The Context API in the OMA Next Generation Service Interface

Martin Bauer, Ernő Kovacs, Anett Schülke  
NEC Laboratories Europe  
Kurfürsten-Anlage 36, 69115 Heidelberg, Germany  
{martin.bauer, erno.kovacs, anett.schuelske}@neclab.eu,  
Naoko Ito, NEC Corporation, naoko@ej.jp.nec.com  

Abstract—Application Platforms and related APIs are becoming the foundations for emerging eco-systems of online applications/services. End-users are more and more choosing devices and services based on the available offers in the respective eco-system. Attracting and retaining application developers and end-users to a networked platform is nowadays more and more depending on the features available to developers in the form of APIs and (e.g. cloud-based) execution environments rather than on the actual portfolio of services. The Open Mobile Alliance (OMA) Next Generation Service Interfaces (NGSI) is one cornerstone of evolving a network operator towards an application platform provider with open APIs. Within NGSI, the Context API enables access to a plethora of rich information about users, places, events, and things - all made available by or brokered through the network operator. This paper explains the OMA NGSI Context API and its rationale, describes the operations, and discusses typical use cases and benefits it can bring to operators and application developers.

Keywords: OMA NGSI, Context, Context-awareness, API, application platforms, web services, SDP

I. INTRODUCTION

The tremendous success of Web 2.0 services have proven once again that the Web has become a platform on which viable and large scale business can be conducted. But technology evolution has not stopped there. Companies are evolving their (existing) popular services to become application platforms. A successful eco-system connects hundreds of thousands of developers with millions of users. It leverages on their platform. Voice-related services (especially for Enterprises), SMS-based Value Added Services, as well as the early success of iMode (and similar services) are first examples of services enabling new applications and offering a diverse range of services. But these early successes are dwarfed now by the speed and increased richness of the services and applications built on the Web. Clearly, these newcomers have put priority on transforming their platform and their APIs into easy-to-use tools which are commonly available for building an application eco-system.

The question is how network operators can use the same approach to advance their widely used base services into an application platform. Within the Open Mobile Alliance (OMA, [1]), activities have started to define the Next Generation Service Interfaces (NGSI), based on the lesson learned from the Web [2]. OMA NGSI is an attempt to provide the necessary APIs to developers of rich personal services by turning operator networks and sensor-rich devices into an application platform.

In this paper, we especially examine the role of the Context API inside NGSI to provide access to the rich context information available in operator’s networks.

II. OMA’S NEXT GENERATION SERVICE INTERFACE

The Next Generation Service Interfaces (NGSI) that are being standardized at OMA aim at stimulating the usage of various service enablers into new services and applications usable by traditional Web developers, starting from extensions of today’s Parlay X APIs (latest version: 3GPP Rel 8 Parlay/Parlay X APIs, [3]) and addressing various new API topics. Considering how widely networks influence our daily lives nowadays, these new API topics target a broad range of information sources, social communities, e-commerce, as well as network usage types. The goal of NGSI is to create a wide set of open APIs enabling Next Generation Services to be created and used in areas of individual (“Life Creation”), corporation (“Business Process Innovation”), and society (“Safe and Secure Public / Medical Services”).

Network operators have a long history of working with service providers and 3rd parties to realize new services based on their platform. Voice-related services (especially for Enterprises), SMS-based Value Added Services, as well as the early success of iMode (and similar services) are first examples of services enabling new applications and offering a diverse range of services. But these early successes are dwarfed now by the speed and increased richness of the services and applications built on the Web. Clearly, these newcomers have put priority on transforming their platform and their APIs into easy-to-use tools which are commonly available for building an application eco-system.
The NGSI APIs cover the following topics:

- Extensions for **personal communication services**, including Rich Multimedia services support. This includes advanced multimedia conferences, e.g. conference member management and advanced call handling schemes, extensions for third party call control and call notifications, media enhancements, behavior configuration (Find me/forking), control on display/restrict calling line identification

- Enhancements to the **conference experience** (including manipulation of group lists, multimedia streaming control)

- Advanced **content sharing and delivery**, including (a) functionalities to add meta-data information to the shared content in order to associate additional 3rd party content related with user profiles, (e.g. service recommendations or targeted advertisement), (b) synchronized content delivery (e.g. combination of video contents and advertisement clips), and (c) access to media/data storages and related control mechanisms, including federation and management of identities within/across domains,

- Sophisticated **media control** (e.g. media control scripts and guidance insertion, or stream control like play/list functions) and relevant charging.

- **Context and Personalization** within the scope of the above-proposed extensions to multimedia services. (a) improvement for the policy management (e.g. integrating context to allow use cases like presence-based forwarding), (b) user-driven manipulation of multi-media experience - to manipulate the experience in a user-centric manner driven by user preferences or/and context (e.g. personalized ad-insertion by application context)

![Figure 1: High-level view of NGSI enabler](image)

Figure 1 illustrates the main NGSI concepts. NGSI define the APIs necessary for realizing these functionalities and are expected to provide either REST or SOAP bindings to ease application integration (1). Such APIs are expected to serve as front-end for existing OMA enablers such as Location, Presence, User Profile Management, Service User Profile Management (SUPM), Converged IP Messaging (CPM), Device Profile Evolution (DPE), and others (2). Access to these new capabilities provided through the NGSI APIs will give application developers new means for creating personalized applications, which can invoke network services, utilize user identity and profiles, and can react dynamically on changes in the end-users context (3).

In summary, the NGSI provides flexible APIs for easy-to-use access to the operator’s core platform. Exposing these APIs will be a cornerstone for operator’s networks to evolve into an application platform.

III. **CONTEXT MANAGEMENT**

Focusing on one of the main new topic addressed by NGSI APIs, Context Management is the component that enables network operators to provide context information about users, places, events, and things (see Internet-of-Things, [5]) in a common and abstracted way.

In the MUSIC EU project [4] for example, four trial applications have been developed that use context to adapt user interfaces and mobile application behavior. In the MUSIC “ RATP Travel Assistant”, a mobile application prototype for Paris Metro travelers, train delays are modeled as context data and the application subscribes to delays for the lines that are part of a user’s itinerary. When a delay occurs, it is published (“update” operation) in the network as context information and notified to all subscribed Travel Assistant applications that, based on location context, can propose a new itinerary to the user. Application behavior is also made dependent to device’s context information such as: battery level, available connectivity and screen capabilities, which are all modeled and described as attributes using the same abstract context model.

By providing a standard API for accessing such information, context is made available to 3rd party application developers that can build new services without having to define specific mechanisms for context distribution and management. Operator’s especially have all kind of context information available, e.g. from their network measurements, from the service usage, from the devices, as well as from user profile and interest. In the future, new network elements like Femto Cells or new services build on using the ETSI M2M protocol stack will give even richer opportunities to operators. This is a unique opportunity to explore advanced functions of the networks and provide services that only an operator can provide. The OMA NGSI Context API is an important standard for this.

The specificity of context information is that it is not only a measured value (like sensors in a sensor networks) and its related measurement unit, but that it is associated with the entity it describes and provides semantic meanings for the values and the relationship contained in the describing attributes. In other words, besides delivering the information from a sensor that measures 29°, it also indicates that this information describes the room temperature of Room 3.24. Relying on this information, context processing includes all
kind of workflows that aggregate, analyze, infer, and predict new context information.

In order to be a generic API for all kinds of context information, OMA NGSI Context API strictly focuses on a limited set of generic operations and makes only minimum assumptions about the context information models used to describe context information and its relationship with context entities.

A. Context Information Model

Context Management systems typically handle context information about a wide range of heterogeneous context entities, such as persons, places, events, things, etc. Ideally, the set of entities can be dynamically extended.

In the model adopted within the OMA NGSI Context API (see Figure 2), a minimum set of assumptions are made to allow efficient operations on context. The main assumption in the proposed model is the representation of relationships between entity and information: context entities are identified and classified by an ID and, optionally, a type. Each context entity is associated with a set of attributes, as well as a set of metadata characterizing these attributes. Meta-data for context attributes contain timestamps, expiration time, source of information, observation areas, location of the measuring device, and further Quality-of-Context information like accuracy, precision, recall, confidence, and resolution.

The context information model does not make assumptions about how entity IDs are created and managed or whether they are unique. It only assumes that entity IDs can be matched exactly or through regular expression patterns to identify large sets of context entities. Context attributes can contain all kind of data structures as well as multi-value sets. In addition, meta-data is used to further describe the information contained in the attributes, e.g. to express the time when it was sensed, the quality of the sensed information, the origin of the information, and many other meta information. The context model used in OMA NGSI inherits from the European projects SENSEI [6], MUSIC [4] and CCAST [7].

B. OMA NGSI Context Interfaces and Operations

The OMA NGSI Context API provides two different interfaces (see Figure 3):

1. **Context Source Management**: it enables external providers to be integrated into the operator’s network. The provided operations allow to register context sources and their information with the network operator as well as to provide a discovery function with which participating components can check for available context information. With these two functions, a federated system between e.g. the context management system of a network operator and a set of context management systems operated by interested suppliers can be established and maintained.

2. **Context Information Management**: it enables provisioning and accessing to context information. The provided operations allow updating (from a context source), querying (synchronously and usually short-term), as well as subscribing to and being notified about context information (asynchronous and long lasting access).

The following sections explain the interface in more details.

IV. OMA NGSI CONTEXT OPERATIONS DETAILS

The following chapter discusses the defined set of context operations. Figure 3 gives an architectural overview of the Context Management within NGSI.

A. **Register**

This operation is used to announce sets of available context entities. To avoid announcing each single context entity made available by a 3rd party, the operation allows batch registration using (a) naming patterns, and (b) entity types. Additionally, the registration provides the set of available attributes, metadata, as well as further information about the registered entities. It also provides the reference (a URL) where the Context Information can be accessed.
B. Discovery

“Discovery” is the dual operation of the “Register”. It allows checking which context information is currently made available by a providing “context source component”. The important difference with directly querying for entities is that it only delivers the structure information, but does not try to compute the actual values of these context attributes. Computing values of attributes could potentially trigger heavy computation and communication load inside the network or additional communication with the user’s device. For this operation both query and subscribe/notify modes are available.

C. Update

The “Update” operation is used to provision one or more attributes into the Context Management component. Usually, the different attributes of context entities are not stored or computed together, but are delivered through different mechanisms from underlying enablers, e.g. from the OMA Location Enabler or from OMA Device Profile Evolution Enabler (DPE). Therefore, it is highly important to be able to update single attributes or collections of attributes instead of the complete context information. Furthermore, updates of the same attribute can be delivered from different providers. Location is a typical example in which the user’s location can be measured with different quality by location sensors like GPS, CellId, WLAN-in-range, as well as measured interactions of the user with physical objects in his environment. These can be then distinguished based on meta-data (which mention the provider information as well as the Quality-of-Context information).

D. Query

“Query” is the synchronous request for context information. Various types of queries are supported: (a) entity queries for context entities with well-known IDs, (b) ID pattern queries for a set of entities, or (c) type-based queries which are searching for entities with a common type. As ID pattern and type-based queries can easily create large result sets, the query operation contains restrictions which are basically Boolean operations on the values of the attributes of an entry. As the post-processing evaluation of a restriction can potentially create large load on the system (e.g. in the form of signaling traffic, compute load, or communication with the end terminals), the concept of “Scopes” has been introduced. Scopes are evaluated against meta-data known about the context providers. This pre-processing can potentially reduce the result set without the need to fetch or compute unnecessary context attributes.

As an example, context providers might specify a geographic region, which details the limited observation area it covers. A scope in a query can then specify an area-of-interest: If the system detects a match between the observation area and the area-of-interest, the system will send the query to the respective context provider. The standard defines a basic scope that shall be supported by all implementations, and an informative set of scopes that an implementation of OMA NGSI can support. The basic scope is basically an XPath expression that is evaluated against the meta-data describing a “registered” context source component. The extended scope suggestions contain

- SimpleGeoLocation scope: restricting a context queries to a rectangle-like sector on the surface of Earth
- GML_Location: restricting context queries to areas which are described using GML. This could be either 2D surface or a 3D space
- CivicLocation
- NetworkDomain scope that limits the elements of the network which are asked for context information, e.g. core network servers, access network elements, or end-user terminals

E. Subscribe

Besides synchronous query requests, subscribe-notify access to context information is also important for applications [10]. In OMA NGSI Context API, “Subscribe” is the operation enabling long-lasting monitoring of the system. “Subscribe” basically uses the same parameters as “Query” but includes additional parameters, which further detail the handling of notification operation. When a subscription is issued, the matching entries are returned in the form of an asynchronous “notify” callback operation (one or more notifications can be sent). The target of the “notify” operation can be either the requestor of the subscription or a 3rd party. Hence, users of this API can build “external” controlled processing trees for processing of context information. This separation of control and data flow eases the creation and management of context processing workflows.

Further, the mentioned additional parameters
- allow the duration of the subscription to be negotiated,
- define the condition when to send another notification, and
- distinguish between one-time subscriptions (basically an asynchronous query) and long-lasting subscriptions, which are repeatedly delivering information.

V. Business Aspects

As illustrated above, OMA NGSI Context API enables access to context information about entities of the real world: users, places, events, things, and any other entities that providers can think of. The Internet has made access to information very easy and created a free-of-charge mentality. Nevertheless, there is still value in high-quality information that cannot be easily obtained. The chance for operators is to create value out of low-level observations in their networks and from partners.

As an example, by monitoring the position of mobile phones near the entrances of tourist attractions, an operator can roughly determine the waiting time for this attraction. While “privacy laws” basically will prohibit selling the low-level position information on its own, the aggregated high-level waiting time parameter can (a) not be traced back to individuals, and (b) become useful information for attractive services provided by the operator itself or by 3rd parties. Similar experiments have been conducted with monitoring the speed of cars in certain areas to determine the likelihood of a
traffic jam. Especially the density of mobile phone might give indications, but also an increased rate of phone calls as well as Internet activities. Such type of sensing has been validated on a large scale through snapshots of touristic, traffic and social flows within the Real Time Rome experiment [11], leveraging on data collected from the communication networks and exploiting interconnection between people, device, place, object.

Another potential business model that relies on context information is to provide free access to a limited and basic set of context information and price queries resulting in large result sets. The discussed “scope” concept can be particularly useful to limit and distinguish basic free searches from premium searches. For example, queries/subscriptions might be free-of-charge for the close area around a user, but might be charged for when searching a larger or remote area.

A further business model could provide premium context information to special partners of the operator, e.g. to its MVNO (Mobile Virtual Network Operator) partners. The OMA NGSI Context API enable MVNOs to provide more customized services based on network and user information. In the long run, a market might appear in which operators offer to create and install custom-made context processing in their telecommunication cloud and sell this as a specific service to customers.

VI. OUTLOOK

At the time of writing, the standardization of OMA NGSI Context API has finalized a candidate version 1.0. This version is now in the process of being approved. OMA NGSI first defined an abstract API. In a subsequent step, the abstract API need to have a concrete binding, e.g. to SOAP or to REST.

Utilizing context information can create additional value for network operators that can be used for attractive services. As the explosion of location-based services has shown, 3rd parties are the most creative and agile elements in the emerging new value chains that are creating new and innovative applications. The OMA NGSI Context API is a way of exposing selected and valuable information from the network to application developers in a standard manner and through that it represents an important building block to weave applications into the fabric of the network economy.

Another interesting development is the specification of M2M interfaces (e.g. in ETSI M2M) which enable to build context processing components that use a wider range of sensor and measurement equipment.

ACKNOWLEDGMENT

The authors of this paper acknowledge the partial financial support given to this research by the European Union in the 6th Framework Programme through the MUSIC-IST project ([14], contract number 35166) and in the 7th Framework Programme, through the IST-SENSEI project ([12], contract number ICT-215923), and the “C-CAST” - Content Casting project ([7], contract number ICT-216462). We also like to thank our colleagues in these projects. Without their support, this work would not have been possible.

REFERENCES