Lightweight Directory Access Protocol). Il propose un ensemle de classes et d'attributs LDAP permettant de représenter toute modélisation GDMO dans un annuaire LDAP. Il décrit aussi l'implémentation qui a été réalisée et qui est disponible dans la distribution actuelle de MODERES-Java ainsi qu'une interface de programmation en Java facilitant l'accès aux spécifications GDMO sauvegardées dans un annuaire LDAP.

Mots-clés : GDMO, CMIS, Information de gestion, LDAP, Services d'annuaires

Contents

1 Introduction

In all network and service management solutions, the sharing of management information models between applications is a major requirement. Even for basic management operations, managers need to know the information models supported by their agents. Unlike the WBEM (Web-Based Enterprise Management) [1] approach which defines a functional profile for retrieving management information models, the OSI management approach still lacks the availibility of such a service, leading management platform constructors to provide complex proprietary solutions.

Directory services in general, and particulary LDAP (Ligthweight Directory Access Protocol) [2] directory services, offer an easy to implement and powerful standard way for storing and sharing information between applications. Initially developped as a support to web-based applications such as white pages and e-mail servers, directory services have reached great success in many other areas. In the network and service management domain, LDAP directory services have been recently proposed within the DEN (Directory-Enabled Networks) [3] approach to store management policies for policy-based management applications.

Combining the success of LDAP directory services, and the need to share management information models between OSI management applications, are the major motivations of the study presented in this report. This document presents a directory-based repository for storing OSI (Open Systems Interconnection) management information knowledge formalized in the GDMO language (Guidelines for Definition of Managed Objects) [4]. The implementation called MISS-API (Management Information Schema Service API) builded on top of the mapping issued from this approach, enables OSI management applications to retrieve the management information knowledge from a LDAP directory server.

The structure of this document is as follows. In section 2 we introduce the architecture of our MIS (Management Information Schema) service. Section 3 provides an overview of the LDAP schema that has been defined to store, in a directory server, GDMO based management information models. Section 4 details the attributes and object classes of the LDAP schema. Section 5 presents the MISS-API implementation. Finally, in section 6, conclusions are given.

2 Architecture

A MIS (Management Information Schema) service is used by management applications to access the information models supported by a management platform. Such a service needs a central repository (database), called Management Information Schema Base or simply "Schema Base", to store management information models. It is usually implemented using proprietary frameworks. For example, the Evidian [5] OpenMaster management platform provides a MISS API available for applications written in SML (System Management Language).

In order to develop a standard-based MIS service, we performed a study on using LDAP [12, 13, 14, 7, 8, 11] (Ligthweigth Directory Access Protocol) directory services as a central repository for OSI management information knowledge. In fact, LDAP directory services is nowadays considered to be an easy and powerfull way for storing and retrieving application's static data, which is adequate for a MIS service application.

Usually in OSI management approach, management information desfinitons are written in GDMO files. Then, at agent initialization time, those files are compiled also by the manager and stored within a manager specific database (the MIS base). In this mechanism, managers should provide and maintain the repository.

In our directory based central repository for management information knowledge, GDMO files are compiled and stored within the directory. Then, only relevant management information models are accessed, throw the standard LDAP protocol, by the manager when performing management operations. In this approach, the two basic components of the MIS service, which are the MIS Base and the MIS protocol are part of the LDAP directory service. This scenario is illustrated in figure 1. The major benefit of this approach is to offer a standard way to access information. It also delegates part of the management process to the directory service, which seems relevant for this specific case.

[1] Standardised by the DMTF (Distributed Manageemnt Task Force) :`http://www.dmtf.org`
[2] Standardised by the IETF (Internet Engineering Task Force) :`http://www.ietf.org`
[3] Standardised by both DMTF and IETF
[4] Standardised by the ITU (International Telecommunication Union)
[5] `http://www.evidian.com`

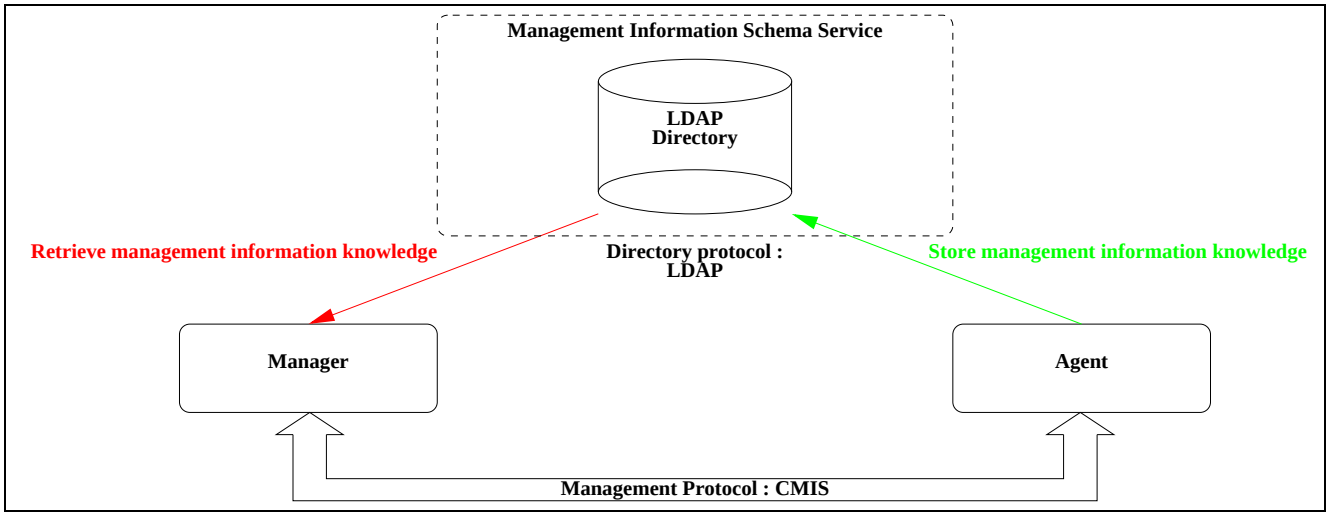


Figure 1: Architecture of the MIS service

3 Overview of the LDAP schema

This section discusses the directory model used to store any GDMO [2] specification within a LDAP server. First, it gives an overview of the information model used for this purpose, introducing the set of LDAP object classes defining the schema [13]. An overview of the naming model defining the hierarchy of entries within the directory is also given. Detailed specifications of this models are given in the section 4.

3.1 The information model

The LDAP information model, called LDAP schema, is the set of object classes and attributes used to store the application data in the directory. To store the OSI management information knowledge, we defined a LDAP schema enabling the representation of any GDMO specification within the directory. This mapping is based on a *technique mapping*, wich consists in defining for each GDMO meta-element (GDMO templates for example) a set of LDAP object classes and attributes enabling its representation in the directory. So, any GDMO definition based on those meta-elements would be represented by one or more entries in the directory. In this section, we give an overview of the main LDAP object classes defined for this purpose :

- **gdmo-module** is used to represent a module declaration ;
- **gdmo-action** is used to represent an action template ;
- **gdmo-attribute** is used to represent an attribute template ;
- **gdmo-attributegroup** is used to represent an attribute group template ;
- **gdmo-behaviour** is used to represent a behaviour template ;
- **gdmo-managedobjectclass** is used to represent a managed object class template. Entries of this class may contain entries of **gdmo-conditionalpackage** object class for a complete representation of the template optional features ;
- **gdmo-namebinding** is used to represent a name-binding template ;
- **gdmo-notification** is used to represent a notification template. Entries of this class may contain entries of **gdmo-notificationinfosyntaxfield** object class for a complete representation of the template optional features ;
- **gdmo-package** is used to represent a package template. Entries of this class may contain entries of **gdmo-actionandparameters**, **gdmo-attributeandproperties**, **gdmo-attributegroupelement** and **gdmo-notificationandparameters** object classes for a complete representation of the template optional features ;

- **gdmo-parameter** is used to represent a parameter template.

The complete set of LDAP object classes defining the schema, and relationships between them, are given in figure 2.

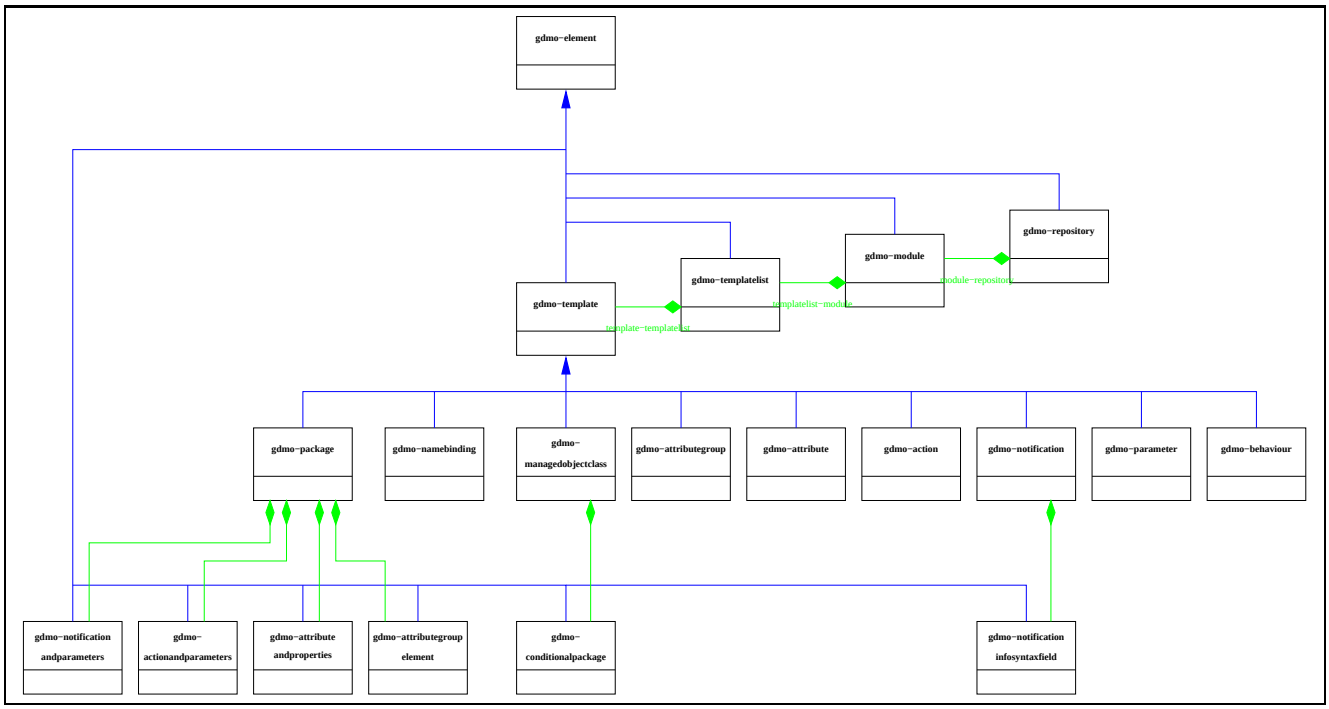


Figure 2: LDAP schema object classes

3.2 The naming model

The naming model defines the containement relationship, in the directory information tree (DIT) between entries issued from the information model. It also defines the naming attributes used to identify those entries. Such definitions can not be defined in an explicit manner in the LDAP model, but are application specific recommendations.

Aggregation relationships between object classes shown in figure 2, form the basis for the naming model and do not appear in the LDAP information model in an explicit manner. Such relationships are only conceptual notations used to define our specific application directory information tree. At an implementation level, aggregations are mapped into containment relationships between entries issued from the information model, as explained below :

- *module-repository aggregation :* each `gdmo-module` entry should be registered in the scope of the top-level `gdmo-repository` entry. For this containment, the `gdmo-modulename` attribute should be used as a naming attribute ;
- *templatelist-module aggregation :* each `gdmo-templatelist` entry should be registered in the scope of a `gdmo-module` entry. For this containment, `gdmo-templatetype` should be used as a naming attribute. This relationship does not appear in GDMO constructs, but is only introduced in our LDAP schema for organisational purpose to separate templates entries according to their types ;
- *template-templatelist aggregation :* each `gdmo-template` entry (wich include every `gdmo-template` subclasses entries) should be registered in the scope of a `gdmo-templatelist` entry. For this containment, `gdmo-label` attribute should be used as a naming attribute. As a GDMO template name is unique in the

scope of the module in which it is defined, association of the *template-templatelist* and *templatelist-module* aggregations insure uniquiness of this template naming model ;

- *condition-class aggregation :* each `gdmo-conditionalpackage` entry should be registered in the scope of a `gdmo-managedobjectclass` entry (the one representing the managed object class in which it is defined). For this containment, the `gdmo-label` attribute should be used as a naming attribute ;
- *action-package aggregation :* each `gdmo-actionandparameters` entry should be registered in the scope of a `gdmo-package` entry (the one representing the package in which it is defined). For this containment, the `gdmo-actionlabel` attribute should be used as a naming attribute ;
- *attribute-package aggregation :* each `gdmo-attributeandproperties` entry should be registered in the scope of a `gdmo-package` entry (the one representing the package in which it is defined). For this containment, the `gdmo-attributelabel` attribute should be used as a naming attribute ;
- *attributegroup-package aggregation :* each `gdmo-attributegroupelement` entry should be registered in the scope of a `gdmo-package` entry (the one representing the package in which it is defined). For this containment, the `gdmo-attributegrouplabel` attribute should be used as a naming attribute ;
- *notification-package aggregation :* each `gdmo-notificationandparameters` entry should be registered in the scope of a `gdmo-package` entry (the one representing the package in which it is defined). For this containment, the `gdmo-notificationlabel` attribute should be used as a naming attribute ;

Figure 3 gives an example of the directory information tree hierarchy that should be stored in the directory, based on the containement relationships presented here. It also illustrates the naming model of the entries in the directory tree.

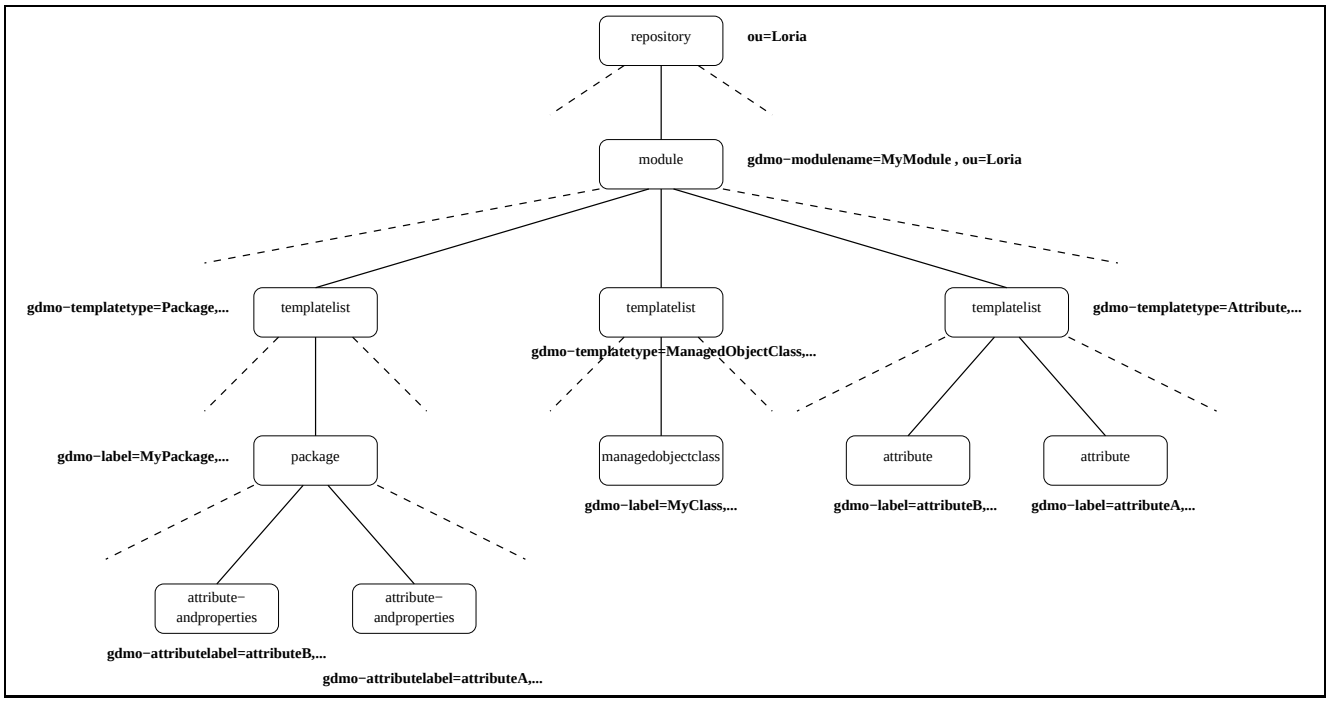


Figure 3: An example of the directory information tree

4 Specification of the LDAP schema

This section presents the directory schema used to store any GDMO specification within a LDAP server. The schema defines a representation for any GDMO data structure called GDMO template. The representation of ASN.1 [10] data structures referenced by some GDMO templates is out of the scope of this document.

This section first details the set of LDAP attributes defined within our schema. Then, it details the set of LDAP object classes used to store the GDMO templates as directory entries. Examples of such entries are also given in the form of LDIF notations (LDAP Data Interchange Format).

4.1 Attribute types

The specifications given in this section are based on the concept of attribute syntax inheritance. Such paradigm is authorized in the version 3 of LDAP information specification, without being mandatory for server implementation. Although the attribute syntax inheritance is used in the present document, it is not fundamental and can be easily replaced by an explicit syntax declaration for attribute definitions.

4.1.1 gdmo-label

The **gdmo-label** attribute is used to store a GDMO template label. It is single valued and takes the `Directory String` syntax inherited from the `name` attribute defined in [11].

```
( <gdmo-label-oid>  NAME  ' gdmo-label '
    SUP name
    SINGLE-VALUE
)
```

4.1.2 gdmo-actionlabel

The **gdmo-actionlabel** is used by the **gdmo-actionandparameters** entries as a naming attribute. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-actionlabel-oid>  NAME  ' gdmo-actionlabel '
    SUP gdmo-label
)
```

4.1.3 gdmo-attributelabel

The **gdmo-attributelabel** is used by the **gdmo-attributeandproperties** entries as a naming attribute. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-attributelabel-oid>  NAME  ' gdmo-attributelabel '
    SUP gdmo-label
)
```

4.1.4 gdmo-attributegrouplabel

The **gdmo-attributegrouplabel** is used by the **gdmo-attributegroupelement** entries as a naming attribute. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-attributegrouplabel-oid>  NAME  ' gdmo-attributegrouplabel '
    SUP gdmo-label
)
```

4.1.5 gdmo-classlabel

The **gdmo-classlabel** is used by the **gdmo-mocsemanticlinks** entries as a naming attribute. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-classlabel-oid>  NAME  ' gdmo-classlabel '
    SUP gdmo-label
)
```

4.1.6 gdmo-notificationlabel

The **gdmo-notificationlabel** is used by **gdmo-notificationandparameters** entries as a naming attribute. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-notificationlabel-oid>  NAME  ' gdmo-notificationlabel '
    SUP gdmo-label
)
```

4.1.7 gdmo-namingattributelabel

The **gdmo-namingattributelabel** is used by **gdmo-namebinding** entries to store the naming attribute label in a name-binding template. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-namingattributelabel-oid>  NAME  ' gdmo-namingattributelabel '
    SUP gdmo-label
)
```

4.1.8 gdmo-superiorclasslabel

The **gdmo-superiorclasslabel** is used by **gdmo-namebinding** entries to store the superior object class label in a name-binding template. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-superiorclasslabel-oid>  NAME  ' gdmo-superiorclasslabel '
    SUP gdmo-label
)
```

4.1.9 gdmo-subordinateclasslabel

The **gdmo-superiorclasslabel** is used by **gdmo-namebinding** entries to store the subordinate object class label in a name-binding template. It inherits a `Directory String` syntax from the `gdmo-label` attribute.

```
( <gdmo-subordinatelabel-oid>  NAME  ' gdmo-subordinateclasslabel '
    SUP gdmo-label
)
```

4.1.10 gdmo-list

The **gdmo-list** is used by **gdmo-templatelist** entries to store the list of template labels. It is multi-valued and inherits a `Directory String` syntax from the `name` attribute.

```
( <gdmo-list-oid> NAME ' gdmo-list '
    SUP name
)
```

4.1.11 gdmo-actionlist

The **gdmo-actionlist** is used by **gdmo-package** entries to store the labels of actions declared by a package template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-actionlist-oid> NAME ' gdmo-actionlist '
    SUP gdmo-list
)
```

4.1.12 gdmo-attributelist

The **gdmo-attributelist** is used by **gdmo-package** entries to store the labels of attributes declared by a package template. It is also used by **gdmo-attributegroup** entries to store the labels of attributes declared by an attribute group template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-attributelist-oid> NAME ' gdmo-attributelist '
    SUP gdmo-list
)
```

4.1.13 gdmo-attributegrouplist

The **gdmo-attributegrouplist** is used by **gdmo-package** entries to store the labels of attribute groups declared by a package template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-attributegrouplist-oid> NAME ' gdmo-attributegrouplist '
    SUP gdmo-list
)
```

4.1.14 gdmo-behaviourlist

The **gdmo-behaviourlist** is used by any GDMO template entry that may declare a behaviour template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-behaviourlist-oid> NAME ' gdmo-behaviourlist '
    SUP gdmo-list
)
```

4.1.15 gdmo-notificationlist

The **gdmo-notificationlist** is used by **gdmo-package** entries to store the labels of notifications declared by a package template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-notificationlist-oid> NAME ' gdmo-notificationlist '
    SUP gdmo-list
)
```

4.1.16 gdmo-parameterlist

The **gdmo-parameterlist** is used by any GDMO construct that may declare a parameter template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-parameterlist-oid>  NAME  ' gdmo-parameterlist '
    SUP gdmo-list
)
```

4.1.17 gdmo-packagelist

The **gdmo-packagelist** is used by **gdmo-managedobjectclass** entries to store the labels of packages declared by a managed object class template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-packagelist-oid>  NAME  ' gdmo-packagelist '
    SUP gdmo-list
)
```

4.1.18 gdmo-conditionalpackagelist

The **gdmo-conditionalpackagelist** is used by **gdmo-managedobjectclass** entries to store the labels of conditional packages declared by a managed object class template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-conditionalpackagelist-oid>  NAME  ' gdmo-conditionalpackagelist '
    SUP gdmo-list
)
```

4.1.19 gdmo-superclasslist

The **gdmo-superclasslist** is used by **gdmo-managedobjectclass** entries to store the labels of super-classes declared by a managed object class template. It is also used by **gdmo-mocsemanticlinks** entries to store labels of all super-classes for a given managed object class. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-superclasslist-oid>  NAME  ' gdmo-superclasslist '
    SUP gdmo-list
)
```

4.1.20 gdmo-subclasslist

The **gdmo-subclasslist** is used by **gdmo-mocsemanticlinks** to store the labels of all sub-classes for a given managed object class. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-subclasslist-oid>  NAME  ' gdmo-subclasslist '
    SUP gdmo-list
)
```

4.1.21 gdmo-superiorclasslist

The **gdmo-superiorclasslist** is used by **gdmo-mocsemanticlinks** to store the labels of all superior classes (declared as superior by a name-binding) for a given managed object class. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-superiorclasslist-oid>  NAME  ' gdmo-superiorclasslist '
    SUP gdmo-list
)
```

4.1.22 gdmo-subordinateclasslist

The **gdmo-subordinateclasslist** is used by **gdmo-mocsemanticlinks** entries to store the labels of all subordinate classes (declared as subordinate by a name-binding) for a given managed object class. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-subordinateclasslist-oid>  NAME  ' gdmo-subordinateclasslist '
    SUP gdmo-list
)
```

4.1.23 gdmo-additionalattributelist

The **gdmo-additionalattributelist** is used by **gdmo-attributegrouptemplate** entries to store additional attribute declaration when including an attribute group template in a package template. It is multi-valued and inherits a `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-additionalattributelist-oid>  NAME  ' gdmo-additionalattributelist '
    SUP gdmo-list
)
```

4.1.24 gdmo-actionmode

The **gdmo-actionmode** is used by **gdmo-action** entries to store an action mode template. Its syntax is a `Directory String` and its possible values are :
`CONFIRMED` , `UNCONFIRMED`

```
( <gdmo-actionmode-oid>  NAME  ' gdmo-actionmode '
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 {16}
    SINGLE-VALUE
)
```

4.1.25 gdmo-actioninfosyntax

The **gdmo-actioninfosyntax** is used by **gdmo-action** entries to store the ASN.1 type reference for an action information syntax. Its syntax is a `Directory String` formatted as follows :
`<asn1-document>` "." `<asn1-type>`

```
( <gdmo-actioninfosyntax-oid>  NAME  ' gdmo-actioninfosyntax '
    SUP name
    SINGLE-VALUE
)
```

4.1.26 gdmo-actionreplysyntax

The **gdmo-actionreplysyntax** is used by **gdmo-action** entries to store the ASN.1 type reference for an action reply syntax. Its syntax is a `Directory String` formatted as follows :

`<asn1-document>` **“.”** `<asn1-type>`

```
( <gdmo-actionreplysyntax-oid>  NAME  ' gdmo-actionreplysyntax '
    SUP name
    SINGLE-VALUE
)
```

4.1.27 gdmo-attributegroupmode

The **gdmo-attributegroupmode** is used by **gdmo-attributegroup** entries to store an attribute group template mode. Its syntax is a `Directory String` and its possible values are :

`FIXED` , `NOT-FIXED`

```
( <gdmo-attributegroupmode-oid>  NAME  ' gdmo-attributegroupmode '
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 {16}
    SINGLE-VALUE
)
```

4.1.28 gdmo-attributegroupdescription

The **gdmo-attributegroupdescription** is used by **gdmo-attributegroup** entries to store an attribute group description. It inherits the `Directory String` syntax from the `description` attribute [11].

```
( <gdmo-attributegroupdescription-oid>  NAME  ' gdmo-attributegroupdescription '
    SUP description
    SINGLE-VALUE
)
```

4.1.29 gdmo-attributesuperior

The **gdmo-attributesuperior** is used by **gdmo-attribute** entries to store the label of the superior attribute for an attribute syntax declaration. It inherits the `DirectoryString` syntax from the `gdmo-label` attribute.

```
( <gdmo-attributesuperior-oid>  NAME  ' gdmo-attributesuperior '
    SUP gdmo-label
)
```

4.1.30 gdmo-attributesyntax

The **gdmo-attributesyntax** is used by **gdmo-attribute** entries to store the ASN.1 type reference of an attribute syntax. Its syntax is a `Directory String` formatted as follows :

`<asn1-document>` **“.”** `<asn1-type>`

```
( <gdmo-attributesyntax-oid>  NAME  ' gdmo-attributesyntax '
    SUP name
    SINGLE-VALUE
)
```

4.1.31 gdmo-createmodifier

The **gdmo-createmodifier** is used by **gdmo-namebinding** entries to store the list of associated create modifiers for a name-binding template. Its syntax is a `Directory String` and its possible values are :
`WITH-REFERENCE-OBJECT` , `WITH-AUTOMATIC-INSTANCE-NAMING`

```
( <gdmo-createmodifier-oid>  NAME  ' gdmo-createmodifier '
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 {32}
)
```

4.1.32 gdmo-createparameterlist

The **gdmo-createparameters** is used by **gdmo-namebinding** entries to store the list of parameter labels associated with create modifiers for a name-binding template. It is multi-valued and inherits `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-createparameterlist-oid>  NAME  ' gdmo-createparameterlist '
    SUP gdmo-list
)
```

4.1.33 gdmo-deletemodifier

The **gdmo-deletemodifier** is used by **gdmo-namebinding** entries to store the associated delete modifier for a name-binding template. Its syntax is a `Directory String` and its possible values are :
`ONLY-IF-NO-CONTAINED-OBJECTS` , `DELETES-CONTAINED-OBJECTS`

```
( <gdmo-deletemodifier-oid>  NAME  ' gdmo-deletemodifier '
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 {32}
)
```

4.1.34 gdmo-deleteparameterlist

The **gdmo-deleteparameters** is used by **gdmo-namebinding** entries to store the list of parameter labels associated with delete modifier for a name-binding template. It is multi-valued and inherits `Directory String` syntax from the `gdmo-list` attribute.

```
( <gdmo-deleteparameterlist-oid>  NAME  ' gdmo-deleteparameterlist '
    SUP gdmo-list
)
```

4.1.35 gdmo-fieldname

The **gdmo-fieldname** is used by **gdmo-notificationinfosyntaxfield** as a naming attribute. It inherits `Directory String` syntax from the `name` attribute.

```
( <gdmo-fieldname-oid>  NAME  ' gdmo-fieldname '
    SUP name
    SINGLE-VALUE
)
```

4.1.36 gdmo-notificationinfosyntax

The **gdmo-notificationinfosyntax** is used by **gdmo-notification** entries to store the ASN.1 type reference for a notification information syntax. Its syntax is a `Directory String` formatted as follows :
`<asn1-document>` "." `<asn1-type>`

```
( <gdmo-notificationinfosyntax-oid>  NAME  ' gdmo-notificationinfosyntax '
    SUP name
    SINGLE-VALUE
)
```

4.1.37 gdmo-notificationreplysyntax

The **gdmo-notificationreplysyntax** is used by **gdmo-notification** entries to store the ASN.1 type reference for a notification reply syntax. Its syntax is a `Directory String` formatted as follows :
`<asn1-document>` "." `<asn1-type>`

```
( <gdmo-notificationreplysyntax-oid>  NAME  ' gdmo-notificationreplysyntax '
    SUP name
    SINGLE-VALUE
)
```

4.1.38 gdmo-registrationID

The **gdmo-registrationID** is used by any GDMO template that need a registration identifier. Its syntax is the `OID` syntax defined in [13].

```
( <gdmo-registrationID-oid>  NAME  ' gdmo-registrationID '
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
    SINGLE-VALUE
)
```

4.1.39 gdmo-superiorsubclasssupport

The **gdmo-superiorclasssupport** is used by **gdmo-namebinding** entries to extend the superior role in the name-binding relationship to all subclasses of the declared superior object class. It has a `BOOLEAN` syntax and takes the `true` value if the name-binding has to be extended to all subclasses of the superior class, `false` otherwise.

```
( <gdmo-superiorsubclasssupport-oid>  NAME  ' gdmo-superiorsubclasssupport '
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
    SINGLE-VALUE
)
```

4.1.40 gdmo-subordinatesubclasssupport

The **gdmo-subordinateclasssupport** is used by **gdmo-namebinding** entries to extend the subordinate role in the name-binding relationship to all subclasses of the declared subordinate object class. It has a `BOOLEAN` syntax and takes the `true` value if the name-binding has to be extended to all subclasses of the subordinate class, `false` otherwise.

```
( <gdmo-subordinatesubclasssupport-oid> NAME ' gdmo-subordinatesubclasssupport '
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
    SINGLE-VALUE
)
```

4.1.41 gdmo-templatetype

The **gdmo-templatetype** is used by **gdmo-templatelist** entries to register different template types under different nodes in the directory. Its syntax is a `Directory String` and its possible values are :
`Action` , `Attribute` , `AttributeGroup` , `Behaviour`, `ManagedObjectClass` , `NameBinding` , `Notification` , `Package` , `Parameter`.

```
( <gdmo-templatetype-oid> NAME ' gdmo-templatetype '
    SUP name
    SINGLE-VALUE
)
```

4.1.42 gdmo-condition

The **gdmo-condition** is used by **gdmo-conditionalpackage** entries to represent the condition attached with the conditional package. It inherits `Directory String` syntax from the `description` attribute.

```
( <gdmo-condition-oid> NAME ' gdmo-condition '
    SUP description
    SINGLE-VALUE
)
```

4.2 Object classes

The specifications given in this section are based on the concept of object class inheritance. This concept is allowed in the version 3 of LDAP information specification, without being mandatory for server implementation. Although the object class inheritance is used in the present document, it is not fundamental and can be easily replaced at implementation level by an explicit declaration of the whole set of attributes for any object class definition, including those declared by named super-classes.

4.2.1 gdmo-element

The **gdmo-element** object class is used as a super-class for all the object classes defined by this document. It is abstract, and it inherits the `objectclass` attribute from the **top** object class [11].

```
( <gdmo-element-oid> NAME ' gdmo-element '
    SUP top
    ABSTRACT
)
```

4.2.2 gdmo-repository

The **gdmo-repository** object class is used to create the top-level entry for a GDMO repository. Only one entry of this type should be created in the directory. The **description** attribute is defined in [11].

```
( <gdmo-repository-oid> NAME ' gdmo-repository '
    SUP gdmo-element
```

```
    MAY description
)
```

4.2.3 gdmo-module

The **gdmo-module** object class is used as a top-level entry to store GDMO elements declared by a module.

```
( <gdmo-module-oid>  NAME  ' gdmo-module '
    SUP gdmo-element
    MUST gdmo-modulename
)
```

4.2.4 gdmo-templateList

The **gdmo-templatelist** object class is an organisational construct used to separate templates entries depending on their types (action templates, attribute templates, etc...). Entries of this object class are registred in a **gdmo-module** sub-tree, and only one entry for each template type can be registered under a given **gdmo-module** entry.

```
( <gdmo-templatelist-oid>  NAME  ' gdmo-templatelist '
    SUP gdmo-element
    MUST gdmo-templatetype
    MAY gdmo-list
)
```

4.2.5 gdmo-template

The **gdmo-template** object class is used as a super-class for all the object classes representing a GDMO template (`gdmo-action`, `gdmo-attribute`, `gdmo-attributegroup`, `gdmo-behaviour`, `gdmo-managedobjectclass`, `gdmo-namebinding`, `gdmo-notification`, `gdmo-parameter` and `gdmo-package`). It is abstract, and can not be instanciated.

```
( <gdmo-template-oid>  NAME  ' gdmo-template '
    SUP gdmo-element
    ABSTRACT
    MUST gdmo-label
    MAY gdmo-registrationID
)
```

4.2.6 gdmo-action

The **gdmo-action** object class is used to store in the directory the representation of an action template.

```
( <gdmo-action-oid>  NAME  ' gdmo-action '
    SUP gdmo-template
    MAY gdmo-behaviourlist ,
       gdmo-parameterlis ,
       gdmo-actionmode ,
       gdmo-actioninfosyntax ,
       gdmo-actionreplysyntax
)
```

4.2.7 gdmo-attribute

The **gdmo-attribute** object class is used to store in the directory the representation of an attribute template.

```
( <gdmo-attribute-oid>  NAME  ' gdmo-attribute '
    SUP gdmo-template
    MAY gdmo-behaviourlist ,
       gdmo-parameterlist ,
       gdmo-attributesuperior ,
       gdmo-attributesyntax ,
       gdmo-matchingrulelist
)
```

4.2.8 gdmo-attributegroup

The **gdmo-attributegroup** object class is used to store in the directory the representation of an attribute group template.

```
( <gdmo-attributegroup-oid>  NAME  ' gdmo-attributegroup '
    SUP gdmo-template
    MAY gdmo-attributelist ,
       gdmo-attributegroupmode ,
       gdmo-attributegroupdescription
)
```

4.2.9 gdmo-behaviour

The **gdmo-behaviour** object class is used to store in the directory the representation of a behaviour template.

```
( <gdmo-behaviour-oid>  NAME  ' gdmo-behaviour '
    SUP gdmo-template
    MAY gdmo-behaviourdefinition
)
```

4.2.10 gdmo-managedobjectclass

The **gdmo-managedobjectclass** object class is used to store in the directory the representation of a managed object class template.

```
( <gdmo-managedobjectclass-oid>  NAME  ' gdmo-managedobjectclass '
    SUP gdmo-template
    MAY gdmo-superclasslist ,
       gdmo-packagelist ,
       gdmo-conditionalpackage
)
```

4.2.11 gdmo-namebinding

The **gdmo-namebinding** object class is used to store in the directory the representation of a name-binding template.

```
( <gdmo-namebinding-oid>  NAME  ' gdmo-namebinding '
    SUP gdmo-template
```

```
    MAY gdmo-behaviourlist ,
        gdmo-superiorclasslabel ,
        gdmo-superiorsubclasssupport ,
        gdmo-subordinateclasslabel ,
        gdmo-subordinatesubclasssupport ,
        gdmo-namingattributelabel ,
        gdmo-createmodifier ,
        gdmo-createparameterlist ,
        gdmo-deletemodifier ,
        gdmo-deleteparameterlist
)
```

4.2.12 gdmo-notification

The **gdmo-notification** object class is used to store in the directory the representation of a notification template.

```
( <gdmo-notification-oid> NAME ' gdmo-notification '
    SUP gdmo-template
    MAY gdmo-behaviourlist ,
        gdmo-parameterlist ,
        gdmo-notificationinfosyntax ,
        gdmo-notificationinfosyntaxfieldlist ,
        gdmo-notificationreplysyntax
)
```

4.2.13 gdmo-pacakage

The **gdmo-package** object class is used to store in the directory the representation of a package template.

```
( <gdmo-package-oid> NAME ' gdmo-package '
    SUP gdmo-template
    MAY gdmo-behaviourlist ,
        gdmo-actionlist ,
        gdmo-attributelist ,
        gdmo-attributegrouplist ,
        gdmo-notificationlist
)
```

4.2.14 gdmo-parameter

The **gdmo-parameter** object class is used to store in the directory the representation of a parameter template.

```
( <gdmo-parameter-oid> NAME ' gdmo-parameter '
    SUP gdmo-template
    MAY gdmo-behaviourlist ,
        gdmo-parametercontex ,
        gdmo-parametersyntax ,
        gdmo-parameterattribute
)
```

4.2.15 gdmo-conditionalpackage

The **gdmo-conditionalpackage** object class is used to store in the directory the representation of a declaration of a conditional package for a managed object class template. Entries of this type must be registered in the scope of a **gdmo-managedobjectclass** entry.

```
( <gdmo-conditionalpackage-oid>  NAME  ' gdmo-conditionalpackage '
    SUP gdmo-element
    MUST gdmo-label
    MAY gdmo-condition
)
```

4.2.16 gdmo-actionandparameters

The **gdmo-actionandparameters** object class is used to store in the directory the representation of a declaration of an action with it's possibly associated parameters for a package template. Entries of this type must be registered in the scope of a **gdmo-package** entry.

```
( <gdmo-actionandparameters-oid>  NAME  ' gdmo-actionandparameters '
    SUP gdmo-element
    MUST gdmo-actionlabel
    MAY gdmo-parameterlist
)
```

4.2.17 gdmo-attributeandproperties

The **gdmo-attributeandproperties** object class is used to store in the directory the representation of a declaration of an attribute with it's mandatory associated properties and possibly associated parameters for a package template. Entries of this type must be registered in the scope of a **gdmo-package** entry.

```
( <gdmo-attributeandproperties-oid>  NAME  ' gdmo-attributeandproperties '
    SUP gdmo-element
    MUST gdmo-attributelabel
    MAY gdmo-parameterlist ,
       gdmo-propertylist
)
```

4.2.18 gdmo-attributegroupelement

The **gdmo-attributegroupelement** object class is used to store in the directory the representation of a declaration of an attribute group with it's possibly associated additional attributes for a package template. Entries of this type must be registered in the scope of a **gdmo-package** entry.

```
( <gdmo-attributegroupelement-oid>  NAME  ' gdmo-attributegroupelement '
    SUP gdmo-element
    MUST gdmo-attributegrouplabel
    MAY gdmo-additionalattributelist
)
```

4.2.19 gdmo-notificationandparameters

The **gdmo-notificationandparameters** object class is used to store in the directory the representation of a declaration of a notification with it's possibly associated parameters for a package template. Entries of this type must be registered in the scope of a **gdmo-package** entry.

```
( <gdmo-notificationandparameter-oid>  NAME  ' gdmo-notificationandparameter '
    SUP gdmo-template
    MUST gdmo-notificationlabel
```

```
4       MAY gdmo-parameterlist
5   )
```

4.2.20 gdmo-notificationinfosyntaxfield

The **gdmo-notificationinfosyntax** object class is used to store in the directory the representation of notification information syntax with it's possibly associated attribute field list. Entries of this type must be registered in the scope of a **gdmo-notification** entry.

```
1   ( <gdmo-notificationinfosyntax-oid>  NAME  ' gdmo-notificationinfosyntax '
2       SUP gdmo-element
3       MUST gdmo-fieldname
4       MAY gdmo-label
5   )
```

4.2.21 gdmo-mocsemanticlinks

The **gdmo-mocsemanticlinks** object class is used to store in the directory semantic links related to a managed object class such as the list of all super-classes. Entries of this type are registered in the scope of **gdmo-managedobjectclass** entry, and represent convenience objects for some high-level semantic search operations over the directory. Such semantic links are done by a complete semantic check by MODERES Java tools before storing operation, and must be reviewed for each new module registration operation.

```
1   ( <gdmo-mocsemanticlinks-oid>  NAME  ' gdmo-mocsemanticlinks '
2       SUP top
3       AUXILIARY
4       MAY gdmo-subclasslist ,
5          gdmo-superclasslist ,
6          gdmo-subordinateclasslist ,
7          gdmo-superiorclasslist ,
8          gdmo-attributelist ,
9          gdmo-actionlist ,
10         gdmo-notificationlist
11  )
```

5 Implementation and API user guide

At an implementation level, we have developped a set of methods for the GDMO/LDAP mappings. Those methods are included in the MODERES packages and are presented in the first part of this section. We have also developped a client API, based on those mapping features, to be used by management application for schema retrieval. This API is implemented in a separate package, and is presented in the second part of this section.

5.1 The MODERES mapping features

The MODERES Java toolkit developed by the RESEDAS research group, contains a set of parsers, semantics checkers and APIs, enabling handling management information models issued from different standards.

For OSI management information knowledge, formalized in GDMO notations, MODERES Java provides a GDMO parser [6, 4], a semantic checker[5], and several backends such as the GDMO/HTML [3] backend for visualisation and the GDMO/SDL'92 [1] for validation.

To enable the storage and retrieval of GDMO specifications to and from a LDAP directory, we have extended the MODERES Java toolkit with a new backend called **GDMO/LDAP Translator**. This backend contains two set of static methods provided by two Java classes within the MODERES **fr.loria.resedas.moderes.ldap-mappings** package. These classes are :

- **MODERES_GDMOtoLDAPTranslator :** contains static methods enabling storing in a LDAP directory server java objects, defined in the MODERES package **fr.loria.resedas.moderes.gdmorepository**, and representing previously parsed and checked GDMO specifications ;
- **MODERES_LDAPtoGDMOTranslator :** contains static methods enabling retrieving from a LDAP directory, java objects defined in the MODERES package **fr.loria.resedas.moderes.gdmorepository**, and representing GDMO specifications previously stored in the directory.

Detailed description of the two classes `MODERES_GDMOtoLDAPTranslator` and `MODERES_LDAPtoGDMOTranslator`, in addition to the package `fr.loria.resedas.moderes.gdmorepository`, are given in the form of Java API documentation provided by the MODERES Java web site **www.jsman.com**. Figure 4 summerize the architecture of this implementation.

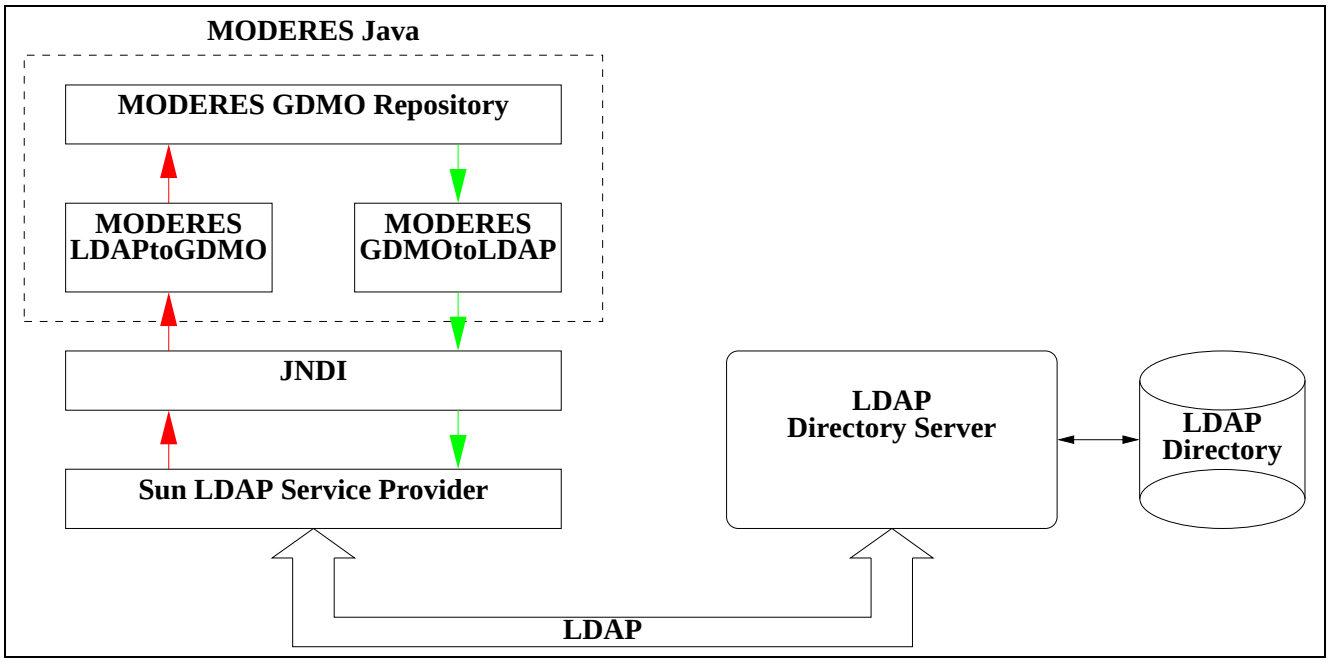


Figure 4: Architecture of the implementation

To run the MISS API, you need to configure your LDAP directory and to install two Java packages, as described in the README file given within the distribution, and summerized below :

- LDAP configuration : you need to install the LDAP object classes and attributes of the schema given in the file `ldap.schema` (See annexe A) ;
- JNDI package (Java Naming and Directory Interface) : delivered as an extension to java API packages, and provides a unified interface for developping java applications for naming and directory services ;
- Sun LDAP Service Provider Interface : delivered by Sun Microsystems, and providing an implementation of the LDAP protocol ;

5.2 The MISS Client API

The MISS (Management Information Schema Service) client API, is intended to be used by management applications to retrieve OSI management information knowledge from an LDAP directory server. Very close in its design to the MIS API written in SML (System Management Language) and provided by the OpenMaster platform (www.evidian.com), this API provides an access to management information schema for java applications.

Two java packages **fr.loria.resedas.miss** and **fr.loria.resedas.moderes.gdmorepository** define the API. For detailed description, users should refer to the java API documentation.

```
1  import fr.loria.resedas.miss.*
2  import fr.loria.resedas.moderes.gdmorepository.*
3public class ClientExample {
4   public static void main(String args[]) {
5     try {
6        // creates a new instance of MISS client
7        MISS_Client client = new MISS_Client();

8        // connects to a LDAP server with specific credentials
9       String url = "ldap://dolcourt.loria.fr:389/o=loria.fr";
10      String uid = "uid=antares,ou=People,o=loria.fr";
11      String passwd = "antares";
12       client.openContext(url,uid,passwd);

13       // try to get a managed object class template definition
14       // with passed arguments the module name and the class name
15      GDMO_ManagedObjectClassTemplate mocTemplate =
16         client.getManagedObjectClass(args[0],args[1]);

17       // prints the class label if found
18      if ( mocTemplate != null )
19        System.out.println(``Class found :''+mocTemplate.getLabel().getLabel());
20      else
21        System.out.println(``Class not found !'');

22       // close the connection
23       client.closeContext();

24    } catch (MISS_Exception e) {  // catch any exception throwed by the MISS Client
25        System.err.println("MISS Exception occured : "+e);
26    }
27  }
28 }
```

Figure 5: A sample code using the MISS API

The **MISS_Client** is the main class of this API. It enables management applications to connect to a LDAP server, and then to retrieve GDMO specifications stored within. When management information knowledge is required, applications should follow the steps given below. For illustration purpose, a sample code is given in figure 5. The various steps are :

1. instanciate a new `MISS_Client` object (figure 5, line 7) ;
2. connect to a LDAP directory server by giving its URL address and user credentials for authentication (figure 5, line 12) ;
3. get the desired informations using one of the getter methods defined by the `MISS_Client` class (figure 5, line 15 and 16) ;
4. close the connection (figure 5, line 23).

All methods defined by the `MISS_Client` class may throw a `MISS_Exception` when accessing the directory. As it is a generic exception, applications should use `getType()` of catched exception object, to distinguish between different errors in the origin of the thrown exception. Details of the `MISS Client API` are given in the form of `Java API documentation` provided with the MISS package.

6 Conclusion

LDAP directory services provides a simple and powerfull way for storing and sharing data between applications. In network and service management domain, both DMTF and IETF are leading working groups on directory-based management applications. These major efforts are the DEN approach and the policy-based management framework, and both of them use LDAP directory services to store and retrieve CIM based management informations.

In this document we have presented our LDAP-based approach for the definition of an OSI based management service. This service called Management Information Schema Service (MISS) uses LDAP directories to store and retrieve OSI management information knowledge formalized in GDMO. This is done, by the definition of new set of LDAP object classes and attributes that maps the GDMO meta constructs to a LDAP schema, so an accurate description of GDMO specifications can now be stored in the directory. Based on this mappings, we also developped a java API called MISS-API that can be used by java management applications to retrieve those descriptions from the directory.

As a futur work and natural extension of this study, we will define a LDAP schema enabling the storage of ASN.1 definitions referenced by GDMO templates (syntax definitions referenced by attribute, notification and action templates). This will result to a complete MISS service for storage and retrieval of OSI management information knowledge.

References

[1] L. Andrey, O. Festor, E. Nataf, A. Schaff, and S. Tata. Validation de bases d'information de gestion: expérience multi-fdt sur un modèle de gestion d'interconnexion de commutateurs. *Technique et Science Informatique*, 16(6), June 1997. Méthodes Formelles: validation de systèmes complexes.

[2] CCITT.X.722. Technologies de l'Information - Interconnexion de Systèmes Ouverts - Structure des Informations de Gestion: Directives por la Définition des Objets Gérés , 1992.

[3] O. Festor. MODE-PP HTML: A GDMO/GRM to HTML translator -Release 1.0- Reference Manual. Technical Report 0199, INRIA Lorraine, 1996.

[4] O. Festor. MODERES Java: Architecture and Core Packages. Technical Report RT-0205, INRIA, May 1997.

[5] O. Festor. The GDMO and GRM Modules Semantic Checker of the MODERES Java Toolkit. Technical Report RT-0208, INRIA, July 1997.

[6] O. Festor, E. Nataf, and L. Andrey. MODE-FE: A GRM/GDMO Parser and its API -Release 1.0- Reference Manual. Technical Report 0190, INRIA Lorraine, 1996.

[7] T. Howes. The String Representation of LDAP Search Filters. Technical report, Décembre 1997. RFC2254.

[8] T. Howes and M. Smith. The LDAP URL Format. Technical report, Décembre 1997. RFC2255.

[9] ISO-10165.4. Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects, 1992.

[10] ISO-8824.1. Abstract Syntax Notation Number One (ASN.1) - Part 1: Specification of Basic Notation, 1992.

[11] M. Wahl. A Summery of the X.500(96) User Schema for use with LDAPv3. Technical report, Décembre 1997. RFC2256.

[12] M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3). Technical report, Décembre 1997. RFC2251.

[13] M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3) : Attribute Syntax Definition. Technical report, Décembre 1997. RFC2252.

[14] M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3) : UTF-8 String Representation of Distinguished Names. Technical report, Décembre 1997. RFC2253.

7 Glossary

- **ASN.1 :** Abstract Syntax Notation One.
- **CIM :** Common Information Model.
- **CMIP :** Common Management Information Protocol.
- **CMIS :** Common Management Information Services.
- **DEN :** Directory Enabled Networks.
- **DIT :** Directory Information Tree.
- **GDMO :** Guidelines for the Definition of Managed Objects.
- **JNDI :** Java Naming and Directory Interface.
- **LDAP :** Lightweight Directory Access Protocol.
- **LDIF :** LDAP Data Interchange Format.
- **MIB :** Management Information Base.
- **MIS :** Management Information Schema.
- **MISS :** Management Information Schema Services.
- **SML :** System Management Language.
- **UML :** Unified Modeling Language.
- **WBEM :** Web-Based Enterprise Mangement.

A Annexe : LDAP Schema for GDMO definitions

A.1 Attribute definitions

```
attributeType ( gdmo-superiorclass-oid NAME 'gdmo-superiorclass'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-registrationid-oid NAME 'gdmo-registrationid'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE  )

attributeType ( gdmo-condition-oid NAME 'gdmo-condition'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-matchingrulelist-oid NAME 'gdmo-matchingrulelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-notificationinfosyntax-oid NAME 'gdmo-notificationinformationsyntax'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-attributelabel-oid NAME 'gdmo-attributelabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-deletemodifier-oid NAME 'gdmo-deletemodifier'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-actioninformationsyntax-oid NAME 'gdmo-actioninformationsyntax'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-originalmodulefile-oid NAME 'gdmo-originalmodulefile'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE  )
```

```
attributeType ( gdmo-actionlabel-oid NAME 'gdmo-actionlabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-attributesuperior-oid NAME 'gdmo-attributesuperior'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-propertylist-oid NAME 'gdmo-propertylist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-subclasslist-oid NAME 'gdmo-subclasslist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-attributegroupdescription-oid NAME 'gdmo-attributegroupdescription'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-superiorsubclasssupport-oid NAME 'gdmo-superiorsubclasssupport'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-originalmodulename-oid NAME 'gdmo-originalmodulename'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-fieldattributelist-oid NAME 'gdmo-fieldattributelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-superiorclasslist-oid NAME 'gdmo-superiorclasslist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-createmodifier-oid NAME 'gdmo-createmodifier'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-attributesyntax-oid NAME 'gdmo-attributesyntax'
```

```
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-attributegroupmode-oid NAME 'gdmo-attributegroupmode'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-derivedfromlist-oid NAME 'gdmo-derivedfromlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-subordinatesubclasssupport-oid NAME 'gdmo-subordinatesubclasssupport'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-additionalattributelist-oid NAME 'gdmo-additionalattributelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-label-oid NAME 'gdmo-label'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-createparameterlist-oid NAME 'gdmo-createparameterlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-attributelist-oid NAME 'gdmo-attributelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-actionreplysyntax-oid NAME 'gdmo-actionreplysyntax'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-parameterattribute-oid NAME 'gdmo-parameterattribute'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-packagelist-oid NAME 'gdmo-packagelist'
        EQUALITY caseIgnoreMatch
```

```
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-parametercontext-oid NAME 'gdmo-parametercontext'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-expandedlabel-oid NAME 'gdmo-expandedlabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-behaviourlist-oid NAME 'gdmo-behaviourlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-subordinateclasslist-oid NAME 'gdmo-subordinateclasslist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-parametersyntax-oid NAME 'gdmo-parametersyntax'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-deleteparameterlist-oid NAME 'gdmo-deleteparameterlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-behaviourdefinition-oid NAME 'gdmo-behaviourdefinition'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-notificationlabel-oid NAME 'gdmo-notificationlabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-attributegrouplabel-oid NAME 'gdmo-attributegrouplabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-actionmode-oid NAME 'gdmo-actionmode'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
```

```
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-modulename-oid NAME 'gdmo-modulename'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-templatetype-oid NAME 'gdmo-templatetype'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-actionlist-oid NAME 'gdmo-actionlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-fieldname-oid NAME 'gdmo-fieldname'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-namingattribute-oid NAME 'gdmo-namingattribute'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-notificationlist-oid NAME 'gdmo-notificationlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-attributegrouplist-oid NAME 'gdmo-attributegrouplist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-expandedclasslabel-oid NAME 'gdmo-expandedclasslabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-superclasslist-oid NAME 'gdmo-superclasslist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-notificationreplysyntax-oid NAME 'gdmo-notificationreplysyntax'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
```

```
        SINGLE-VALUE )

attributeType ( gdmo-parameterlist-oid NAME 'gdmo-parameterlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-classlabel-oid NAME 'gdmo-classlabel'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-conditionalpackagelist-oid NAME 'gdmo-conditionalpackagelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-subordinateclass-oid NAME 'gdmo-subordinateclass'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )


attributeType ( gdmo-attributeidlist-oid NAME 'gdmo-attributeidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-condattributeidlist-oid NAME 'gdmo-condattributeidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-actionidlist-oid NAME 'gdmo-actionidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-subclassidlist-oid NAME 'gdmo-subclassidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-condnotificationidlist-oid NAME 'gdmo-condnotificationidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-condattributelist-oid NAME 'gdmo-condattributelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-condnotificationlist-oid NAME 'gdmo-condnotificationlist'
        EQUALITY caseIgnoreMatch
```

```
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-nbassubordinateidlist-oid NAME 'gdmo-namebindingassubordinateidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-nbassuperiorlist-oid NAME 'gdmo-namebindingassuperiorlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-condactionidlist-oid NAME 'gdmo-condactionidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-notificationidlist-oid NAME 'gdmo-notificationidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-nbassubordinatelist-oid NAME 'gdmo-namebindingassubordinatelist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-nbassuperioridlist-oid NAME 'gdmo-namebindingassuperioridlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-superiorclassidlist-oid NAME 'gdmo-superiorclassidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-subordinateclassidlist-oid NAME 'gdmo-subordinateclassidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-superclassidlist-oid NAME 'gdmo-superclassidlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-condactionlist-oid NAME 'gdmo-condactionlist'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributeType ( gdmo-defaultvalue-oid NAME 'gdmo-defaultvalue'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
```

```
        SINGLE-VALUE )

attributeType ( gdmo-initialvalue-oid NAME 'gdmo-initialvalue'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-permittedvalues-oid NAME 'gdmo-permittedvalues'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )

attributeType ( gdmo-requiredvalues-oid NAME 'gdmo-requiredvalues'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )
```

A.2 Object class definitons

```
objectClass ( gdmo-element-oid NAME 'gdmo-element'
        SUP top
        STRUCTURAL )

objectClass ( gdmo-templatelist-oid NAME 'gdmo-templatelist'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-templatetype  )

objectClass ( gdmo-template-oid NAME 'gdmo-template'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-label
        MAY ( gdmo-expandedlabel $ gdmo-registrationid ) )

objectClass ( gdmo-package-oid NAME 'gdmo-package'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-actionlist $ gdmo-attributegrouplist $ gdmo-attributelist $
                gdmo-behaviourlist $ gdmo-notificationlist ) )

objectClass ( gdmo-attribute-oid NAME 'gdmo-attribute'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-attributesuperior $ gdmo-attributesyntax $ gdmo-behaviourlist $
                gdmo-matchingrulelist $ gdmo-parameterlist ) )

objectClass ( gdmo-attributegroup-oid NAME 'gdmo-attributegroup'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-attributegroupdescription $ gdmo-attributegroupmode $
                gdmo-attributelist )  )
```

```
objectClass ( gdmo-action-oid NAME 'gdmo-action'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-actioninformationsyntax $ gdmo-actionmode $ gdmo-actionreplysyntax $
                gdmo-behaviourlist $ gdmo-parameterlist ) )

objectClass ( gdmo-notification-oid NAME 'gdmo-notification'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-behaviourlist $ gdmo-fieldattributelist $
                gdmo-notificationinformationsyntax $ gdmo-notificationreplysyntax $
                gdmo-parameterlist ) )

objectClass ( gdmo-behaviour-oid NAME 'gdmo-behaviour'
        SUP gdmo-template
        STRUCTURAL
        MAY gdmo-behaviourdefinition  )

objectClass ( gdmo-actionandparameters-oid NAME 'gdmo-actionandparameters'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-actionlabel
        MAY ( gdmo-expandedlabel $ gdmo-parameterlist ) )

objectClass ( gdmo-attributeandproperties-oid NAME 'gdmo-attributeandproperties'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-attributelabel
        MAY ( gdmo-expandedlabel $ gdmo-parameterlist $ gdmo-propertylist $
                gdmo-defaultvalue $ gdmo-initialvalue $
                gdmo-permittedvalues $ gdmo-requiredvalues ) )

objectClass ( gdmo-attributegroupelement-oid NAME 'gdmo-attributegroupelement'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-attributegrouplabel
        MAY ( gdmo-additionalattributelist $ gdmo-expandedlabel ) )

objectClass ( gdmo-conditionalpackage-oid NAME 'gdmo-conditionalpackage'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-label
        MAY ( gdmo-condition $ gdmo-expandedlabel ) )

objectClass ( gdmo-fieldattribute-oid NAME 'gdmo-fieldattribute'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-fieldname
        MAY ( gdmo-expandedlabel $ gdmo-label ) )

objectClass ( gdmo-namebinding-oid NAME 'gdmo-namebinding'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-behaviourlist $ gdmo-createmodifier $ gdmo-createparameterlist $
                gdmo-deletemodifier $ gdmo-deleteparameterlist $ gdmo-namingattribute $
                gdmo-subordinateclass $ gdmo-subordinatesubclasssupport $
                gdmo-superiorclass $ gdmo-superiorsubclasssupport ) )
```

```
objectClass ( gdmo-notificationandparameters-oid NAME 'gdmo-notificationandparameters'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-notificationlabel
        MAY ( gdmo-expandedlabel $ gdmo-parameterlist ) )

objectClass ( gdmo-parameter-oid NAME 'gdmo-parameter'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-behaviourlist $ gdmo-parameterattribute $ gdmo-parametercontext $
                gdmo-parametersyntax )  )

objectClass ( gdmo-managedobjectclass-oid NAME 'gdmo-managedobjectclass'
        SUP gdmo-template
        STRUCTURAL
        MAY ( gdmo-conditionalpackagelist $ gdmo-derivedfromlist $ gdmo-packagelist ) )

objectClass ( gdmo-module-oid NAME 'gdmo-module'
        SUP gdmo-element
        STRUCTURAL
        MUST gdmo-modulename
        MAY ( gdmo-originalmodulefile $ gdmo-originalmodulename ) )



objectClass ( gdmo-mocsemanticlinks-oid NAME 'gdmo-mocsemanticlinks'
        SUP top
        AUXILIARY
        MAY ( gdmo-actionidlist $ gdmo-actionlist $ gdmo-attributeidlist $
                gdmo-attributelist $ gdmo-condactionidlist $ gdmo-condactionlist $
                gdmo-condattributeidlist $ gdmo-condattributelist $
                gdmo-condnotificationidlist $ gdmo-condnotificationlist $
                gdmo-namebindingassubordinateidlist $ gdmo-namebindingassubordinatelist $
        gdmo-namebindingassuperioridlist $ gdmo-namebindingassuperiorlist $
                gdmo-notificationidlist $ gdmo-notificationlist $
                gdmo-subclassidlist $ gdmo-subclasslist $ gdmo-subordinateclassidlist $
                gdmo-subordinateclasslist $ gdmo-superclassidlist $ gdmo-superclasslist $
                gdmo-superiorclassidlist $ gdmo-superiorclasslist ) )
```

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803