Documento de diseño del lenguaje Athena

Abstract Schema Definition Language

Alberto Hernández Chillón Diego Sevilla Ruiz Jesús García Molina

{alberto.hernandez1,dsevilla,jmolina}@um.es

Facultad de Informática
Universidad de Murcia

Murcia, España, Septiembre 2019

Índice de contenido

Índice de contenido

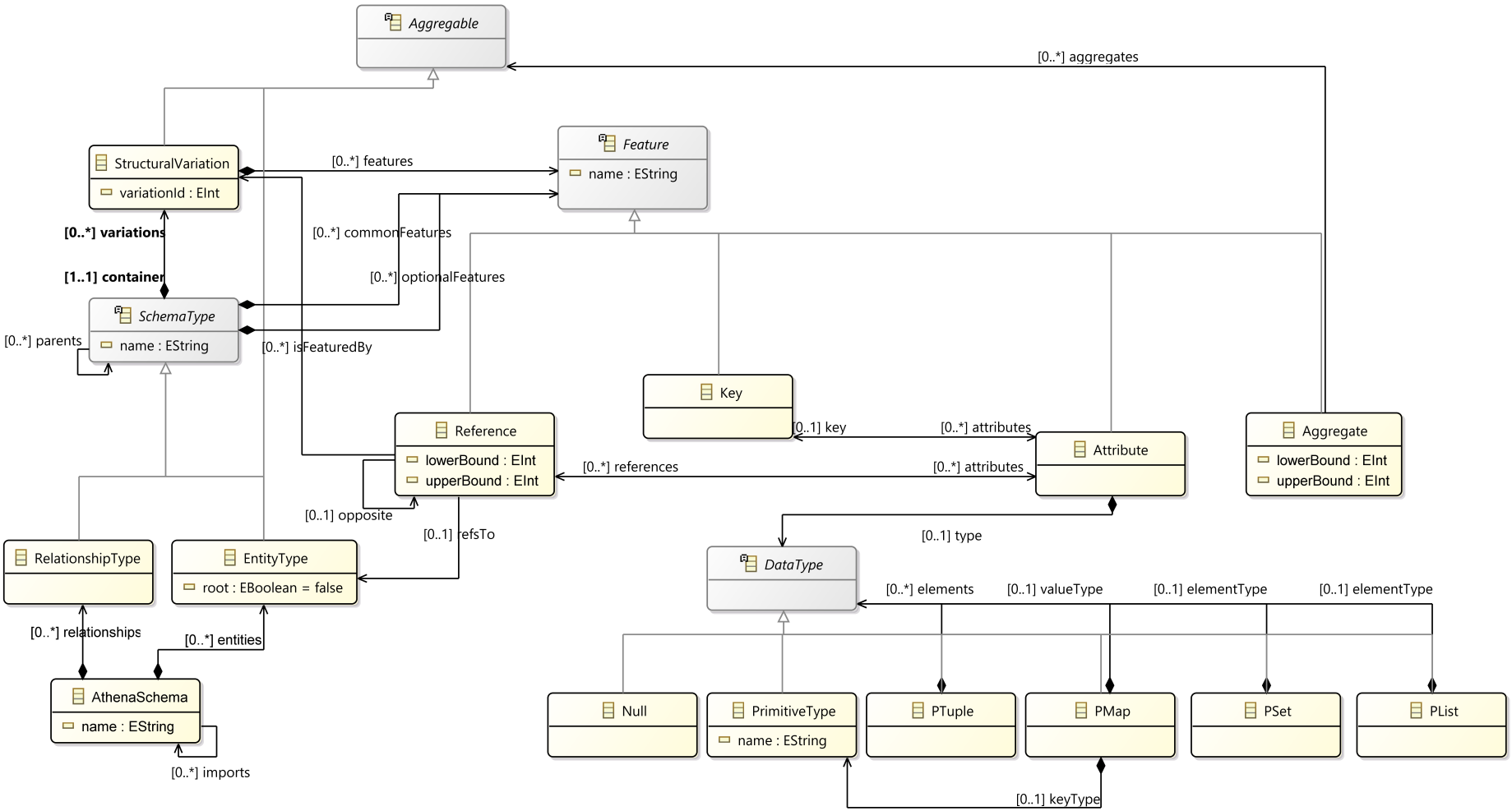
Objetivos del lenguaje

El lenguaje debe permitir...

- Soportar el metamodelo NoSQLSchema para que se puedan generar especificaciones automatizadas
- Trabajar con información incompleta:
  - Esquemas con entidades faltantes
  - Entidades con atributos faltantes
  - Atributos sin tipo
  - Atributos sin nombre
  - Entidades sin variaciones
- Permitir la definición de operaciones sobre los esquemas definidos
- Permitir la definición de datos e importación sobre los esquemas definidos
- Permitir la definición de operaciones sobre los datos definidos

Índice de contenido

Metamodelo Athena (I)



Metamodelo Athena (II)

Diferencias con NoSQLSchema

- Un *SchemaType* puede contener *features* comunes u opcionales
- Una variación no tiene *count*, *firstTimestamp* ni *lastTimestamp*
- No se da la jerarquía *StructuralProperty* ni *LogicalProperty*
- Una *Feature* no es opcional. La opcionalidad se deduce del contexto
- Un *Attribute* puede no tener ningún tipo
- Un tipo complejo puede no tener tipo primitivo

Cuestiones:

- ¿Puede una referencia no especificar a qué *SchemaType* referencia? **NO**
- ¿Puede una agregación no especificar a qué *variación* agrega? **NO**
- ¿Puede una agregación especificar un *EntityType*? **SÍ**
- ¿Puede un *Attribute* no tener nombre? **NO**

Índice de contenido

Definición de esquemas

Declaración de un esquema Athena:

- Se declara un esquema por fichero
- Ficheros con extensión *.athena*
- Palabra reservada + nombre del esquema
- El nombre del esquema será referenciado en otros lenguajes

```
schema example
```

Índice de contenido

Definición de EntityTypes y RelationshipTypes (I)

Definición de forma resumida:

- Palabra reservada y nombre de la entidad o relación
- Opción de declarar si la entidad es raíz o no
- Especificación de propiedades comunes y opcionales
- ¿Deberían poderse especificar variaciones?

Definición de forma explícita:

- Con la palabra reservada **fullspec**
- Posibilidad de especificar variaciones
- Posibilidad de especificar propiedades comunes
- ¿Deberían poderse especificar propiedades opcionales?

Definición de EntityTypes y RelationshipTypes (II)

```
schema example;

root entity Entity1 { ... },
entity Entity2 { ... },
root entity Entity3 { ... },
relationship Rel1 { ... },
relationship Rel2 { ... },
relationship Rel3 { ... },
fullspec root entity Entity4
{
  common { ... }
  // Variations
},
fullspec relationship Relationship4
{
  common { ... }
  // Variations
}
```

Índice de contenido

Definición de Variaciones y Propiedades (I)

Sobre las Variaciones:

- No se pueden especificar si la EntidadType o RelationshipType se definió de forma breve

Sobre las Propiedades:

- Si una propiedad se define en *common* entonces la propiedad es común a todas las variaciones
- En un SchemaType definido brevemente, las propiedades por defecto son comunes a todas las variaciones
- En un SchemaType definido brevemente, las propiedades pueden ser opcionales si se preceden con el carácter '*'

Definición de Variaciones y Propiedades (II)

```
root entity Entity1
  { feat1, feat2, *feat3, *feat4 },
entity Entity2
  { feat1, feat2, *feat3, *feat4 },
fullspec root entity Entity3
{
  common { feat1, feat2 }
  variation 1 { feat3 },
  variation 2 { feat4 },
  variation 3 { feat5 }
},
fullspec entity Entity4
{
  common { feat1 }
  variation 1 { feat2 },
  variation 2 { feat3 },
  variation 3 { feat4 }
}
```

Índice de contenido

Definición de Atributos, Agregados, Referencias y Claves (I)

Definiciones de Atributos:

- ¿Identificador obligatorio? **SÍ**
- ¿Tipo opcional? **SÍ**
- ¿Cómo declarar un tipo opcional? ¿Se declara con un carácter '?'? ¿No se declara nada? **Con ***
- Ejemplos de tipos: String, Number, Boolean, PList, PSet, PTuple
- ¿Cómo especificar un tipo complejo? ¿PList<>? ¿PList<?>? **Ahora mismo, con <>**

Definiciones de Aggregates:

- ¿Identificador obligatorio? **SÍ**
- ¿Cardinalidad obligatoria? **NO**
- ¿Es obligatorio especificar las variaciones a agregar? **¿SÍ?**
- ¿Se puede especificar una entidad a agregar? **SÍ**

Definición de Atributos, Agregados, Referencias y Claves (II)

```
attribute1 : Number,
attribute2,
attribute3 : PList<>,
attribute4 : PList<String>,
attribute5 : PSet<>,
attribute6 : PTuple<>,
attribute7 : PMap<String, Number>,
attribute8: Null

aggregate2 : aggr -> (Entity1_2),
aggregate3 : aggr [ ..-1] -> (Entity1_2),
aggregate4 : aggr [0..-1] -> (Entity1_2),
aggregate5 : aggr [1..1] -> (Entity1_2, Entity1_3)
```