ISO/IEC JTC 1/SC 32 N 1177

Date: 2004-10-25

REPLACES: 32N1108

ISO/IEC JTC 1/SC 32

Data Management and Interchange

Secretariat: United States of America (ANSI)
Administered by Pacific Northwest National Laboratory on behalf of ANSI

DOCUMENT TYPE	Working Draft Text (for information or comment)
TITLE	Second ISO/IEC WD 19763-3 Framework for Metamodel Interoperability-- Part-3 : Metamodel Framework for Ontology Registry
SOURCE	Editor - OKABE, Masao
PROJECT NUMBER	1.32.22.01.03.00
STATUS	For comment and review.
REFERENCES	
ACTION ID.	COM
REQUESTED ACTION	Please review for the WG 2 meeting
DUE DATE	
Number of Pages	34
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32

Pacific Northwest National Laboratory *, 13667 Legacy Circle Apt H, Herndon, VA, 20171, United States of America

Telephone: +1 202-566-2126; Facsimile; +1 202-566-1639; E-mail: MannD@battelle.org

available from the JTC 1/SC 32 WebSite http://www.jtc1sc32.org/

*Pacific Northwest National Laboratory (PNL) administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

ISO/IEC JTC 1/SC 32 WG2

Date: 2004-10-22

ISO/IEC WD 19763-3:2004(E)-2

ISO/IEC JTC 1/SC 32/WG 2

Secretariat: ANSI

Information Technology – Framework for Metamodel Interoperability-- Part-3 : Metamodel Framework for Ontology Registry

Composed By

SKLSE, Wuhan University, P.R.China

OKABE, Masao

Document type: International Standard
Document subtype:
Document stage: (50) WorkingDraft
Document language: E

Contents

Figures and tables

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IECWD 19763 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19763 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 32, *Data Management and Interchange*.

ISO/IEC 19763 consists of the following parts, under the general title *Information technology* — **Framework for Metamodel Interoperability**:

— Part 1: Reference Model

— Part 2: Core Model

— Part 3: Ontology Registry

— Part 4: Mapping

Introduction

Today, in the EB(E-Business) or EC(E-Commerce) through the internet, the effective interchange of business transactions or other related information across countries and cultures became the first concerns for people in both IT industry and other non-IT industries.

To follow the current trends of EB or EC, industrial consortia have been in charge of standardization of domain specific business objects including business process models and software components using common modeling facilities and exchanging facilities such as UML and XML. They are very active to standardize domain specific business process models and standard modeling constructs such as data elements, entity profiles and value domains.

Moreover, the interoperations among the autonomous web based applications such as Web services are becoming important. For that purpose, ontology is a key issue. Ontology is a formal expression of the universe of discourse. A lexicon, a taxonomy, a thesaurus and a usual conceptual model such as a business process model by UML are all examples of ontology. In addition ontology includes a more complex axiomatic theory.

The efforts to standardize the metamodel of ontology described in specific languages are taken by OMG. In addition to that, to promote the ontology-based interoperations, a unified framework for registering administrative information of ontology is indispensable.

This part of ISO/IEC 19763 intends to provide a unified framework for registering administrative information of ontology, based on the ISO/IEC 19763-2 Framework for Metamodel Interoperability Part-2 Core Model and using the existing standardized metamodel of ontology described in specific languages.

Information Technology–Framework for Metamodel interoperability –Part 3:Metamodel Framework for Ontology Registry

1 Scope

The primary purpose of the multipart standard ISO/IEC 19763 is to specify the framework for metamodel interoperability. This part of ISO/IEC 19763 specifies the metamodel that provides a facility to register administrative information of ontology.

The metamodel that it specifies intends to promote the interoperations among application systems.

It does not specify the metamodels of ontology described in specific languages and the mapping among them.

Figure-1 shows the scope of this part of ISO/IEC 19763.

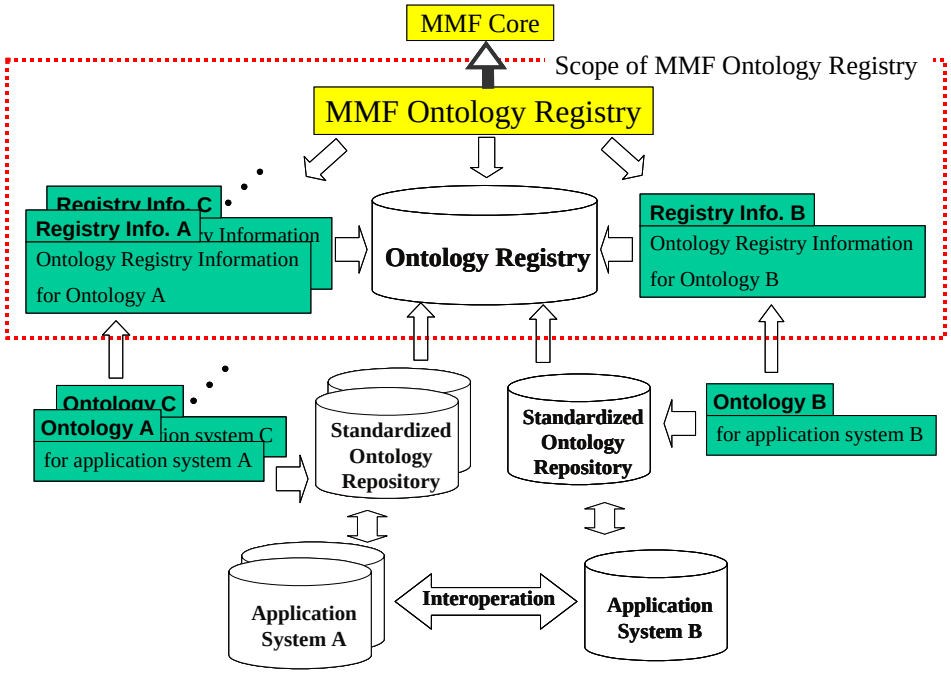

Figure 1 – Scope of MMF Ontology Registry

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11179-3, Information technology – Metadata registries (MDR) - Part 3 : Registry metamodel

ISO/IEC 19763 (all parts), Information technology – Framework for Metamodel Interoperability
ISO/IEC 19501-1:2002, Information technology – Unified Modeling Language (UML) – Part 1: Specification

3 Definitions and abbreviated terms

3.1 Definitions

For the purposes of this International Standard the following definitions apply.

The definitions provided in ISO/IEC 11179-3, ISO/IEC 19763 (all parts), ISO/IEC 19501-1:2002 shall apply to this International Standard.

3.2 Abbreviated terms

3.2.1 MMF Core
ISO/IEC 19763-2, Information technology – Framework for Metamodel Interoperability – Part-2 : Core Model

3.2.2 MMF Ontology Registry
ISO/IEC 19763-3, Information technology – Framework for Metamodel Interoperability – Part-3 : Ontology Registry

3.2.3 MDR
ISO/IEC 11179-3, Information technology – Metadata registries (MDR) - Part 3 : Registry metamodel

3.2.4 OWL
OWL Web Ontology Language

3.2.5 TM
ISO/IEC 13250, Information Technology-Topic Maps

3.2.6 RDFs
RDF Vocabulary Description Language 1.0: RDF Schema

4 Structure of MMF Ontology Registry

4.1 Overview of MMF Ontology Registry

MMF Ontology Registry provides the administrative information concerning ontology registry. Figure2 shows the whole metamodel of MMF Ontology Registry.

Onto_Domain indicates the origination of ontology, which can be designated by one or more Source_Onto**.** **Source_Onto** consists of Onto_Constructs. **Onto_Instance** designates the ontology model that is registered. Onto_Instance has two subclasses: SOC_Variant and Local_Onto. **SOC_Variant** (Variant of Component of Source_Onto) designates the result of transformation on some part of Source_Onto and refers SO_Component. **SO_Component**(Component of Source_Onto) consists of Atomic_Onto_Constructs. **Local_Onto** is composed of several SOC_Variants that come from different Source_Ontos. **Onto_Selection** designates a selection from the Onto_Instance used to build an ontology for users' requirement. **Onto_Concept** is concepts in Source_Onto

corresponding to the ones in Onto_Selection.

Onto_Classifier designates an ontology descriptive language. **Onto_Cosntruct** desigantes the constructs used to build ontology models. Onto_Construct has two subclass: Atomic_Onto_Construct and Evolution_Info. **Atomic_Onto_Construct** is the basic constructors to describe ontology. **Evolution-Info** designates the transformations and compositions on the ancestors of ontology.

The exact specification of each metaclass is given in 4.3

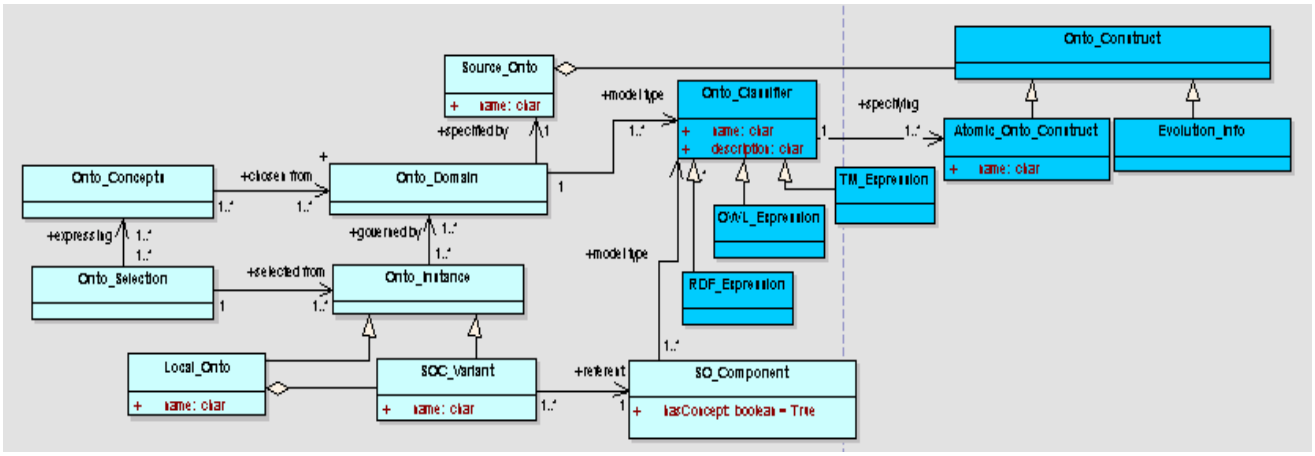

Figure 2 - Metamodel of MMF Ontology Registry

4.2 Relationship between MMF Core and MMF Ontology Registry

MMF Ontology Registry inherits the basic structure from MMF Core. Figure 3 shows the relationship between MMF Core and MMF Ontology.

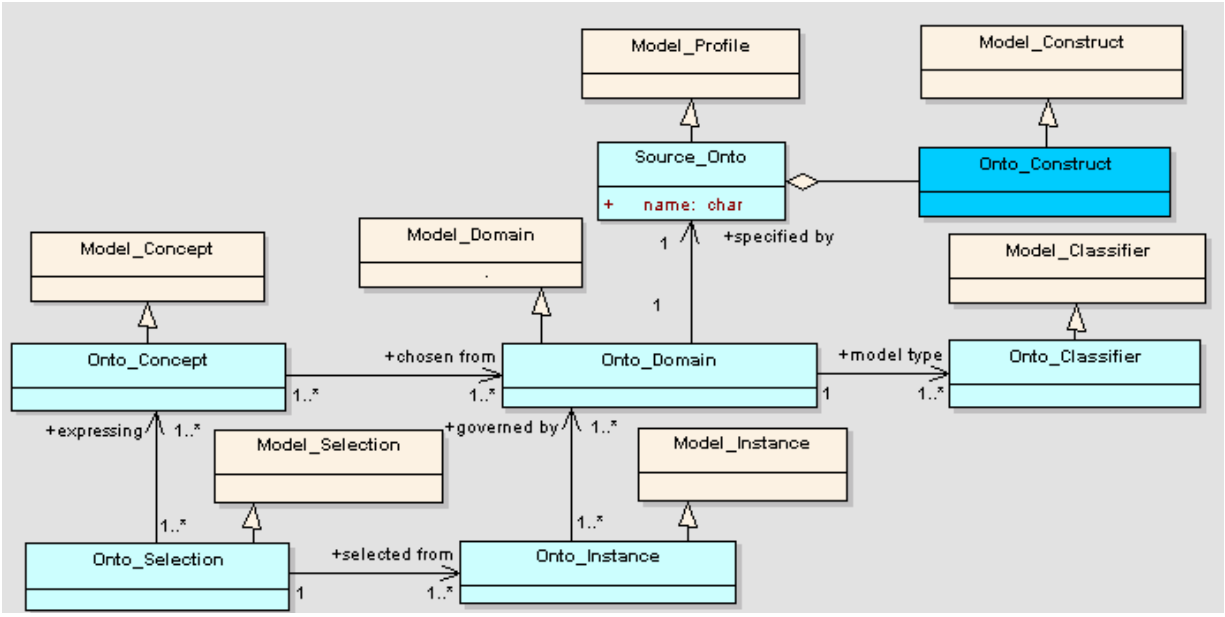

Figure 3 - Relationship between MMF Core and MMF Ontology Registry

4.3 MMF Ontology Registry

4.3.1 Onto_Domain

Onto_Domain	Onto_Domain is a metaclass indicating the origination of Onto_Instance. Onto_Domain is specified by Source_Onto. The model type of Onto_Domain is indicated by Onto_Classifier. A Namespace is designated to hold the names of all the concepts in the Onto_Domain.The names are unique in the namespace. **MMFCore relationship:** Onto_Domain inherits from Model_Domain in MMFCore.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	
specifiedBy	*1..**	*Source_Onto*	*The content of the Onto_Domain is specified by Source_Onto.*
modelType	*1..**	*Onto_Classifier*	*The model type of the Onto_Domain.*
Constraint			

4.3.2 Onto_Instance

Onto_Instance	Onto_Instance is a metaclass designating all the ontology models originating from the Onto_Domain. It is an abstract class and it has two subclasses Local_Onto and SOC_Variant. **MMFCore relationship:** Onto_Instance inherits from Model_Instance in MMFCore.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	

governedBy	*1..1*	*Onto_Domain*	
Constraints			

4.3.3 Onto_Selection

Onto_Selection	Onto_Selection is a metaclass designating a selection from the Onto_Instance corresponding to Onto_Concept. The reference “selected from”indicates the Onto_Instance which provides candidates for this selection. The reference”expressing” indicates the concepts expressed by this selected ontology model. **MMFCore relationship:** Onto_Selection inherits from Model_Selection in MMFCore.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	
expressing	*1..**	*Onto_Concept*	
selectedFrom	*1..1*	*Onto_Instance*	
Constraints			

4.3.4 Onto_Concept

Onto_Concept	Onto_Concept is a metaclass designating the concepts expressed by Onto_Selection. The reference ”chosenFrom” indicates the Onto_Domain that provides specification of these concepts. The reference “sign” indicates the names of these concepts.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	
sign	*1..**	*Named_Element(from Core)*	
chosenFrom	*1..1*	*Onto_Domain*	
Constraints			

4.3.5 Onto_Classifier

Onto_Classifier	Onto_Classifier is a metaclass designating the model type of Onto_Domain and SO_Component. It is an abstract class and it's the superclass of the expressions in the concrete ontology languages, such as OWL and TM.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*String*	
description	*1..**	*String*	
specifying	*1..**	*Atomic_Onto_Construct*	
Constraints			

4.3.6 Onto_Construct

Onto_Construct	Onto_Construct is a metaclass designating the constructs used to build ontology models. It is an abstract class and it has two subclasses: Atomic_Onto_Construct and Evolution_Info. **MMFCore relationship:** Onto_Construct inherits from Model_Construct from MMFCore.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
Constraints			

4.3.7 Source_Onto

Source_Onto	Source_Onto is a metaclass designating the source of the Onto_Instances. Transformation or compostion are exerted on some parts of Source_Onto to produce Onto_Instance. Source_Onto is a comparative concept. Generally speaking, the initiative Source_Ontos are constructed according to some widely accepted domain specifications. After registration,

	Local_Onto can be treated as Source_Onto for its descendents. Source_Onto consists of Onto_Construct. **MMFCore relationship:** Source_Onto inherits from Model_Profile in MMFCore.

Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	
name	*1..1*	*String*	
consistsOf	*1..**	*Onto_Construct*	
Constraints			

4.3.8 Atomic_Onto_Construct

Atomic_Onto_Construct	Atomic_Onto_Construct is a metaclass designating the fundamental constructors of logic.

Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1*	*String*	
Constraints			

4.3.9 SO_Component

SO_Component	SO_Component is a metaclass designating some parts of Source_Onto. SO_Component consists of Atomic_Onto_Construct. The model type of SO_Component is specified by Onto_Classifier.

Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from	*1..1*	*Administration Record (from*	

MDR)		*MDR)*	
hasConcept	*1..1*	*Boolean* *(Default value=true)*	*There must be a t least one concept in SO_Component.*
consistsOf	*1..**	*Atomic_Onto_Construct*	
modelType	*1..1*	*Onto_Classifier*	
Constraints			

4.3.10 Evolution_Info

Evolution_Info	Evolution_Info is a metaclass designating the possible change to the Source_Onto to produce Onto_Instance or the change the Source_Onto has experienced. It is an abstract class and it is superclass of Transformation_Info and Composition_Info.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
Constraints			

4.3.11 OWL_Expression

OWL_Expression	OWL_Expression is a metaclass designating expressions in OWL. It is a subclass of Onto_Pacakage		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*String*	
description		*String*	
Constraints			

4.3.12 TM_Expression

TM_Expression	TM_Expression is a metaclass designating expressions in TM. It is a subclass of Onto_Classifier.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*String*	
description	*1..**	*String*	
Constraints			

4.3.13 RDFs_Expression

RDFs_Expression	RDFs_Expression is a metaclass designating expressions in RDFs. It is a subclass of Onto_ Classifier.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*String*	
description	*1..**	*String*	
Constraints			

4.3.14 SOC_Variant

SOC_Variant	SOC_Variant is a metaclass designating the ontology models that result from the transformation on SO_Component. Transformation_Rules are employed on SO_Component to get SOC_Variant. And the details of transformation process are recorded by Transformation_Info. SOC_Variant is a subclass of Onto_Instance.Hence it is a subclass of Administered_Item(from MDR).		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**

administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	
name	*1..1*	*String*	
referent	*1..1*	*SO_Component*	
aboutTransformation	*1..1*	*Transformation_Info*	
Constraints			

4.3.15 Local_Onto

Local_Onto	Local_Onto is a metaclass designating the ontology models that are composed of SOC_Variants originating from different Source_Ontos. Composition_Rules are employed on SOC_Variants to get Local_Onto. And the details of composition process are recorded by Composition_Info. Local_Onto is a subclass of Onto_Instance.Hence it is a subclass of Administered_Item(from MDR).		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
administration_Record(from MDR)	*1..1*	*Administration Record (from MDR)*	
name	*1..1*	*String*	
consistsOf	*2..**	*SOC_Variant*	
aboutComposition	*1..1*	*Composition_Info*	
Constraints			

Annex A (informative) Example of MMF Ontology Registry

Two ontologies ,SO1 and SO2 ,are taken as an example to show the meaning of the key metaclasses in MMF Ontology Registry.

A.1 Example to show the meaning of Source_Onto.

SO1 and SO2 are two Source_Onto in Onto_Domain 1

Figure 4 shows the meaning of SO1: “There are some buyer, each buyer has a credit. John is a buyer and John has Credit_a as a credit.”

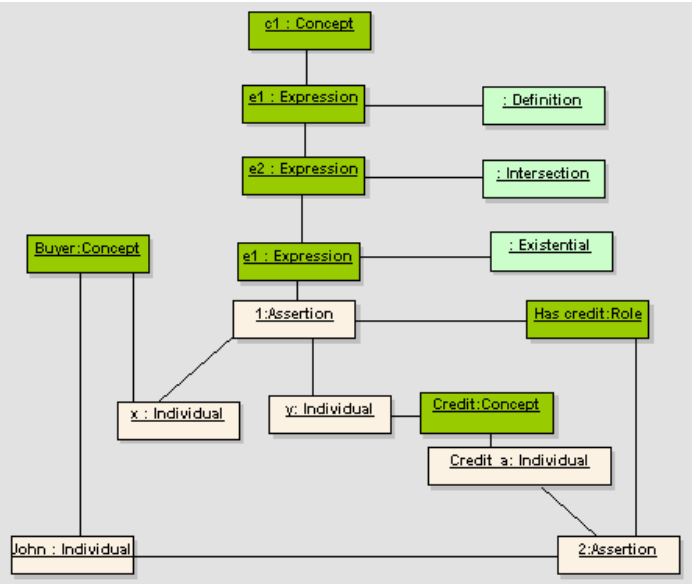

Figure 4 - SO1

Figure5 shows the meaning of SO2: “Tom has a Problem x. Tom sends an email to Jerry “

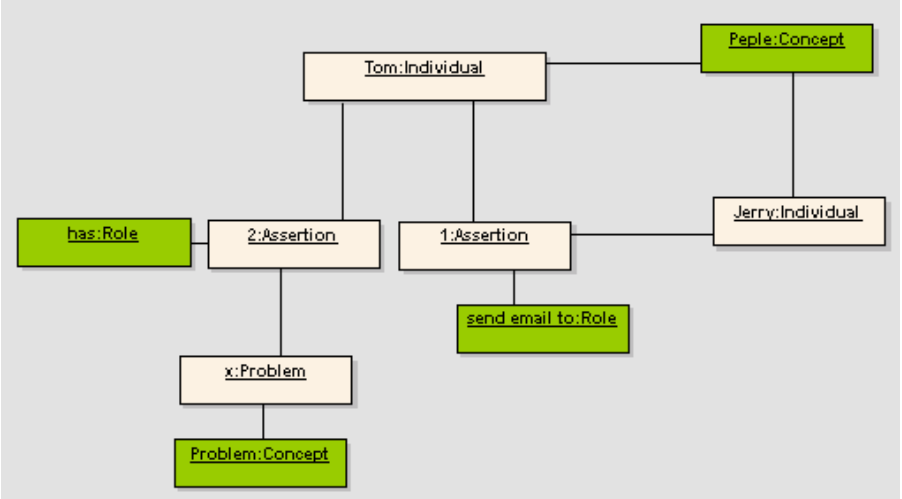

Figure 5- SO2

A.2 Example to show the meaning of Atomic_Onto_Construct

Figure 6 shows all the Atomic_Onto_Constructs of SO1.Each object in the Figure is an Atomic_Onto_Construct.

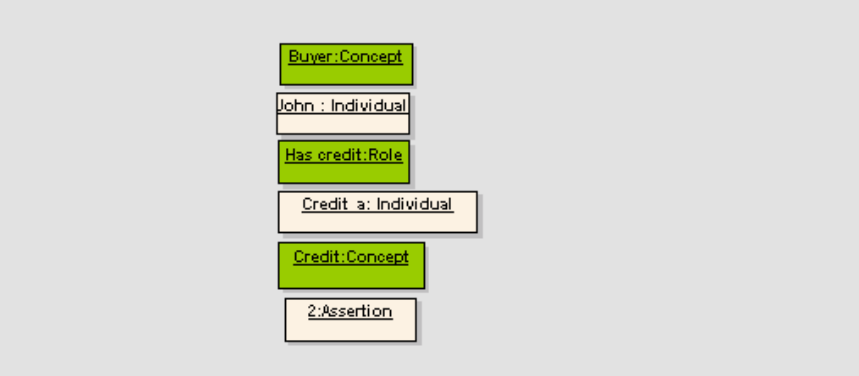

Figure 6 - Atomic_Onto_Construct of SO1

A.3 Example to show the meaning of SO_Component

Figure 7 shows some SO_Components of SO1.

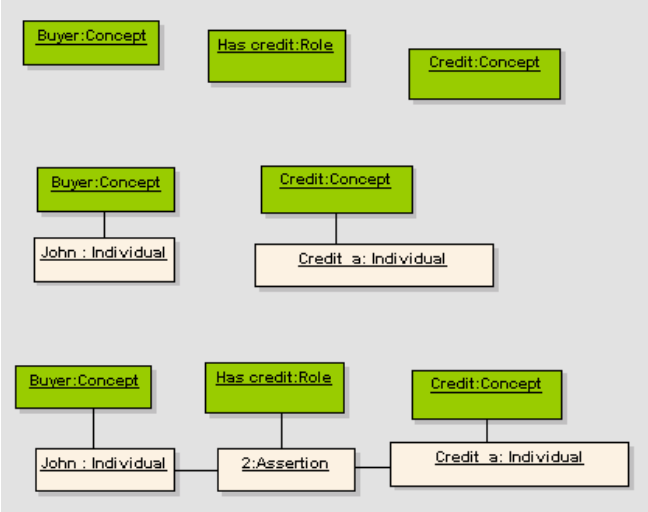

Figure 7 - SO_Component of SO1

A.4 Example to show the meaning of SOC_Variant

Figure 8 shows the two possible SOC_Variants based on SO1.

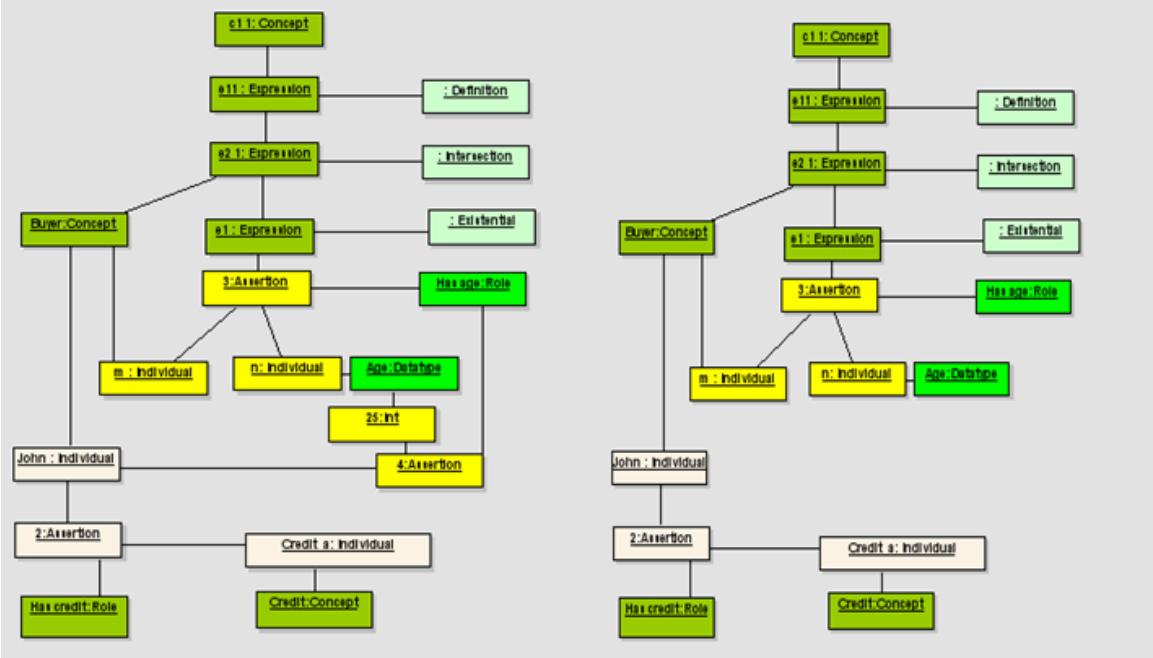

Figure 8 - SOC_Variant of SO1

Based on a SO_Component of SO1,a role and an assertion are added to produce the left SOC_Variant in Figure 8. The meaning of this SOC_Variant is " John is a buyer and John has Credit_a as a credit. Each Buyer has an age and John is of age 26."

Based on another compoent, a role is added to produce the right SOC_Variant in Figure 8. The meaning of this SOC_Variant is" John is a buyer and John has Credit_a as a credit. Each Buyer has an age. "

Figure 9 shows the meaning of a SOC_Variant of SO2: "People live in Place. Tom lives in Wuhan."

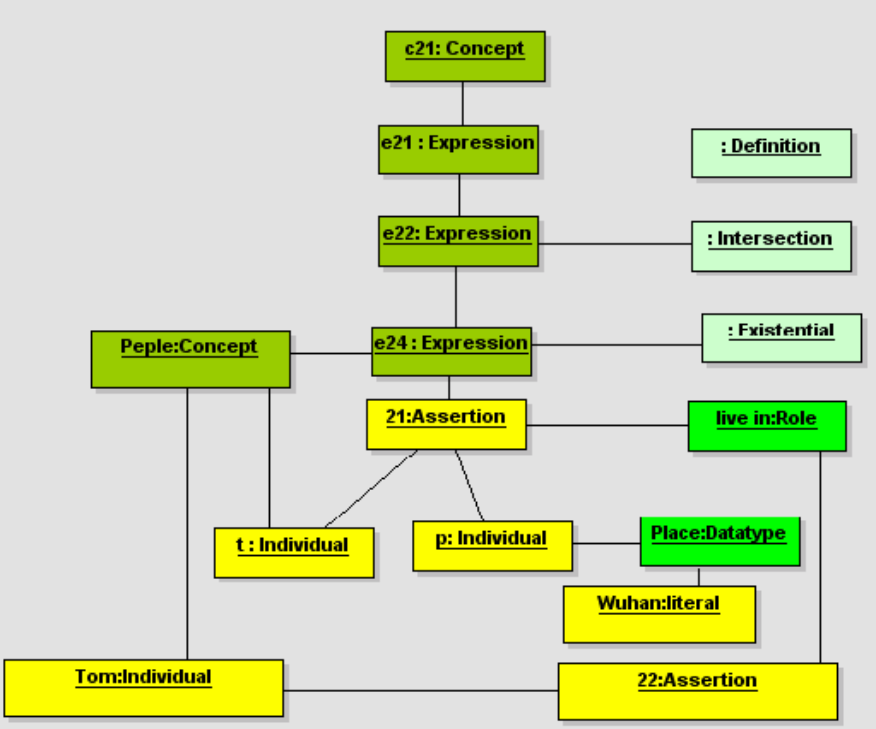

Figure 9 - SOC_Variant of SO2

A.5 Example to show the meaning of Local_Onto

Figure 10 shows the meaning of a Local_Onto that is based on a SOC_Variant of SO1 and one of SO2.
Besides the two SOC_Variants, a Role and an Assertion are added to produce the Local_Onto.
The meaning of this Local_Onto is:" People live in Place. Tom lives in Wuhan. There are some Buyers, each of who has a credit. John is a Buyer and he has a credit Credit_a.Each Buyer has an age. John is of age 26.People use Credit. Tom use credit Credit_for_Tom."

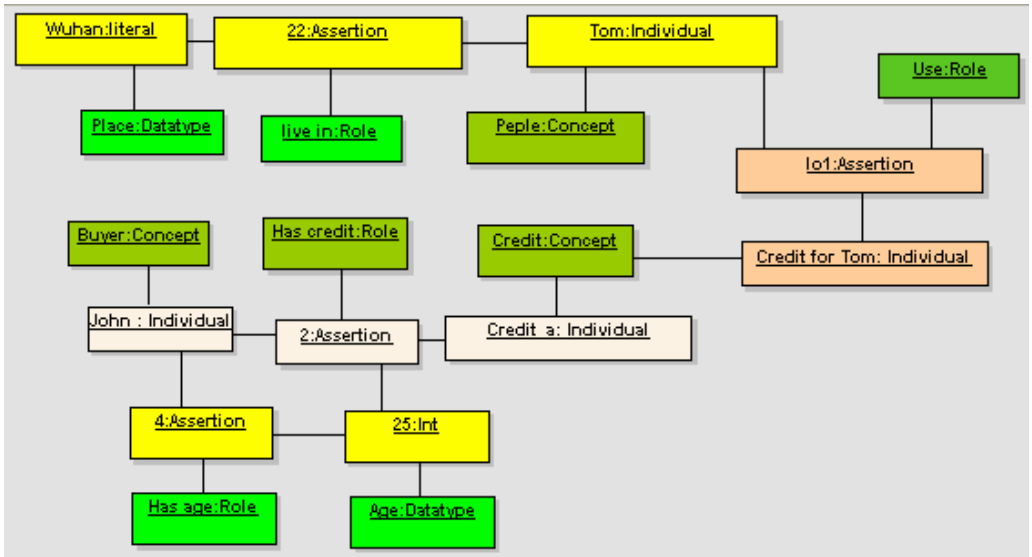

Figure 10 - Local_Onto

Annex B (informative) Other Constructs

This annex shows the whole metaclass of MMF Ontology Registry and explains the informative parts.

B.1 Atomic_Onto_Construct model

Atomic_Onto_Construct is the superclass of ***DL_element***, which has two subclasses, ***ABox_element*** and ***TBox_element***

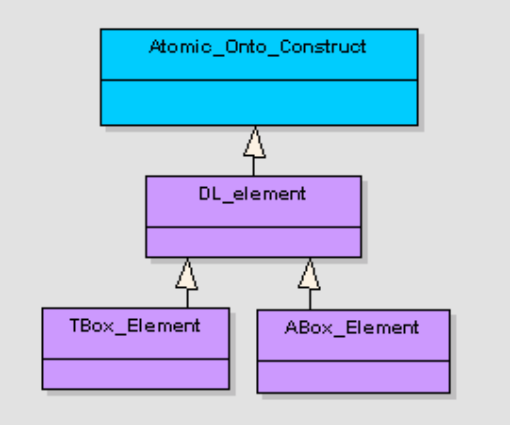

Figure11 - Atomic_Onto_Construct model

DL is the core of ODM. In MMF Ontology Registry, DL_element is subclass of Atomic_Onto_Construct. TBox_element and ABox_element are the subclasses of DL_element. in ODM, Other descriptive languages has some relationship with Onto_Classifier. OWL_Expression, RDFs_Expression,TM_Expression,SCL_Expressions are all the subclasses of Onto_Classifier.

B.1.1 DL_Element

DL_Element	An element that is an abstraction drawn from all the elements of Description Language. A DL Knowledgebase is traditionally divided into three principal parts: · Terminology or schema, the vocabulary of the application domain, called the 'TBox', · Assertions, which are named individuals expressed in terms of the vocabulary, called the 'ABox' and · Description Language that define terms and operators for build expressions.

Attribute or Reference	**Occurrences**	**Datatype**	**Description**

Constraints

B.1.2 TBox_Element

TBox_Element	An element that is an abstraction drawn from all the elements of TBox. A TBox contains all of a DL models terminology. This includes all the sub-classes of Term that are not sub-classes of Instance. TBox_Element is subclass of DL_Element.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
Constraints			

B.1.3 ABox_Element

ABox_Element	An element that is an abstraction drawn from all the elements of ABox. An ABox contains all of a DL models instances. This includes all the sub-classes Instance. ABox_Element is subclass of DL_Element.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
Constraints			

B.2 Rule model

Each ***Atomic_Onto_Construt*** has some rules on evolution and authentification. There are two kinds of Evolution ***Rule***s: ***Transformation_Rule*** and ***Composition_Rule***. SOC_Variant is the result of adopting Transformation_Rules on SO_Component.Composition rules can be used on SOC_Variant from different Source_Onto for constructing Local_Onto. ***Evolution_Rule***s have corresponding ***Verification_ Criterion***s to check semantic consistency when changes happen. Semantic consistency here means that the concept before change should not cause semantic collision with the concept after change. For the SOC_Variant from the same Source_Onto, possible transformation includes changing name, adding new attribute, association and axiom.However, the new name should be unique in namespace, And new attribute,association,axiom can not collide with the existing ones. For the composition of SOC_Variant form different Source_Ontos, the new association between various SOC_Variants should not cause semantic conflict with the existing ones in Source_Onto. After the adding of associations, then we can build new corresponding axioms, which will not collide with the original axiom from Source_Ontos.

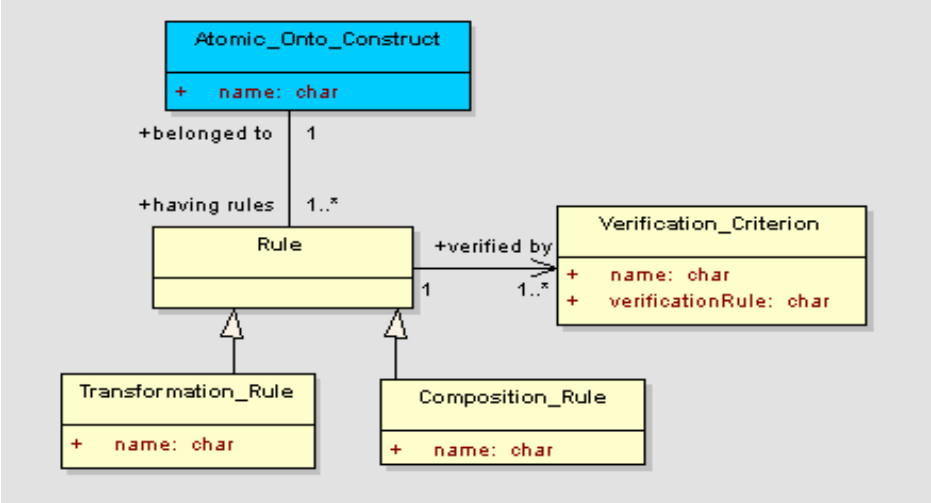

Figure12 - Rule model

B.2.1 Rule

Rule	Rule is an abstraction drawn from all the applicable rules of an Atomic_Onto_Construct.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
Constraints			

B.2.2 Verification_Criterion

Verification_Criterion	Verification_Criterion is a metaclass designating the criterion to judge the correctness of the usage of a transformation rule or a composition rule. Verification_Criterion ensures that the transformation rule will not produce something in conflict with the semantics of Source_Onto,and the composition rule will not produce something in conflict with the semantics of SOC_Variants.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*String*	
verificationRule	*1..1*	*String*	*The rule to predict whether the usage of a transformation rule or a composition rule is*

			adequate.
Constraints: There must be a well-formed structure to define the Verification_Criterion. The structure will be a next-stage task for MMF Ontology Registry. The structure will be the foundation of MMF Ontology Registry, but it will not be a part of MMF Ontology Registry			

B.2.3 Transformation_Rule

Transformation_Rule	The rule used to the transform the SO_Component to get the SO_Variant. With the constraints of the Verification_Criterions, the usage of Transformation_Rule will not produce something in conflict with the semantic of the SO_Component.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*Sting*	
VerifiedBy(from Rule)	*1..**	*Verificaiton_Criterion*	*The verification criterion for the transformation rule.*
Constraints：There must be a well-formed structure to define the Transformation_Rule. This structure will be a next-stage task for MMF Ontology Registry. The structure will be the foundation of MMF Ontology Registry,but it will not be a part of MMF Ontology Registry.			

B.2.4 Composition_Rule

Composition_Rule	The rule used to compose the SOC_Variants to get the Local_Onto. Compostion_Rule has some verification criterions, which ensure the Local Onto has no semantic collision with the SOC_Variants. The validity of the usage of the Compositon_Rule needs the authentification from the Registry Authority.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
name	*1..1*	*String*	
verifiedBy(from Rule)	*1..**	*Verification_Criterion*	*The verification_Criterion*

			used to validate the usage of the Composition_Rule.
Constraints：There must be a well-formed structure to define the Composition_Rule. This structure will be a next-stage task for MMF Ontology Registry. The structure will be the foundation of MMF Ontology Registry, but it will not be a part of MMF Ontology Registry.			

B.3 Evolution_Info model

Evolution-Info shows the compositions and transformations on the ancestor of ontology model. Evolution_Info has two subclasses, ***Transformation_Info*** and ***Composition_Info***. Transformation_Info indicates the transformations on SO_Component for getting Local_Onto. Transformation_Info is consisted of ***TI_Item***s. TI_Item records the use and verification information of each ***Transformation_Rule***. Composition_Info is consisted of ***CI_Item***s. CI_Item records the use and verification information of each ***Composition_Rule***.

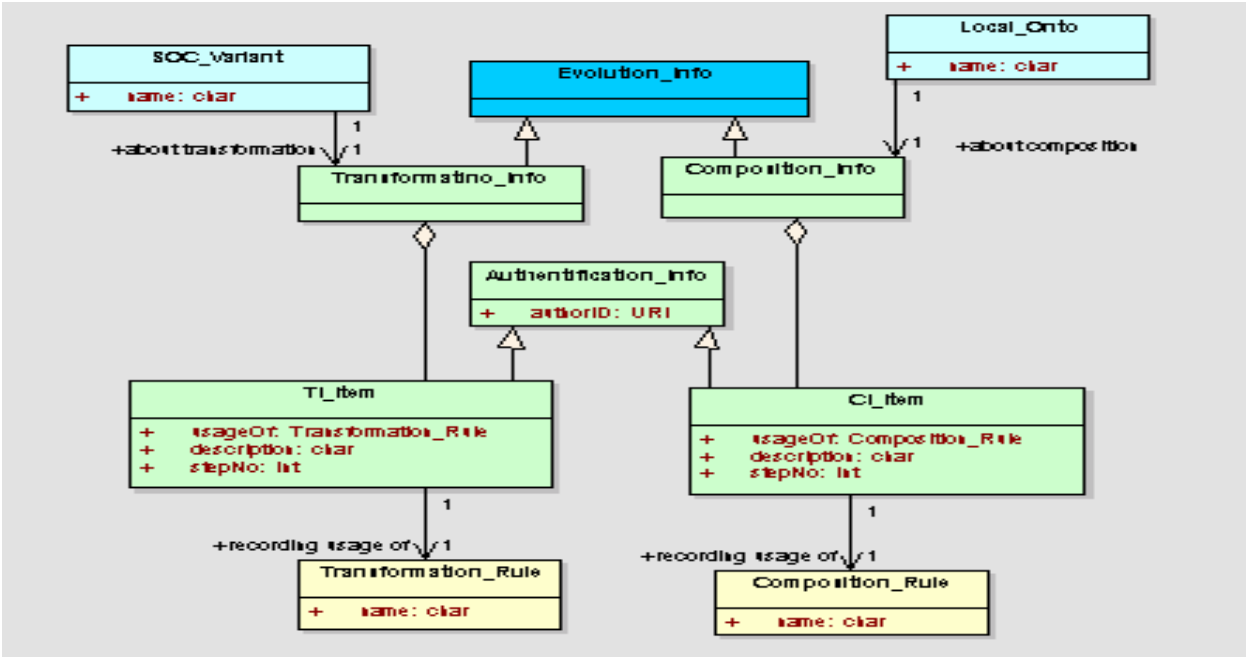

Figure 13 - Evolution_Info model

B.3.1 Authentification_Info

Authentification_Info	Authentification_Info is a metaclass designating the authentification information of the usage of the rules. Authentification_Info is an abstract class and it is superclass of Transfomation_Info and Composition_Info. These two subclasses record the authentification information about the usage of Transformation_Rule and Composition_Rule respectively.

Attribute or Reference	Occurrences	Datatype	Description
authorID	*1..1*	*URI*	
Constraints			

B.3.2 Transformation_Info

Transformation_Info	Transformation_Info is a metaclass designating the information about the usage of Transformation rule on the SO_Component. Transformation_Info consists of Transformation_Info_Item. The authentificaiton info about the usage of each Transformation_Rule is recorded in a Transformation_Info_Item.

Attribute or Reference	Occurrences	Datatype	Description
numOfStepsInTransformation	*1..1*	*Integer*	*The number of the steps in the transformation process.*
involving	*1..1*	*SO_Component*	
resulting	*1..1*	*SOC_Variant*	
consistsOf	*1..**	*TI_Item*	
Constraints			

B.3.3 Composition_Info

Composition_Info	Composition_Info is a metaclass designating the information about the usage of composition rules to get the Local_Onto. Compostion_Info consists of Composition_Info_Item, which records the usage of each Composition_Rule. The number of the composition steps is kept in the Compositon_Info.

Attribute or Reference	Occurrences	Datatype	Description
numOfStepsInComposition	*1..1*	*Integer*	*The number of the steps in the composition process.*

consistsOf	*1..**	*CI_Item*	*The Items of the Composition_Info.*
Involving	*2..**	*SOC_Variant*	
resulting	*1..1*	*Local_Onto*	
Constraints			

B.3.4 TI_Item

TI_Item	The basic unit of Transformation_Info. It records the authentification inforamtion of the usage of a Transformation_Rule. The identifier of the authentification authority,The Transformation_Rule, the Atomic_of_Onto_Construct and the change on it will be recorded in a Transformation_Info_Item.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**
authorityID(from Authentification_Info	*1..1*	*URI*	*The authority to authenticate the adequacy of the transformation.*
usageOf	*1..1*	*Transformation_Rule*	*The transformation rule used in this step.*
description	*1..1*	*String*	
stepNo	*1..1*	*Integer*	*The precedence number of the step.*
Constraints: There must be a well-formed structure to describe the Transformation_Rule. In accordance with it, some attributes of TI_Item such as *descriptionOfTransformation* will change the datatype.			

B.3.5 CI_Item

CI_Item	The basic unit of Composition_Info. It records the authentification information of the usage of a Composition_Rule. The identifier of the Authentification Authority, the Composition_Rule used, the LO_Components involved will be recorded in a Compostion_Info_Item.		
Attribute or Reference	**Occurrences**	**Datatype**	**Description**

authority_ID(from Authentification_Info	*1..1*	*URI*	*The authority to authenticate the adequacy of the composition.*
stepNo	*1..1*	*Integer*	*The precedence number of the step.*
usageOf	*1..1*	*Composition_Rule*	*The composition rule used in this step.*
description	*1..1*	*String*	
Constraints: There must be a well-formed structure to describe the Composition_Rule. In accordance with it, some attributes of CI_Item such as and *descriptionOfComposition* will change the datatype.			

B.4 Administered_Item model

This figure shows the all Administered_Items as a summary.

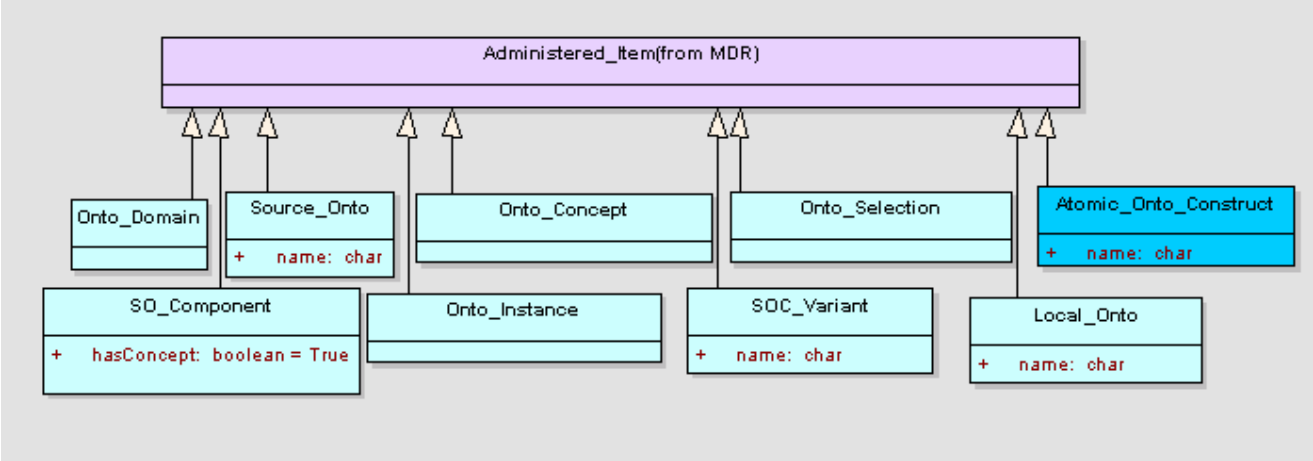

Figure 14 - Administered_Item model

Annex C (informative) Examples of Registry Information

SO1, SO2, Loal_Onto and the other artifacts in Annex B are taken as an example to show how to register them with MMF Ontology Registry.

C.1 Registry information of Onto_Domain

SO1 and SO2 are two Source_Onto in Onto_Domain1.Table 1 shows the registry information of OntoDomain1.

Table 1 - Registry information of an Onto_Domain

Onto_Domain 1	
Attribute or Reference	**Occurrences Datatype**
administration_Record(from MDR)	*# Onto_Domain 1*
specifiedBy	*SO1*
	SO2
modelType	*OWL Expression*

C.2 Registry information of Source_Onto

Table 2 shows the registry information of SO1

Table 2 - Registry information of a Source Onto

SO1	
Attribute or Reference	**Value**
administration_Record(from MDR)	*#SO1*
name	*SO1*
consistsOf	Buyer:Concept
	John:Indivdual
	hasCredit:Role
	Credit_a:Individual
	Credit:Concept
	2:Assertion

C.3 Registry information of Atomic_Onto_Construct

Table 3 shows the registry info of an Atomic_Onto_Construct of SO1

Table 3 - Registry information of an Atomic_Onto_Construct

Buyer:Concept	
Attribute or Reference	**Value**
administration_Record(from MDR)	#
name	*Buyer:Concept*
havingRule	*#Rule 1*
	#Rule 2
	#Rule 3
	#Rule 4
	#Rule 5

C.4 Registry information of SO_Component

Table 4 shows the registry information of a SO_Component of SO1

Table 4 - Registry information of a SO_Component

SO_Component 1 of SO1	
Attribute or Reference	**Value**
administration_Record(from MDR)	#
hasConcept	*True*
ConsitsOf	*Buyer:Concept(from SO1)*
	HasCrecit:Role(from SO1)
	Credit:Concept(from SO1)
	John:Individual(from SO1)
	Credit_a:Individual(from SO1)
modelType	*OWL Expression*

C.5 Registry information of SOC_Variant

Table 5 shows the registry info of a SOC_Variant in Figure 8.

Tabel 5 - Registry information of a SOC_Variant

SOC_Variant 1 of SO1	
Attribute or Reference	**Value**
administration_Record(from MDR)	#
name	*SOC_Variant 1 of SO1*
referent	*SO_Component 1 of SO1*
aboutTransformation	# *Transformation Info 1*

C.6 Registry information of Local_Onto

Table 6 shows the registry info of the Local_Onto in Figure 10.

Table 6 - Registry information of a Local_Onto

Local_Onto 1	
Attribute or Reference	**Value**
administration_Record(from MDR)	#
name	*Local_Onto 1*
consistsOf	*SOC_Variant 1 of SO1*
	SOC_Variant 1 of SO1
aboutCompostion	# *Composition_Info 1*