NGSI Context Management

Candidate Version 1.0 – 03 Aug 2010

Open Mobile Alliance

OMA-TS-NGSI_Context_Management-V1_0-20100803-C

Contents

Figures

Tables

No table of figures entries found.

1. Scope

This document is the Technical Specification for the NGSI-9 and NGSI-10 interfaces. It specifies the context management functions of the NGSI Enabler.

NGSI v1.0 defines abstract interfaces. This technical specification builds the basis for the definition of binding technologies based on the abstract interface definitions given in the TS.

The TS is not basis for testing without a respective binding.

2. References

2.1 Normative References

[NGSI-AD] "NGSI Architecture", Open Mobile Alliance™, OMA-AD-NGSI-V1_0, URL:http://www.openmobilealliance.org/

[NGSI-RD] "NGSI Requirements", Open Mobile Alliance™, OMA-RD-NGSI-V1_0, URL:http://www.openmobilealliance.org/

[GeoShape] "GML 3.1.1 PIDF-LO Shape Application Schema for use by the Internet Engineering Task Force (IETF)", Thomson, M. and C. Reed, Candidate OpenGIS Implementation Specification 06-142r1, Version: 1.0, April 2007. URL: http://www.opengeospatial.org/standards/gml

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC4119] "A Presence-based GEOPRIV Location Object Format", SJ. Peterson, December 2005, URL:http://www.ietf.org/rfc/rfc4119.txt

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[RFC5139] "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", M. Thomson etc, February 2008, URL: http://www.ietf.org/rfc/rfc5139.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[XML-Schema-Part2] "XML Schema Part 2: Datatypes Second Edition" W3C Recommendation 28 October 2004, URL: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html

2.2 Informative References

[LOCSIP] "Location in SIP/IP core Specification" Candidate Version 1.0, OMA, 18 August 2009 http://www.openmobilealliance.org/Technical/release_program/docs/LOCSIP/V1_0-20090818-C/OMA-TS-LOCSIP-V1_0-20090818-C.pdf

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

The definitions of the OMA Dictionary [OMADICT] are valid for this document unless otherwise stated below.

Context Entity	see [NGSI-RD]
Context Information	see [NGSI-RD]
Context Information Model	see [NGSI-RD]
Context Registration	see [NGSI-RD]

3.3 Abbreviations

NGSI	Next Generation Service Interfaces
OMA	Open Mobile Alliance

4. Introduction

The Context Management APIs provides interfaces in order to

- Manage the Context Information about Context Entities, for example the lifetime and quality of information.
- Access (query, subscribe/notify) to the available Context Information about Context Entities.

This TS specification is part of the TS specifications for the NGSI enabler,

4.1 Version 1.0

The NGSI TS Context Management document specifies the NGSI-9 and NGSI-10 Interfaces with the following functions:

- Register and retrieve the availability of Context Entities and/or Context Information.
- Update Context Information in accordance to a specified Context Information Model.
- Query for and subscribe to Context Information about Context Entities.

4.1.1 Backward Compatibility to the Parlay Specifications

None.

5. Technical Details

5.1 Service Description

5.1.1 Overview

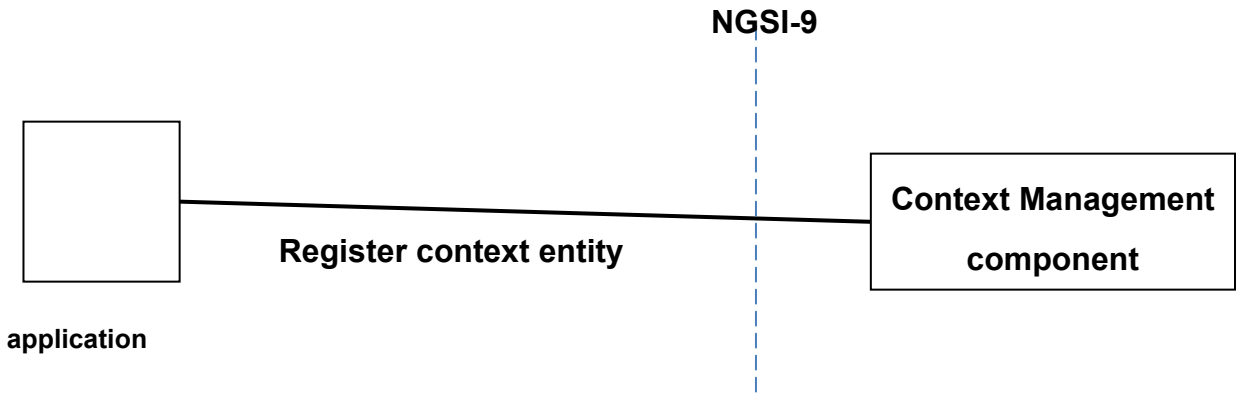
The Context Management component provides the NGSI-9 and NGSI-10 interfaces to manage Context Information about Context Entities. Actual implementation and its internal structure of Context Management component are out of scope.

Through these interfaces, a Context Management component will provide its context management services to actors outside of a single network. These actors can:

- provide Context Information (update operations)
- consume Context Information (query and subscribe/notify operations)
- discover context entities through query or notifications (register and discover operations)

5.1.2 NGSI-9: Context Entity Discovery Interface

5.1.2.1 Register Context Entity Operation



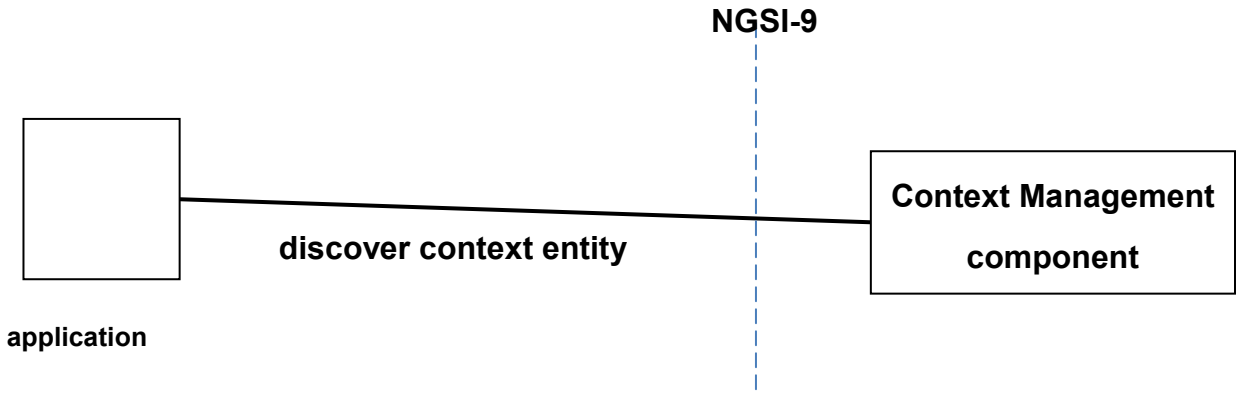
The RegisterContextEntity operation enables the Context Management component to allow registering and updating Context Entities, their attributes and availability.

This interface is used by an application acting as or in behalf of a context provider to register new Context Entities with the Context Management component. The application uses NGSI-9 interface when the new Context Entities become available or when there is an update to the set of attributes for the existing Context Entities. The interface is used to supply Context Entity metadata to the Context Management component including:

- Context entity id
- Context entity attributes and their availability

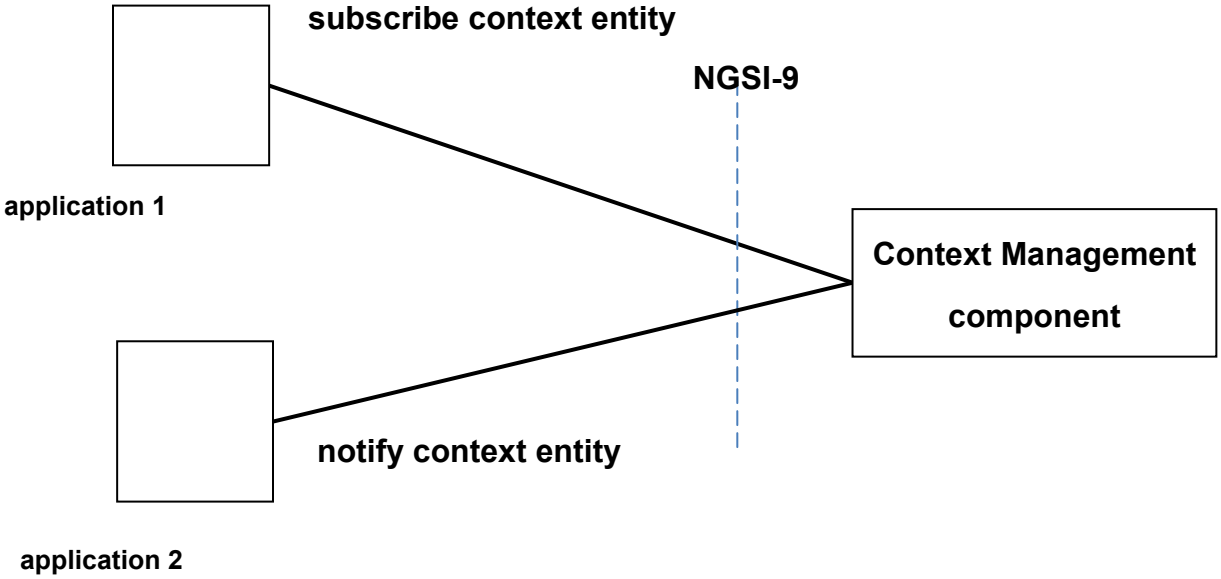
Note that this operation does not handle the values of the attributes.

5.1.2.2 Discover Context Entity Operation



The DiscoverContextEntity operation enables an actor to discover available Context Entities and their attributes.

5.1.2.3 Subscribe and Notify based Context Entity Discovery Operation



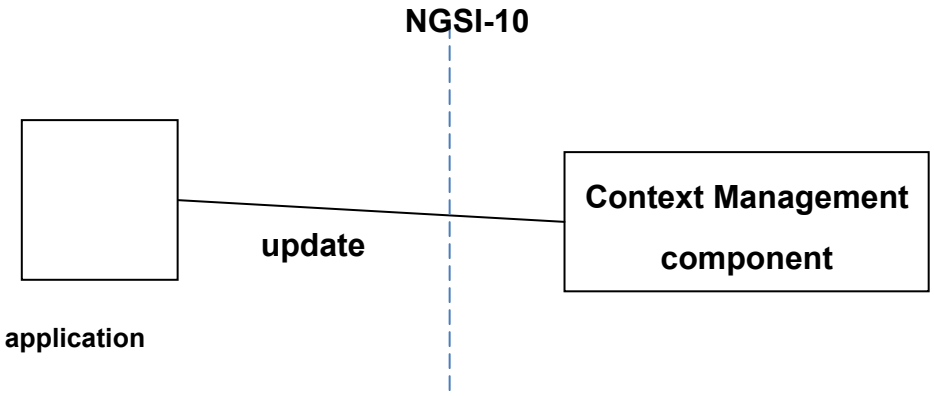
The SubscribeNotifyContextEntity operation enables an application (application 1) to issue a subscription to the Context Management component on behalf of application 2, such that application 2 receives the respective notification upon the availability of new Context Entities or changes to available Context Entities and their attributes. Application 1 can change the subscription or cancel it. As the result of subscription, the Context Management component sends notifications to the receiving application (application 2).

This operation is the asynchronous version of DiscoveryContextEntity operation described in 5.1.3.2.

Note: Application 1 and application 2 could be the same entity.

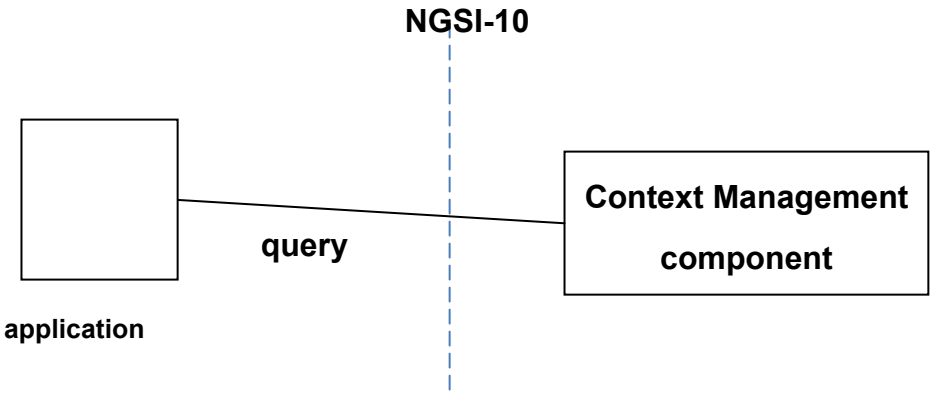
5.1.3 NGSI-10: Context Information Interface

5.1.3.1 Update Context Operation



The update operation enables an application acting as a context producer to provide or update Context Information to the Context Management component.

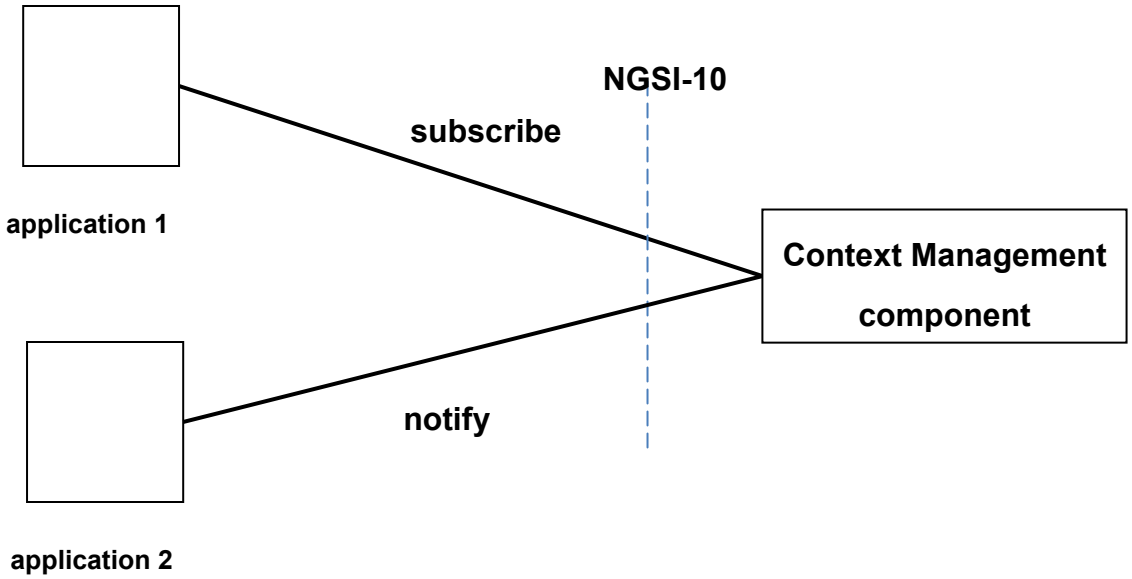
5.1.3.2 Query Operation



The Query operation enables applications acting as context consumers to query for Context Information of

- explicitly listed Context Entities using their Context Entity id
- Context Entities which are specified by patterns of entity id and/or attributes

5.1.3.3 Subscription and Notification Operation



The Subscription operation enables an application (application 1) to issue a subscription to the Context Management component on behalf of application 2, such that application 2 receives the respective notifications for changes of context attribute values. Application 1 can change the subscription or cancel it. As the result of subscription, the Context Management component sends notifications to the receiving application (applicationr 2).

Note: Application 1 and application 2 could be the same actor.

5.2 Context Information Model

Context Entities are entities that are described by Context Information. Figure 1**: Examples of Context Entities**

gives some examples on entities that can be used as Context Entities. Context Entities are described by the Context Information Model.

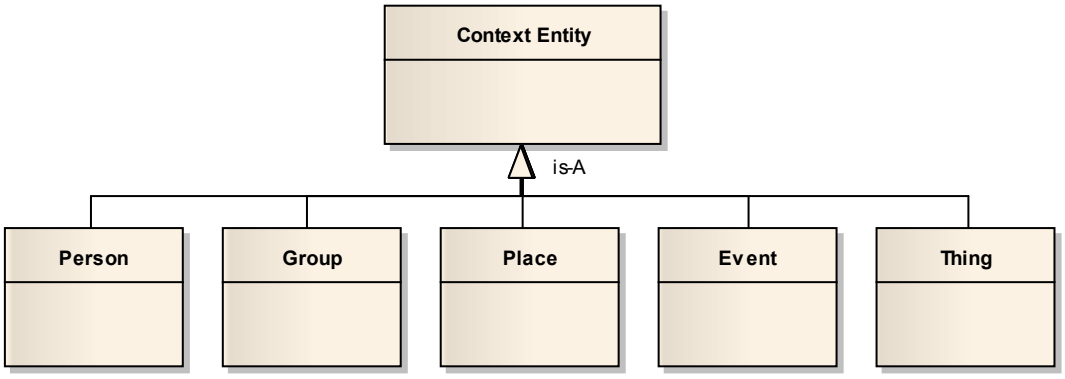


Figure 1: Examples of Context Entities

The Context Information Model details how Context Information is structured and associated to Context Entities in order to describe their situation. In this model, Context Information is organized as Context Elements, which contain set of Context Attributes and associated metadata. Details on this model are provided in the following subsections.

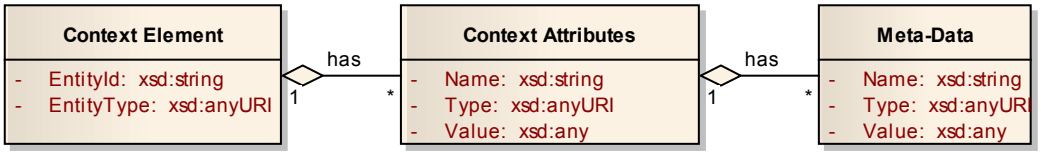


Figure 2: Context Information Model

5.2.1 Context Entity ids and types

Context Entities are identified using an entity identifier (EntityId). The optional entity type may be needed when the EntityId doesn't contain type information or when the EntityId is only unique per entity type.

The entity type is defined as a URI, and thus can be for example an ontology reference (as URL) or a namespace (as URN).

The definition of EntityId and Type values is out of scope of this specification. However, guidelines for their definition are given in Appendix C.

5.2.2 Context attributes and attribute domains

A context attribute represents atomic Context Information. An attribute is defined as a set of information, namely a name, a type, a value and a set of associated metadata (e.g. timestamp, expires, source). The attribute value is expressed as any content, including strings or opaque objects represented using standard formats.

An attribute domain represents the grouping of multiple attributes. Attribute domains allow requestors to specify a set of attributes of interest using a single string as attribute domain name.

The definition of attribute names, attribute types and attribute domains is out of scope of this specification. However, guidelines for their definition are given in Appendix C.

5.3 NGSI-9 Interface Definition

5.3.1 Operation: registerContext

This operation allows registering and updating of registered Context Entities, their attribute names and availability.

The ProvidingEntity URI is used to identify the entity that provides the values of context attributes for registered Context Entities.

5.3.1.1 Input message: registerContextRequest

Part name	Part type	Optional	Description
ContextRegistrationList	ContextRegistration [1..unbounded]	No	List of ContextRegistration structures.
Duration	xsd:duration	Yes	Desired availability period.
RegistrationId	xsd:string	Yes	Registration identifier used to update previous registrations.

5.3.1.2 Output message: registerContextResponse

Part name	Part type	Optional	Description
Duration	xsd:duration	Yes	Confirmed availability period
RegistrationId	xsd:string	No	Registration identifier that could be used to update this registration
ErrorCode	StatusCode	Yes	Error reported by the operation

5.3.2 Operation: discoverContextAvailability

This operation allows the synchronous discovery of the potential set Context Entities, types of Context Entities and related Context Information that can be provided. This is only checked against the registrations issued in the register operation, not against the real source of the Context Information.

In other terms, the discovery operation provides an aggregated list of Context Registration.

5.3.2.1 Input message: discoverContextAvailabilityRequest

Part name	Part type	Optional	Description
EntityId	EntityId [1..unbounded]	No	List of Modifier to identify the Context Entity(ies) to discover.
AttributeList	xsd:string [0..unbounded]	Yes	List of attributes or group of attributes to discover.
Restriction	Restriction	Yes	Restriction on the attributes and meta-data of the Context Information

5.3.2.2 Output message: discoverContextAvailabilityResponse

The response SHALL contain exactly one of the following:

Part name	Part type	Optional	Description
ContextRegistrationResponseList	ContextRegistrationResponse [0..unbounded]	Yes	List of Context Registration responses
ErrorCode	StatusCode	Yes	Error codes for general operation errors

5.3.3 Operation: subscribeContextAvailability

This operation allows the asynchronous discovery of the potential set Context Entities, types of Context Entities and related Context Information that can be provided. This does not guarantee that Context Information about a Context Entity within this set is currently available.

In other terms, this operation allows to subscribe to the notification on the availability of an aggregated list of Context Registrations.

5.3.3.1 Input message: subscribeContextAvailabilityRequest

Part name	Part type	Optional	Description
EntityId	EntityId [1..unbounded]	No	List of identifiers or name patterns of the Context Entity(ies) to discover
AttributeList	xsd:string [0…unbounded]	Yes	List of attributes or group of attributes to be discovered
Reference	xsd:anyURI	No	The interface reference for the notifyContextAvailability operation.
Duration	xsd:duration	Yes	Requested duration of the subscription. Negative values SHALL result in an error. In case of the value 0, the context component SHALL only notify the current value and SHALL NOT send subsequent notifications (one-time subscription). If the Context Management component has a policy to always require duration, the operation SHALL return an error in case the parameter is not present. If the parameter is omitted, the Context Management component MAY select a duration and return this in the response.
Restriction	Restriction	Yes	Restriction on the attributes and meta-data of the Context Information
SubscriptionId	xsd:string	Yes	Used in the notification message and subsequent requests

5.3.3.2 Output message: subscribeContextAvailabilityResponse

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription
Duration	xsd:duration	Yes	Negotiated duration of the subscription
ErrorCode	StatusCode	Yes	Error reported by the operation

5.3.4 Operation: updateContextAvailabilitySubscription

This operation updates a previous subscription to discover Context Information.

5.3.4.1 Input message: updateContextAvailabilitySubscriptionRequest

Part name	Part type	Optional	Description
EntityId	EntityId [1..unbounded]	No	List of identifiers or name patterns of the Context Entity(ies) to discover
AttributeList	xsd:string [0..unbounded]	Yes	List of attributes or group of attributes to be discovered
Duration	xsd:duration	Yes	Requested duration of the subscription. Negative values will result in an error. The receiver of the request MUST reset the new subscription duration starting at the time of reception of the request. If the Context Management component has a policy to always require duration, the operation SHALL return an error in case the parameter is not present. When the parameter is omitted it means that the previous negotiated Duration value is applied, if any.
Restriction	Restriction	Yes	Restriction on the attributes and meta-data of the Context Information
SubscriptionId	xsd:string	Yes	Identifier used in the discoverContextAvailabilityRequest messages

5.3.4.2 Output message: updateContextAvailabilitySubscriptionResponse

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription
Duration	xsd:duration	Yes	Negotiated duration of the subscription
ErrorCode	StatusCode	Yes	Error reported by the operation

5.3.5 Operation: UnsubscribeContextAvailability

This operation deletes a previous subscription to discover Context Information.

5.3.5.1 Input message: unsubscribeContextAvailabilityRequest

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	Yes	Identifier used in the discoverContextAvailabilityRequest messages

5.3.5.2 Output message: unsubscribeContextAvailabilityResponse

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription
StatusCode	StatusCode	No	Status reported by the operation

5.3.6 Operation: notifyContextAvailability

This operation allows receiving the notification about the potential set of Context Registrations subscribed to by the subscriber that implements the notification interface.

5.3.6.1 Input message: notifyContextAvailabilityRequest

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription to which the notification belongs to
ContextRegistrationResponseList	ContextRegistrationResponse [1..unbounded]	No	List of Context Registration responses
ErrorCode			Error codes for general operation errors

5.3.6.2 Output message: notifyContextAvailabilityResponse

Part name	Part type	Optional	Description
ResponseCode	StatusCode	No	Status codes for general operation errors

5.4 NGSI-10 Interface Definition

5.4.1 Operation: queryContext

This operation allows for the synchronous retrieval of Context Information. The requestor of queryContext operation SHALL specify a list of entity identifiers. Such identifiers MAY represent unique entities or entity identifier patterns. An entity identifier patterns SHALL be represented as regular expressions. The requestor of this operation MAY also specify the list of attributes (and/or attribute domains) to be retrieved by this operation. The entity identifiers MAY include type attributes to specify the type of target entities.

It is assumed that the requestor is aware of possible entity types and attributes through Context Entity Discovery operations or by other means. The definition of entity ids, types and attributes is out of scope of this specification.

The requestor of this operation MAY specify restrictions on the returned Context Information. Restrictions are based on the values of attributes and meta-data of the Context Information.

The authentication of the requestor is out of scope of this specification.

Note: in case the NGSI framework will provide common mechanisms for authentication, this specification will rely on such mechanisms to authenticate the requestor.

5.4.1.1 Input message: queryContextRequest

Part name	Part type	Optional	Description
EntityIdList	EntityId [1…unbounded]	No	List of identifiers of the Context Entity(ies) for which the Context Information is requested. Identifiers can contain patterns represented as regular expressions.
AttributeList	xsd:string [0…unbounded]	Yes	List of ContextAttributes and/or AttributeDomains that are queried. Note: If this parameter is absent, the receiver of the request SHALL return all attributes available. NGSI component SHALL return an error if this parameter is required by an service provider policy and is not specified in the message.
Restriction	Restriction	Yes	Restriction on the result set of the query. Restrictions are based on the values of attributes and meta-data of the Context Information

5.4.1.2 Output message: queryContextResponse

The response SHALL contain exactly one of the following:

Part name	Part type	Optional	Description
ContextResponseList	ContextElementResponse [0…unbounded]	Yes	List of Context Information, related attributes (or group of attributes) and metadata.
ErrorCode	StatusCode	Yes	Error codes for general operation errors

5.4.2 Operation: subscribeContext

This operation allows the asynchronous retrieval of Context Information. It is used for subscription to Context Information. The subscription triggers the notifications about the matching ContextEntities based on the defined NotifyCondition information passed in the subscribeContextRequest operation. In the subscribeContextResponse operation a subscription id is returned, which is used for notifications and in update and unsubscribe operations. The subscription duration is negotiated during the subscription request/response operation.

5.4.2.1 Input message: subscribeContextRequest

Part name	Part type	Optional	Description
EntityIdList	EntityId [1…unbounded]	No	List of identifiers of the Context Entity(ies) for which the Context Information is requested. Identifier can contain patterns represented as regular expressions.
AttributeList	xsd:string [0..unbounded]	Yes	List of ContextAttributes and/or AttributeDomains to which the requestor wants to subscribe.
Reference	xsd:anyURI	No	URI that identifies the interface where the notifyContext operation SHALL be invoked.
Duration	xsd:duration	Yes	Requested duration of the subscription. Negative values SHALL result in an error. If the Context Management component has a policy to always require duration, the operation SHALL return an error in case the parameter is not present. If the parameter is omitted, the Context Management component MAY select a duration and return this in the response.
Restriction	Restriction	Yes	Restriction on the attributes and meta-data of the Context Information
NotifyConditions	NotifyCondition [0…unbounded]	Yes	Conditions when to send the notifications.
Throttling	xsd:duration	Yes	Proposed minimum interval between notifications.

5.4.2.2 Output message: subscribeContextResponse

The response SHALL contain exactly one of the following:

Part name	Part type	Optional	Description
SubscribeResponse	SubscribeResponse	Yes	Response to the subscribeContextRequest
SubscribeError	SubscribeError	Yes	The error reported by the receiver of the request

5.4.3 Operation: updateContextSubscription

This operation allows updating a previous subscription to Context Information.

5.4.3.1 Input message: updateContextSubscriptionRequest

Part name	Part type	Optional	Description
Duration	xsd:duration	Yes	Requested duration of the subscription. Negative values will result in an error. The receiver of the request MUST reset the new subscription duration starting at the time of reception of the request. If the Context Management component has a policy to always require duration, the operation SHALL return an error in case the parameter is not present. When the parameter is omitted it means that the previous negotiated Duration value is applied, if any.
Restriction	Restriction	Yes	Restriction on the attributes and meta-data of the Context Information. When the parameter is omitted it means that the previous Restriction is applied.
SubscriptionId	xsd:string	No	Identifier of the reference subscription to be updated
NotifyConditions	NotifyCondition [0…unbounded]	Yes	Conditions when to send notifications When the parameter is omitted it means that the previous NotifyCondition is applied.
Throttling	xsd:duration	Yes	Proposed minimum interval between notifications. When the parameter is omitted it means that the previous Throttling value is applied.

5.4.3.2 Output message: updateContextSubscriptionResponse

The response SHALL contain exactly one of the following:

Part name	Part type	Optional	Description
SubscribeResponse	SubscribeResponse	Yes	Response to the updateContextSubscriptionRequest
SubscribeError	SubscribeError	Yes	The error reported by the receiver of the request

5.4.4 Operation: unsubscribeContext

This operation allows unsubscribing a previous subscription to Context Information.

5.4.4.1 Input message: unsubscribeContextRequest

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	Identifier of the reference subscription to be deleted.

5.4.4.2 Output message: unsubscribeContextResponse

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription
StatusCode	StatusCode	No	The status reported by the receiver of the request

5.4.5 Operation: notifyContext

This operation allows receiving the notification about the Context Information subscribed to by the subscriber that implements the notification interface.

5.4.5.1 Input message: notifyContextRequest

Part name	Part type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription to which the notification belongs to.
Originator	xsd:anyURI	No	The original requestor of the subscription which caused this notification.
ContextResponseList	ContextElementResponse [0…unbounded]	Yes	List of Context Information, related attributes (or group of attributes) and metadata.

5.4.5.2 Output message: notifyContextResponse

Part name	Part type	Optional	Description
ResponseCode	StatusCode	No	The response message reported by the receiver of the request

5.4.6 Operation: updateContext

This operation allows updating a set of Context Information, related attributes and metadata.

Behaviour in Case of empty ContextValue(s) in the request

For each ContextElement of the list of Context Elements received in the updateContextRequest, if an empty Context Value is provided, the operation behaviour SHALL be:

- if the UpdateAction is set to “update” or “append”, the receiver SHALL reject the related changes requested for the specific ContextElement and report an error in the response;
- If the UpdateAction is set to “delete”, the receiver SHALL ignore the ContextValue parameter, perform the related changes requested (delete) and report a success in the response.

Behaviour in Case of non-existing ContextElement(s) or ContextAttribute(s)

For each ContextElement of the ContextElementList received in the updateContextRequest, if the specified ContextElement and/or one of the ContextAttributes does not exist, the operation behaviour SHALL be:

- If the UpdateAction is set to “update” or “delete”, the receiver SHALL reject the requested changes and report an error in the response;

- If the UpdateAction is set to "append", the receiver SHALL create the ContextElement and/or the ContextAttribute(s) respectively and report a success in the response.

5.4.6.1 Input message: updateContextRequest

Part name	Part type	Optional	Description
ContextElementList	ContextElement [1…unbounded]	No	List of Context Elements containing only the subset of Context Information (related attributes (or context domain) and metadata) to be modified.
UpdateAction	UpdateActionType	No	Indicates the type of action that is performed within the update operation: • update: it replaces the value and metadata of the existing attributes with the same name; • append: it adds the new attribute. Note: this may result in multiple attributes with the same name; • delete: it removes the existing value.

5.4.6.2 Output message: updateContextResponse

The response SHALL contain exactly one of the following:

Part name	Part type	Optional	Description
ErrorCode	StatusCode	Yes	Error codes
ContextResponseList	ContextElementResponse [0…unbounded]	Yes	List of response containing the indication of the Context Element and the related statusCode.

5.5 Data Structure definition

This section describes the data structures for the NGSI Context Management part.

5.5.1 ContextElement structure

Element name	Element type	Optional	Description
EntityId	EntityId	No	Identifies the Context Entity for which the Context Information is provided.
AttributeDomainName	xsd:string	Yes	Name of the attribute domain that logically groups together set of Context Information attributes. Examples of attribute domain are: device info (battery level, screen size, …), location info (position, civil address, …).
ContextAttribute	ContextAttribute [0...unbounded]	Yes	List of Context Information attributes. Note: In case of the attributeDomainName is specified all contextAttribute have to belong to the same attributeDomainName.
DomainMetadata	ContextMetadata [0..unbounded]	Yes	Metadata common to all attributes of the logical domain (related to the AttributeDomain)

5.5.2 ContextAttribute structure

Element name	Element type	Optional	Description
Name	xsd:string	No	Name of the Context Information attribute
Type	xsd:anyURI	Yes	Indicates the type of the value field
ContextValue	xsd:any	No	The actual value of the Context Information attribute
Metadata	ContextMetadata [0.. unbounded]	Yes	Metadata about the Context Information attribute (information valid only for the specific attribute)

5.5.3 ContextMetadata structure

Element name	Element type	Optional	Description
Name	xsd:string	No	Name of the metadata.
Type	xsd:anyURI	Yes	Indicates the type of the value field
Value	xsd:any	No	The actual value of the metadata

The following table contains a list of reserved ContextMetadata name string:

Context Metadata name	Description	Data type
Timestamp	Indicates the instant of time when the Context Information has been created	xsd:dateTime
Expires	Indicates the instant of time from which the Context Information should not be considered valid anymore	xsd:dateTime
Source	Identifies the source/provider of the Context Information	xsd:anyURI
ID	Indicates the identifier of the contextAttribute.	xsd:string

Further metadata can be found in Appendix E.

5.5.4 UpdateActionType Enumeration

This enumeration shows the UpdateActionType that can be requested:

Enumeration	Description
Update	Replaces the value and metadata of the existing attributes with the same name
Append	Adds a new attribute. Note: this may result in multiple attributes with the same name
Delete	Removes the existing value

5.5.5 EntityId structure

Element name	Element type	Optional	XML Type	Description
ID	xsd:string	No	element	Identifier of the Context Entity(ies). This value MAY be a string following the anyURI restrictions or a pattern represented as regular expressions following Appendix F of [XML-Schema-Part2].
Type	xsd:anyURI	Yes	attribute	Indicates the type of Context Entity(ies) for which the Context Information is requested. If EntityId uniqueness is only guaranteed in combination with Type, then Type SHALL be present.
IsPattern	xsd:Boolean	Yes	attribute	Indicates whether the EntityId is a pattern (expressed as a regular expression following Appendix of [XML-Schema-Part2]) or an id. If this attribute is omitted, it SHALL be treated as false.

5.5.6 Restriction structure

The Restriction data structure contains two different kinds of restrictions:

- the AttributeExpression filters the result set based on expressions on the values of the context attributes
- the Scope restricts the operational search space on which a given operation needs to operate

Compared to attributeExpression parameter, scopes a-priori limit the set of context sources that are needed for serving the request. The criteria used in operational scopes do not need to be part of the Context Information itself. Appendix D gives more explanation for scopes.

Element name	Element type	Optional	Description
AttributeExpression	xsd:string	No	String containing an XPath restriction. Note: The XPath expression will be evaluated against ContextEntity structures.
Scope	OperationScope [0..unbounded]	Yes	List of scope definition

5.5.7 ContextRegistration structure

This structure is used in the ContextRegistrationList parameter of registerContext operation and the ContextRegistration field of the ContextRegistrationResponse structure.

This structure can be used either to register/update the information about ProvidingApplication or to register/update the availability of ContextEntities and their related attributes.

If the ContextRegistration structure is used to update the registered information about the ContextEntities, EntityIdList SHALL be present together with ContextRegistrationAttribute parameter.

If the ContextRegistration structure is used to register the metadata about ProvidingApplication, RegistrationMetadata SHALL be present.

The registration/update of ContextEntities and ProvidingApplication can be done at the same time.

Element name	Element type	Optional	Description
EntityIdList	EntityId [1..unbound]	Yes	List of identifiers for the Context Entities being registered
ContextRegistrationAttribute	ContextRegistrationAttribute[0...unbounded]	Yes	List of ContextAttributes and/or AttributeDomains which are made available through this registration.
RegistrationMetadata	ContextMetadata [0...unbounded]	Yes	Metadata characterizing this registration
ProvidingApplication	xsd:anyURI	No	URI identifying the application that provides the values of the context attributes for the target Context Entities.

5.5.8 OperationScope structure

The OperationScope data structure defines:

- the ScopeType which selects the type of Scope that is used, and
- the ScopeValue which defines parameter of the scope.

Element name	Element type	Optional	Description
ScopeType	xsd:string	No	Name of the scope type.
ScopeValue	xsd:any	No	Contains the scope value for the defined scope type.

The following table contains a list of reserved ScopeType name strings:

OperationScope scopeType	Description
SimpleScope	Keyword identifying a simple scope in an OperationScope structure.

Note: Further scopes and their description can be found in Appendix D (Informative).

5.5.9 SimpleScope structure

This structure SHALL be used in the scopeValue field of OperationScope structure if scopeType contains the keyword “SimpleScope”.

A SimpleScope is a XPath expression that is evaluated against the ContextRegistration data structure used to register Context Entities.

Element name	Element type	Optional	Description
SimpleScopeExpression	xsd:string	No	String containing an XPath restriction. Note: The XPath expression will be evaluated against the ContextRegistration structures.

5.5.10 NotifyCondition structure

Element name	Element type	Optional	Description
Type	NotifyCondition enum	No	This element specifies the notifyCondition, and SHALL assume one of the following values: • ONTIMEINTERVAL • ONCHANGE • ONVALUE
CondValue	xsd:string [0..unbounded]	Yes	When present, this element qualifies the NotifyCondition based on the type as follows: • Type ONTIMEINTERVAL: exactly one condValue SHALL be present and SHALL represent the time interval between notifications. • Type ONCHANGE: this element SHALL be present and contain the name(s) of the Context Attributes to be monitored for changes. • Type ONVALUE: this element SHALL not be present for this type.
Restriction	xsd:string	Yes	This element SHALL be present only if the NotifyCondition type is set to ONVALUE. When present, this element indicates the restriction that applies before a notification is sent. The parameter is specified as XPath expression.

5.5.11 NotifyCondition enumeration

The NotifyCondition type SHALL be set to one of the following enumerated values:

Enumeration	Description
ONTIMEINTERVAL	The condition is true when the time interval specified in the value field is reached.
ONVALUE	The condition is true if the value of one or more context attributes fits a reference value and/or range, specified in the Restriction parameter
ONCHANGE	The condition is true when a change in one of the specified context attributes has occurred

5.5.12 SubscribeResponse structure

Element name	Element type	Optional	Description
SubscriptionId	xsd:string	No	The identifier of the subscription
Duration	xsd:duration	Yes	Negotiated duration of the subscription. The Context Management component MAY omit this parameter if it allows indefinite subscriptions
Throttling	xsd:duration	Yes	Negotiated minimum interval between notifications. If a Throttling value were proposed into the subscribeContextRequest, this parameter SHALL be specified.

5.5.13 SubscribeError structure

Element name	Element type	Optional	Description
SubscriptionId	xsd:string	Yes	The identifier of the subscription. This parameter is mandatory in case of updateContextSubscriptionResponse
ErrorCode	StatusCode	No	The error reported by the receiver of the request

5.5.14 StatusCode structure

Element name	Element type	Optional	Description
Code	xsd:int	No	Numerical value that identifies the status code.
ReasonPhrase	xsd:string	No	Human readable text that describes the status code
Details	xsd:any	Yes	Contains more details on the StatusCode

The following table contains a list of reserved StatusCode for the Code and ReasonPhrase elements:

Code	ReasonPhrase	Description
200	Ok	This StatusCode indicates a success in the operation performer or requested.
400	Bad request	This StatusCode indicates that the request is not well formed.
403	Forbidden	This StatusCode indicates that the request is not allowed.
404	ContextElement not found	This StatusCode indicates that the ContextElement requested is not found.
470	Subscription ID not found	This StatusCode indicates that the subscription ID specified does not correspond to an active subscription
471	Missing parameter	This StatusCode indicates that a parameter is missing in the request.
472	Invalid parameter	This StatusCode indicates that a parameter is not valid /allowed in the request.
473	Error in metadata	This StatusCode indicates that there is a generic error in the metadata (e.g. Expires older than timestamp).
480	Regular Expression for EntityId not allowed	This StatusCode indicates that a regular expression for EntityId is not allowed by the receiver.
481	Entity Type required	This StatusCode indicates that the EntityType is required by the receiver
482	AttributeList required	This StatusCode indicates that the AttributeList is required
500	Receiver internal error	This StatusCode indicates that an unknown error at the receiver has occurred.

5.5.15 ContextElementResponse structure

Element name	Element type	Optional	Description
ContextElement	ContextElement	No	Context Information related to a Context Entity Note: In case of error, this data structure can contain only the EntityId or the EntityId/Attribute combination that cause the error. In case of success, this data structure contains also ContextAttribute and needed related ContextMetadata (e.g. ID).
StatusCode	StatusCode	No	Identifies the status of the requested operation related to this specific ContextElement.

5.5.16 ContextRegistrationResponse structure

Element name	Element type	Optional	Description
ContextRegistration	ContextRegistration	No	The Context Registration that was requested. Note: In case of error, this data structure can contain only the EntityId or the EntityId/Attribute combination that cause the error.
ErrorCode	StatusCode	Yes	Identifies the status of the requested operation related to this specific ContextRegistration. This element SHALL be omitted in case there is no error.

5.5.17 ContextRegistrationAttribute structure

Element name	Element type	Optional	Description
Name	xsd:string	No	Name of the ContextAttribute and or AttributeDomain.
Type	xsd:string	Yes	Indicates the type of the ContextAttribute value
IsDomain	xsd:boolean	No	Indicates if this structure refers to a ContexAttribute or a AttributeDomain
Metadata	ContextMetadata [0... unbounded]	Yes	Metadata about the Context Information attribute (information valid only for the specific attribute)

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions : OMA-TS- NGSI Context_Management-V1_0	13 Oct 2009	none	Creation of baseline document.
	22 Oct 2009	1, 2, 3, 4, 5	Incorporate 46, 47R01, 48R01, 58.
	28 Jan 2010	5 and related subsections	Incorporate 2010-0007, 2010-0015. Fixe some editor typos.
	04 Feb 2010	5.1	Incorporate 2010-0041R02. Update the TS template to the 2010 version (footer and 2010 copyright in page 2)
	12 Mar 2010	5.2 and 5.3	Incorporate 8R04, 37R02 and 50R01.
	08 Apr 2010	5.2 and 5.3	Incorporate 5R04.
	07 May 2010	5.2 and 5.3	Incorporate 74, 75R03, 79 and 82R02.
	27 May 2010	5.2, 5.3, 5.4	Incorporate 6R02, 59R02, 101, 102 and 103. Template cleaned-up.
	30 May 2010	5.3 and 5.4	Incorporate 76R02
	09 Jun 2010	All	Incorporate 116, 117R01, 118, 121, 122R01, 123R01, 124, 126, 127, 128, 135. Fixed some editorial changes.
	10 Jun 2010	Appendix E, 5.4.1.2, 5.3.2.1, ALL	Incorporate 129R01, 130, 0132, Fixed some editorial changes.
	30 Jun 2010	All	Includes changes related to closed comment during CONR phase, according to OMA-CONRR-NGSI-V1_0-20100630-D and CRs 146R01, 147R02, 148R01, 149R01 and 151.
Candidate Version : OMA-TS- NGSI Context_Management-V1_0	03 Aug 2010	All	Editorial fixes : template styles and History table Status changed to Candidate by TP: OMA-TP-2010-0324-INP_NGSI_V1_0_ERP_for_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

As NGSI v1.0 specifies the level of abstract interfaces, no testing of those is applicable. Therefore, the Static Conformance Requirements (SCR) tables are not defined. Those are subject for definitions in the related technical specification defining the bindings.

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for XYZ Client

None

B.2 SCR for XYZ Server

None

Appendix C. Guidelines for defining Context Entities and Context Information (Informative)

C.1 Guidelines for defining Context Entity types and ids

An entity type can be a OMNA registered namespace (URN), an existing standard namespace or a proprietary namespace. Similarly, any proprietary or standard ontology reference (URL) can be used as entity type.

Similarly, entity identifiers (Id) are URIs, which may contain OMNA registered namespace, an existing standard namespace, a proprietary namespace or any URL.

A OMA enabler or any other system willing to provide access to information about Context Entities through the interfaces defined in this specification may define new types and identifiers of Context Entities, or reuse existing ones (standard or proprietary).

For example, the interfaces defined in this specification could wrap SIP-related enablers (e.g. to access Presence information through SIMPLE enabler), in this case Entity URI could take the form of SIP URI.

In a similar way, for interaction with DPE enabler one could use as Entity URI a URN formed by the "urn:x-oma-application:dpe:" prefix followed by the DPE Server-ID/Client-Id.

Yet in another way, connected objects of different types could generally be represented through an Entity URI that corresponds to their MAC address in the form of a UUID URN. In such cases, the 'type' information may be used to identify the actual type of entity (object) it represents.

C.2 Guidelines for defining context attributes and domains

The definition of an attribute consists in the definition of the attribute name, the attribute type and the format of the attribute value. It may also define some specific associated metadata.

The definition of an attribute type or attribute name can occur by registering a namespace (URN) with OMNA, by reusing an existing standard namespace or by using a proprietary namespace or name. Similarly, any proprietary or standard ontology reference (URL) can be used to define a new attribute type or attribute name.

The definition of an attribute domain can occur by registering a namespace (URN) with OMNA, by reusing an existing standard namespace or by using a proprietary namespace or name. It may also define some specific associated metadata.

A OMA enabler or any other system willing to provide access to information about Context Entities through the interfaces defined in this specification may define new context attributes or attribute domains, or reuse existing ones (standard or proprietary). For example, the interfaces defined in this specification could interact with the SIP-related enablers (e.g. to access Presence information through SIMPLE enabler) by using registered attribute "SIMPLE-Presence", the "urn:org.openmobilealliance:presence" as type and the actual presence information (represented as PIDF fragment) as value.

In a similar way, interaction with DPE enabler could rely on the "DPE-deviceprops" attribute, which uses "urn:org.openmobilealliance:device:properties" as attribute type and the actual set of device properties representation as value. In that case, the single attribute and value would opaquely represent a collection of device properties and their individual values. As an alternative model, each DPE device property could be mapped to a different attribute.

Appendix D. Scopes (Informative)

Context management components will manage different Context Entities from various sources with various characteristics (e.g. sensor information with high update rates). It can be expected that a massive number of Context Entities from a large geographic area needs to be managed thus resulting in potential efficiency problems. Though OMA NGSI cannot make any assumptions about the implementation of the Context Management component, the interface must provide a mechanism through which entities using the NGSI-10 interface can inform the Context Management component about additional hints or constrains reducing the search space. Such hints can help centralized systems to optimize their internal access path and distributed systems to organize their communication patterns internally and with other context providing systems. The Restrictions data structure contains therefore the fields attributeExpression and scope.

A (operation) scope restricts the operational space on which a given operation needs to operate. Compared to attributeExpression (filters applied after accessing the Context Information), operation scopes a-priori limit the set of context sources that are needed for serving the request without accessing the Context Information itself. The criteria used in operational scopes do not need to be part of the Context Information itself. Typical operation scopes are according to real world location (geographic or civic location), the network structure, or organizational structures. The NGSI-10 interfaces supports an extensible set of operational scopes.

Example (informative): a centralized organized context management system can keep a geographical index on all its managed objects. Assuming an operation scope limits the operation to a specific city C, the index would enable fast access to e.g., all Context Entities in that given city C. In a distributed context management system, the same operation scope would result in forwarding the query to only those servers that hold part of the Context Information related to city C.

Section 5.5.8 defines the normative list of scopes for NGSI Context Management TS. This Appendix defines further scopes that can be used by a Context Management component. The following list of scopes is defined:

Keyword	Description
SimpleGeoLocation	Geographic location scope shall be used when using a bounding segment spanned by two geographic coordinates.
GML_Location	This operational scope shall be used when 2D surfaces or 3D spaces are used. This scope is defined using 2D surfaces or 3D spaces as defined in the subset of GMLv3 defined in [GeoShape] [1].
CivicLocation	Symbolic location scope using civic location as specified in [RFC4119] and [RFC5139]
NetworkDomain	Network domain scope provides relative or absolute scope according to network structure. Only context sources that are attached to the network within this scope need to be considered.

The keyword in the table above can be used as the value of the attribute scopeType in the data structure OperationScope if the respective scope is used.

D.1 SimpleGeoLocation

In the two dimensional case, SimpleGeoLocation scopes are defined by providing the north-west and south-east geographic coordinates of segments on the surface of the earth. The borders of the segment are then defined by the longitudes and

[1] In the OMA LOCSIP specification document [LOCSIP], the Presence Information Data Format Location Object (PIDF-LO) is referred to. In turn, the location information in a PIDF-LO may be described in a geospatial manner based on a subset of GMLv3 defined in [GeoShape], or as civic location information defined in [RFC4119] and [RFC5139].

latitudes of the geographic coordinates. In the three dimensional (or rather 2.5 dimensional) case, the coordinates have to contain height information, defining the base segment, plus a height on top of that base. SimpleGeoLocation scopes allow simple checks for overlap and inclusion and are easy to implement.

D.1.1 SimpleGeoLocation scope value structure

The following data structure shall be used for scopeValue if the scopeType contains the keyword "SimpleGeoLocation".

Element name	Element type	Optional	Description
Segment	Segment	no	Segment defining the simple geo location

D.1.2 Segment structure

Element name	Element type	Optional	Description
NW_Corner	gml:point	no	2D or 3D coordinate of north-west corner of the segment on the surface of the earth
SE_Corner	gml:point	no	2D or 3D coordinate of south-east corner of the segment on the surface of the earth
Height	xsd:double	yes	Provides height information for 3D segment

D.2 GML_Location

Scopes of type GML_Location allow the specification of Operation Scopes using two or three dimensional GML primitives as supported in the subset of GMLv3 identified in [GeoShape].

D.2.1 GML_Location scope value structure

The following data structure shall be used for scopeValue if the scopeType contains the keyword "GML_Location".

Element name	Element type	Optional	Description
Surface	gml:AbstractSurfaceType	Yes	2D surface describing location scope
Space	gml:AbstractSolidType	Yes	3D space describing location scope

Note: either a surface or a space needs to be specified.

D.3 CivicLocation scope value structure

Scopes of type CIVIC_LOCATION allow the specification of Operation Scopes as specified in [RFC4119] and [RFC5139].

D.3.1 CivicLocation scope value structure

The following data structure shall be used for scopeValue if the scopeType contains the keyword "CivicLocation".

Element name	Element type	Optional	Description
CivicLocation	ca:civicAdress	No	Civic Location as specified in [RFC 5139]

D.4 NetworkDomain scope value structure

The network domain consists of a list of supported network domains. These can be absolute domains, i.e., explicitly specified giving a name, or they can be relative with respect to a Context Management component. Relative domain scopes are enumerated in D.4.1.2.

D.4.1 NetworkDomain scope value structure

The following data structure shall be used for scopeValue if the scopeType contains the keyword “NetworkDomain”.

Element name	Element type	Optional	Description
NetworkDomain	Set of xsd:string	No	Absolute of relative network domain scopes.

D.4.2 RelativeScope keyword

Keywords	Description
Core	Includes only sources of Context Information in the core network of an operator.
AccessNetworks	Includes all sources of Context Information in the access networks of an operator
Terminals	Includes all sources of Context Information on terminals
External	All external sources accessible are taken into account.

Appendix E. List of further Meta-Data (Informative)

In section 5.5.3 is defined the ContextMetadata structure and a normative list of metadata for NGSI Context Management that shall be supported by all context components. This structure is used to communicate metadata that describes context elements or context registrations.The following table contains a list of additional metadata (mostly related to Quality-of-Context parameters) that can be used:

Context Metadata name	Description	Data type
DelayTime	Typical time lag between the time the event occurred in the real-world and is made available to the system	xsd:duration
ObservationArea	A physical area for which the provided value is valid.	gml:AbstractSurfaceType
SourceLocation	Location of the component (e.g. sensor) that provided the information	gml:point
Accuracy	typical difference between real and reported value	Same as value
Precision	Describes a classification error as the ratio between true positives and the sum of true positives plus false positives	xsd:float
Recall	Describes a classification error as the ratio between true positives and the sum of true positives plus false negatives	xsd:float
Owner	Legal owner of the information	xsd:string
Provider	Legal entity that provides the information, e.g. network operator	xsd: string
Observer	Legal entity that observes the information, e.g. a user	xsd:String
Confidence	Value between 0 and 1 indicating the reliability of the value	xsd:float
Resolution	The Resolution is the granularity or minimal perceivable change of the value measurement	Same as value