***SEVENTH FRAMEWORK PROGRAMME***
***Smart Energy Grids***

*Project Title:*

# Integrating Active, Flexible and Responsive Tertiary Prosumers into a Smart Distribution Grid

## INERTIA

**Grant Agreement No: 318216**
**Collaborative Project**

## Deliverable

## INERTIA Common Information Model

| | | | |
|---|---|---|---|
| Deliverable No. | | **D2.1** | |
| Workpackage No. | **WP2** | Workpackage Title and task type | **INERTIA Distributed Multi-Agent Based Framework** |
| Task No. | **T2.1** | Task Title | **INERTIA Common Information Model** |
| Lead beneficiary | | **TUK** | |
| Dissemination level | | **PU** | |
| Nature of Deliverable | | **R** | |
| Delivery date | | **September 2013** | |
| Status | | **F** | |
| File Name: | | **INERTIA_Deliverable_D2.1.doc** | |
| Project start date and duration | | **October 2012, 36 Months** | |

# Authors List

| Leading Author (Editor) | | | | |
|---|---|---|---|---|
| | *Surname* | *Initials* | *Beneficiary Name* | *Contact email* |
| | Skokan | M.S. | TUK | marek.skokan@tuke.sk |
| **Co-authors** (in alphabetic order) | | | | |
| *#* | *Surname* | *Initials* | *Beneficiary Name* | *Contact email* |
| *1* | Adamopoulou | A.A. | CERTH | adamanna@iti.gr |
| *2* | Jimeno | J.J. | TECNALIA | joseba.jimeno@tecnalia.com |
| *3* | Kostelnik | P.K. | TUK | peter.kostelnik@tuke.sk |
| *4* | Tryferidis | T.T. | CERTH | thanasic@iti.gr |
| *5* | Tsatsakis | K.T. | HYPERTECH | k.tsatsakis@hypertech.gr |
| *6* | Tooren | P.T. | Almende | peet@almende.org |
| *7* | Wingstedt | U.W. | CNet | ulf.wingstedt@cnet.se |

# Reviewers List

| List of Reviewers (in alphabetic order) | | | | |
|---|---|---|---|---|
| *#* | *Surname* | *Initials* | *Beneficiary Name* | *Contact email* |
| *1* | Anadiotis | G.A. | HYPERTECH | g.anadiotis@hypertech.gr |
| *2* | Ioannidis | D.I. | CERTH | djoannid@iti.gr |

| Document history | | | |
|---|---|---|---|
| Version | Date | Status | Modifications made by |
| 1.0 | 31/5/2013 | Initial Document with TOC, Scope of INERTIA Common Information Model, Synergies and differences with other projects | TUK |
| 1.1 | 19/7/2013 | Added sections, Protocols for inter-entity communication grounded in industrial standards (section 3), Analysis of requirements to INERTIA CIM (section 4) | TUK |
| 1.2 | 9/9/2013 | Analysis of requirements to INERTIA CIM (section 4) – requirements description from component developers (HYPERTECH, CERTH, TECNALIA, CNet, Almende), impact on CIM analysed (by TUK)<br>Inertia CIM specification (sec. 5) | HYPERTECH, CERTH, TECNALIA, CNET, ALMENDE, TUK |
| 1.3 | 23/9/2013 | Correction of section 3 and 4 based on feedbacks from partners (HYPERTECH, TECNALIA)<br>Finalisation of all sections (mainly section 5) | TUK |
| 1.4 | 28/9/2013 | Modifications based on peer review reports | TUK |
| 1.5 | 30/9/2013 | Final Version | TUK |

# List of definitions & abbreviations

| Abbreviation | Definition |
|---|---|
| BIM | Building Information Model |
| CIM | Common Information Model |
| CCHP | Combined Cooling, Heating and Power System |
| CVPP | Commercial Virtual Power Plant |
| DB | Database |
| DCIM | IEC CIM extension for Distribution management domain |
| DER | Distributed Energy Resource |
| DG | Distributed Generator |
| DR | Demand Response |
| DSM | Demand Side Management |
| DSO | Distribution System Operator |
| EV | Electric Vehicle |
| EU | European Union |
| gbXML | Green Building XML |
| HVAC | Heating Ventilation and Air Conditioning |
| IAM | Information Access Module |
| ID | Identification |
| IEC | International Electrotechnical Commission |
| KPI | Key Performance Indicator |
| LD | Logical Device |
| LN | Logical Node |
| LH | Local Hub |
| LCH | Local Control Hub |
| LSVPP | Large Scale Virtual Power Plant |
| MV | Medium Voltage |
| OWL | Web Ontology Language |
| PV | Photovoltaic |
| RAW data | Data not yet subjected to analysis |
| RDFS | Resource Description Framework Schema |
| SoC | State Of Charge |
| TVPP | Technical Virtual Power Plant |
| UI | User Interface |
| UP | User Preference |
| UML | Unified Modeling Language |

| | |
|---|---|
| URI | User Resourse Identifier |
| XML | eXtensible Markup Language |
| XSD | XML Schema Definition |
| VA | Visual Analytics |
| VPP | Virtual Power Plant |

# Executive Summary

In order for the INERTIA system to fulfil its mission as a demand side management framework, it must cope with challenges represented by an information rich dynamic environment it is expected to operate within. Many separate heterogeneous and distributed information sources produce data continuously on different levels (including physical data from sensors) and granularities which should be processed in real-time as well as historical fashions. The information should be transformed, integrated, aggregated and stored in order to be understandable and accessible for all INERTIA components that need it to support their operation.

The presented deliverable represents results of Task T2.1. More specifically, it defines Common Information Model (CIM) for Local Control Hub and Aggregator Control Hub of the INERTIA Framework. The aim of CIM is to provide a model of information elements (e.g. concepts, even, relations, interfaces) used for information exchange between components as well as for modelling work performed by other tasks (e.g. DER flexibility models in T2.2 or energy-related performance models in T2.4). The CIM definition is considered as a shared vocabulary that enables to address the information needs for the INERTIA framework components. The information aspects of CIM are defined in this deliverable, while the semantic level of CIM is expected to be fully defined in the deliverable D3.2.

The work presented in the deliverable was based on a two-phase methodology approach. The first phase aimed at sources 'external' to the project. The focus was on the identification on those projects as well as standards which could be relevant for INERTIA. The information models from these other relevant projects and some of the IEC norms were analysed as a possible basis for CIM. The second phase (composed from six steps) reflected the evolution of the INERTIA architecture. The requirements from INERTIA components on storage services were the basis for the definition of CIM. Overall, the employed methodology followed a **component-centric approach** and this approach is reflected by this deliverable as well.

An inspiration was looked for in the following five projects which have been identified as relevant: Adapt4EE, Mirabel, FIEMSER, Fenix, and BeyWatch. Their information models have been analysed in order to identify overlaps, differences and possible synergies between them and CIM. In addition to the projects, two IEC standards were analysed as well. The first standard (actually consisting of several parts two of which were considered) was IEC 61970-301 and IEC 61968 known as common information model for electricity. The second standard was IEC 61850-7-420 focusing on distributed energy resources.

In order to reflect evolving INERTIA architecture, several architecture parts/components' requirements were specified (in form of interface specifications) and analysed. Both levels of the architecture (local hub level as well as aggregator level) have been considered. The following components were included in the analysis:

- for **Local hub CIM**: Linksmart middleware, Occupancy prediction and flow, User profiling, Ambient user interface, Key performance indicators (indicator groups related to energy, flexibility, business, comfort, and grid), Multi agent based local hub gateway, Local hub DER flexibility modelling;

- for **Aggregator control hub CIM**: Multi agent based prosumers control and optimization, Grid coordination and DR activation, Aggregator analytics.

The requirements of these components have been analysed from the point of their data models, interfaces and their methods, as well as their impact on CIM (represented by the IAM component within the architecture).

Subsequently, based on the analysis, elements of CIM have been defined. They were defined for all those components that intend to interact with the IAM component (those components enumerated above). The definition of CIM has been produced in the form of **IAM interfaces, XSD schemas** (both informal graphical visualisation as well as extensive formal definitions of elements and complex types are provided), **DB schemas**, and **references to related semantic models** (BIM and DER semantic models which can be found in the deliverable D3.1). Similarly to the analysis, CIM specification is divided into two parts – specification of Local hub CIM and specification of Aggregator control hub CIM.

The presented deliverable reflects the current state of INERTIA CIM as it is. Although the task devoted to the development of CIM finishes, this actual form of CIM is not guaranteed to be final, since there are other tasks running which may impact the CIM and induce its modifications in next project period.

# Table of Contents

# List of figures

# List of tables

# 1. Introduction

The INERTIA Common Information Model (CIM) will define the high level domain model comprising the basic elements (semantic concepts, events, relations, interfaces etc.) underlying the INERTIA demand side management framework [1].

CIM in computing is open standard that defines how managed elements in an IT environment are represented as a common set of objects and relationships between them [2]. CIM Specification consists of architecture and concepts of CIM, language (by which the CIM Schema is defined), and a method for mapping CIM to other information models. The CIM architecture is usually object-oriented. The CIM elements are typically represented as classes and any relationships between them are represented as CIM associations. Inheritance allows specialization of common base elements into more specific derived elements. The CIM Schema is conceptual schema which defines the specific set of objects and relationships between them that represent a common base for the managed elements in an IT environment.

The INERTIA CIM is specified by the inputs/outputs interfaces between INERTIA components and the Information Access Module (IAM – serving as repository). This specification is formalized by signatures of IAM services, XSD schemas defining data types used in this services, database tables specification for dynamic data storage and semantic model including all static information needed by INERTIA components.

## *1.1 Scope of INERTIA CIM*

In general, the INERTIA CIM is a standard that defines common set of INERTIA specific data objects and relationship between them. The CIM specification is formalized as semantic model (conceptual schema) including all information which is needed by INERTIA components, data structures, description of data storages meta-data (i.e. which information is stored where, what does the stored information contain, the information format, etc.). INERTIA CIM serves as a vocabulary in communication between any components of the INERTIA framework and Information Access module. The scope of INERTIA CIM is to provide information and semantic model for the domain objects used be the INERTIA components. Based on the description of these components from the INERTIA report D1.3 here is the outline of these objects:

- Events and RAW data (dynamic data)
    - Occupancy changes/unexpected occupancy
    - Occupant movement
    - Device/DER operation related events
    - Environment (e.g. visual, thermal)
- Occupancy flow and prediction (dynamic and static data),
    - Historical, Real time, Predicted occupancy
    - Occupancy schedules

- Occupant and profiles (dynamic data and static data)
    - Occupant preferences
    - Group preferences
- Flexibility
    - Forecast of DER flexibility
    - Forecast of Aggregation Hub Flexibility
- Structure for Aggregator Hub, Local Hub and Buildings (static data)
    - Group of Facilities, Facilities, Buildings, Spaces and connections between them (topology)
- DER related (static model of devices and aggregated devices dynamic data DER operation, DER flexibility),
    - Generators – local generators using renewable energy source (e.g. solar energy, wind energy etc.)
    - Storages – local storage systems (e.g. batteries)
    - Loads – consumption of local equipment and appliances (e.g. light, HVAC, Fridge, PCs etc.)
- WSN and equipment devices:
    - taxonomy and description of IoT devices, sensors
    - association of devices to the contextual information, such as location
- Key Performance Indicators (KPIs) (static and dynamic data)
    - KPI obtained from simulations related to static elements (e.g. space, building, etc.)
    - Energy related, Business related, Flexibility and capacity, Comfort related, Grid related
- DR signal
    - Grid Coordinator & DR Activation data model
- Units (static data)
    - Units for measurement and computed data values

The domain objects are described by static data or by dynamic data. The static data (e.g. data about building structure, profiles of equipment etc.) will serve for interpretation of dynamic data (i.e. the RAW/Event data about the use of specific equipment in a given building in a given time). The INERTIA IAM enables to combine such dynamic data and static data using the semantic relations defined in the oncology. IAM process requests from the INERTIA framework components (the requests are based on CIM specification) and store data in the hybrid repository. The response to these requests are serialised (based on CIM) and send over the IAM exporting services to the requesting INERTIA component. This process results in semantically enriched (real time) data ready for analysis within the INERTIA framework. Such analysis will aim at occupancy aware prediction of (aggregated) DER capacity and flexibility.

## *1.2 The Structure of the Deliverable*

The presented deliverable is structured and organised in the following chapters:

- **Chapter 2** presents a methodological approach employed to organise activities aiming at the specification of INERTIA CIM. A two-phase methodology based on component-centric approach is introduced.

- **Chapter 3** identifies other relevant projects and analyses relations between the information models used within these projects and INERTIA CIM. Five different projects have been identified and analysed.

- **Chapter 4** analyses relevant industrial standards, focusing on standards related to CIM in electricity and standards related to distributed energy resources. Standards IEC 61970-301, IEC 61968 and IEC 61850-7-420 were considered.

- **Chapter 5** provides analysis of architectural parts/components from the point of their data requirements. Altogether ten components were investigated and their data models, interfaces and impacts on CIM are presented.

- **Chapter 6** specifies elements of CIM on two levels – Local hub CIM and Aggregator control hub CIM. The definitions have the form of IAM interfaces, informal XSD schemas, DB schemas and references to related semantic models.

- **Annex** presents detailed extensive definitions of XSD elements and complex types. The definitions have a formal character.

# 2. Methodology

The INERTIA Common Information Model is designed with the aim to be able to deal with a large amount of real-time information continuously acquired from several heterogeneous sources. According to DoW [1], the initial idea of the CIM was to:

- enable to translate information in heterogeneous formats into a format understandable to all INERTIA components,
- enable to enrich data from sensors using knowledge represented in semantic models, so that data can be used to support predictability on both a local but also on a global scale incorporating all types of DER,
- describe information sources using the vocabulary that is used by all the INERTIA components,
- define a high level domain model comprising basic elements (e.g. semantic concepts, events, relations, interfaces etc.) employed by the INERTIA demand side management framework,
- define a format that will be accepted and used by all the project partners for straightforward translation from specification to the implementation phase.

The methodological approach provides a guideline for deriving information requirements that need to be reflected in the CIM and ensuring that the above mentioned points are properly reflected at the design of CIM. At first, activities were divided into two phases:

**Phase 1 Analysis of relevant approaches and standards**

In the first phase of T2.1 (during M3-M6 period), there were naturally no requirements from the component developers, since initial specification of architecture was under development and was not stabilised yet. Therefore, the attention was aimed at literature and relevant R&D projects. First activity aimed at identifying those projects which could be relevant to the INERTIA project - those projects that deals with similar topics and had already defined their own information models. Subsequently, these models were analysed with the intention to identify possibly overlapping parts. As a parallel activity, the industry standards that could bring possible invigorating inspiration were identified. The identified industry standards were then analysed with similar intention as employed in case of identified projects. The results from both analyses were then documented as it can be seen in this deliverable (section 3 and section 4).

**Phase 2 INERTIA requirements analysis**

In the second phase of T2.1 (roughly represented by M6-M12 period) the activities were aimed at the evolving INERTIA architecture. The methodology adopted during this phase is presented in the Figure 1. The focus was on components of the architecture that are relevant to the CIM – the components that need to interact with the central data repository represented by the IAM component. The employed methodology is therefore based on **component centric approach** (in opposition to sometimes used data centric approach).

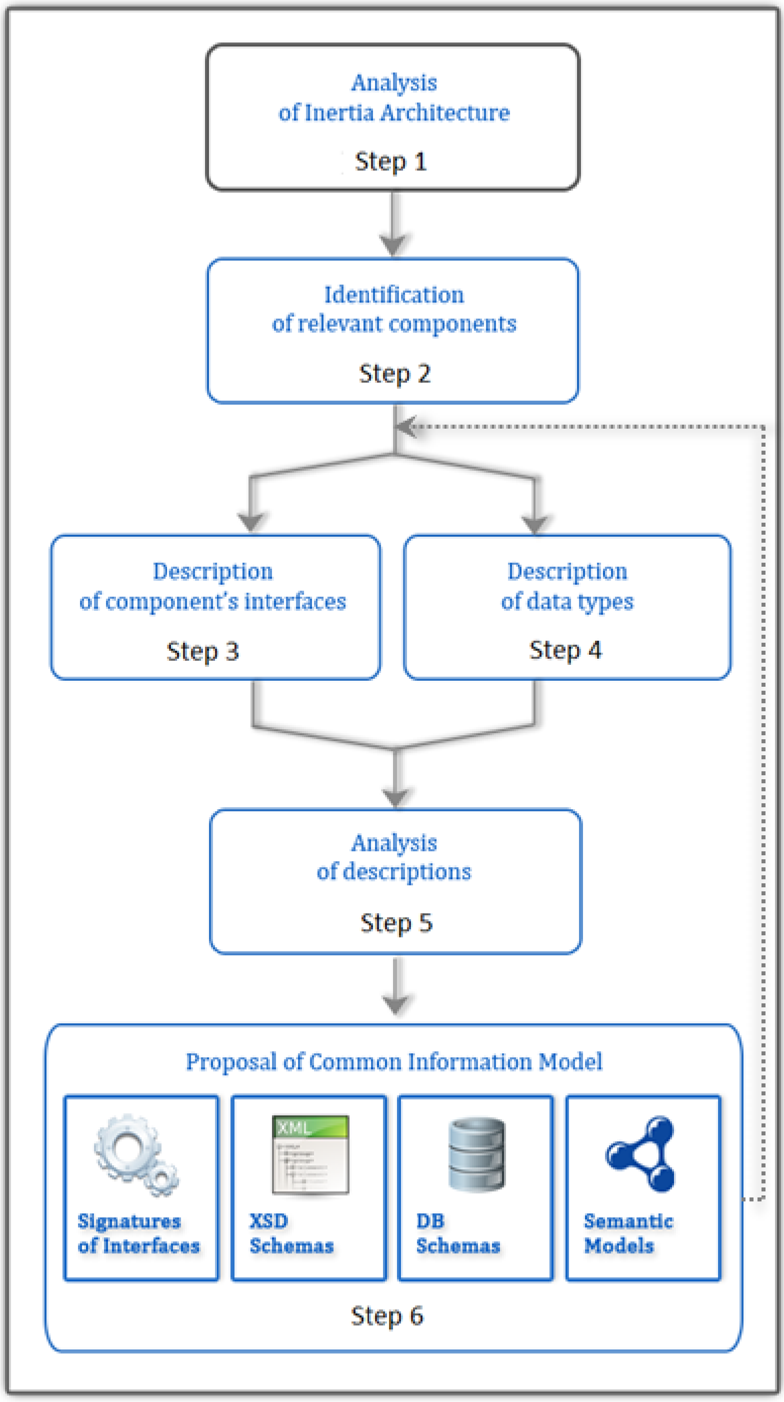



***Figure 1: Steps of the methodology adopted during INERTIA requirement analysis (Phase 2)***

## Step 1 Analysis of INERTIA architecture

The main goal of these activities was to clearly define the role of the INERTIA CIM in the architecture as a common shared vocabulary. The vocabulary enables to access at data (both in read and directions) in a unified way. The data access service is based on the vocabulary – the service itself is provided by the Information Access Module (IAM) which is described completely in the deliverable D1.3.

### Step 2 Identification of relevant components

Different architecture components have different needs for data to be consumed or produced by these components. Thus, all the defined components were analysed from the point of view of their data flow requirements in order to identify those INERTIA components that could benefit from using the IAM component to store in and retrieve data from. The data flow was investigated according to deliverable D1.3. The description of components in D1.3 is containing part about dependencies with other components and brief description of component I/O interfaces. Moreover, the flow of data among the INERTIA components is defined by UML sequence diagrams (in D1.3) that are part of use-cases descriptions. All of these descriptions were used for the identification of relevant components in this step.

### Step 3 Description of component's interfaces

After having identified an initial list of components utilising the IAM, the partners responsible for the development of the identified component were requested to define the specification of interfaces enabling storing data to or retrieving data from the IAM. Table 1 was used for the description of component's interfaces. The specification of the interfaces can be found in Section 5.

***Table 1: Specification of component's interfaces***

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

### Step 4 Description of data types

It is expected that specified interfaces use many calling parameters as well as returning parameters that are of INERTIA specific data types. These data types were also specified by developers of the investigated components. Table 2 was used for the description of data types. The specification of the data types can be found in Section 5.

***Table 2: Specification of INERTIA types***

| Type name | Attributes | Type | Description |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

**Step 5 Analysis of descriptions**

Specifications of interfaces and data types used in the interfaces formed the main input into the analysis of requirements. Thus, this analysis followed a component centric approach. For each investigated component the specification of its interfaces were analysed and the respective interfaces of the IAM component, that will serve component's interfaces, were defined (if the interface is analogous to the interface of the component, e.g..an importing interface of a component is reflected as an exporting interface of IAM, it is called a mirroring interface). If the analysis of generated descriptions revealed that something is missing in them it was necessary to go back to step 3 or step 4 in the methodology. The analysis of requirements can be found in Section 5.

**Step 6 Proposal of Common Information Model**

**Step 6.1 Signatures of the interfaces**

By processing all the component interfaces of the IAM component, their names and signature are defined. These specifications of the IAM interfaces enable to read/store data for those INERTIA components that need data access. The signatures of the interfaces are the first part of the common information model. The signatures of the interfaces can be found in Section 6.

**Step 6.2 XSD specification of data types**

The next step was the analysis of the data types used in the interfaces. Thus, specifications of the calling and returning attributes were analysed. For each component one or more specific data types were proposed. This specification has to define data type and at the same time it has to be easily deployable during the development phase by all INERTIA components. Since all of the component developers agreed to use data in XML format in their interfaces and since the most suitable specification of the data types for XML data are XML schemas, the XML schemas were defined to specify the INERTIA data types. The XSD specifications can be found in Section 6.

The signatures of the interfaces with the known XSD specification of the data types used in them enable to formally Access any INERTIA specific data. To be able to provide such data to the requesting component over such interface or to be able to store them, the data have to be seamlessly and quickly handled and combined in the IAM. This task is enabled using hybrid repository consisting of relational repository with semantic repository.

**Step 6.3 DB schema specification**

The DB repository enables to quickly store and retrieve dynamic data such as RAW data. But these data are too simple to serve the requesting component to perform any intelligent decision.

To be able to use these RAW data for predicting states and actions in the real word, they need to be enhanced with metadata. The metadata enable to process the physical data in an intelligent way. Semantically annotation of RAW data to the instances of semantic models supports their enrichment with situational context. The definition of DB schemas with the fields containing semantic annotation is third part of the CIM specification. The DB schemas are specified in Section 6.

### Step 6.4 Semantic model specification

The semantic model defines static data (data that are not time-dependent). The DB records contain the reference of such instance in one of the field defined by the DB schema.

The definition semantic models are forth part of the CIM specification.

Note that full specification of semantic models is created within works on task T3.3 - INERTIA Ontology, Inference Rules & Semantic Devices. Thus, the full specifications of the semantic models can be found in D3.2. The specifications of the prototype semantic models are presented in D3.1.

In case the process of CIM definition revealed that something is missing, it was necessary to go back to step 3 or step 4 in the methodology.

Note, since the INERTIA modelling activities are not closed yet, it is not the aim of this methodology (and this deliverable) to cover complete dependencies between stored information in the IAM and requested information from IAM.

# 3. Information models from other relevant projects

This section presents differences and possible synergies between INERTIA CIM and information models and/or concepts defined in the other relevant projects. This section was created before the requirements on the INERTIA CIM from INERTIA components were specified. Therefore relations between information models (Inertia CIM and IM used in other project) are expressed using the information objects of INERTIA CIM presented in Section 1 and available sources from web pages of the chosen projects (mainly public deliverables). Before the relation is described each project is shortly presented.

## *3.1 Adapt4EE project*

Adapt4EE aims to develop and validate a holistic energy performance evaluation framework that incorporates architectural metadata (BIM), critical business processes (BPM) and consequent occupant behaviour patterns, enterprise assets and respective operations as well as overall environmental conditions. The Adapt4EE framework, having as a central point of reference the occupancy behaviour (presence and movement) will align energy consumption points to all interrelated enterprise aspects (business processes, enterprise assets and utility state and operations) [3]. Common vocabulary is used in the Adapt4EE framework for the information exchange. This vocabulary is defined by Adapt4EE Common Information Model – CIM. CIM provides information elements and relations between them specified by XML schemas. CIM enables to interpret heterogeneous data from the Adapt4EE framework information sources (i.e. BIM, BPM, Events, etc.) for their further use (e.g. for occupancy analysis, agent simulations). Adapt4EE framework contains component (CIM Interface Module) that uses CIM elements when providing information services for Adapt4EE framework components. CIM is described in the Adapt4EE deliverable D2.2 [4].

The following table presents differences and possible synergies between INERTIA CIM and Adapt4EE CIM.

***Table 3: Differences and possible synergies between INERTIA CIM and Adapt4EE CIM***

| **Inertia CIM elements** | **Synergies**<br><br>(Adapt4EE Reusable information models) | **Differences**<br><br>(absent in Adapt4EE information models) |
|---|---|---|
| DERs and their aggregation model<br>• Generators<br>• Storages<br>• Loads | Loads (partly)<br>Attributes of:<br>• Equipment<br>• HVAC<br>• Lighting<br>HVAC and Lighting with other Adapt4EE elements relevant for Inertia | Aggregation<br>• Generators<br>• Storages<br>• Loads - Appliances used in INERTIA Pilot applications (e.g. Indesit appliances) |

| | | |
|---|---|---|
| Building<br><br>Spaces and connection between them (building topology) | Building – both attributes and relations defined in Adapt4EE BIM model are well reusable<br><br>BIM relations with Business Process Model are extra/not needed | Relations with Hub's topology |
| Occupancy flow and prediction<br><br>• Occupancy models with schedules<br><br>• Flow model | Occupant characteristics defining working hours | No statistical view on building occupancy in terms of schedules of overall building occupancy (min, max, avg., first arrival, last departure etc.) for occupancy prediction purposes.<br><br>No occupancy flow models. |
| Events and RAW data<br><br>• Occupancy changes/unexpected occupancy<br><br>• Occupant movement<br><br>• DER related events<br><br>• Environment/weather | Event types<br><br>• Occupant movement<br><br>• Equipment, HVAC, Lightning usage<br><br>• Space/environment | Occupancy changes/unexpected occupancy<br><br>DER operational events (generators, storages, specific events of equipment and appliances) |
| Environment/Weather<br><br>• Typical weather conditions | Typical weather conditions for the location | |
| Key Performance Indicators (KPIs) (static and dynamic data)<br><br>• KPI obtained from simulations (energy performance, comfort) | Only the division of KPI into energy performance and comfort with some parameters might be reusable. | It depends on definition of INERTIA KPIs |
| Units<br><br>• Units for measuremed and | Adapt4EE units model is well reusable | Specific INERTIA units needed to express DER capacity and flexibility. |

| | | |
|---|---|---|
| computed data values | | |
| WSN and equipment devices:<br>• taxonomy and description of devices<br>• association of devices to the contextual information, such as location | Sensor type definition with its location | Taxonomy of devices |

### *3.1.1* Conclusion

The re-usable parts of Adpat4EE CIM are models for Occupancy, Units, Events, BIM and probably parts of KPIs, weather model and profiles of Equipment, HVAC and Lighting. While the models covering Business aspects, Aggregator Hub topology, Prosumer profile, DER (generators and storage systems) and Flexibility Capacity as well as DR signal and strategies are missing in the Adapt4EE CIM.

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the parts that were re-used from the Adapt4EE CIM were the XSD specifications of Units, and partly BIM model including Space model and Sensor model. No Adapt4EE KPI definition was reused as INERTIA KPIs contains many different aspects.

## *3.2 Mirabel project*

Mirabel is FP7 project with main goal to develop an approach on a conceptual and an infrastructural level that allows energy distribution companies to balance the available supply of renewable energy sources and the current demand in ad-hoc fashion. Many Renewable Energy Sources (RES, e.g., windmills, solar panels) pose the challenge that production is dependent on external factors (wind speed and direction, amount of sunlight, etc.). Hence, such available power is rather difficult for energy distributors to efficiently include into their daily schedules. Mirabel aims to develop concept of micro-requests with time shifts to handle the demand and supply of energy on a household level. Further, it defines methods to predict the energy supply and demand in the small (i.e., for households) and in the large and to update predictions over time. The Mirabel system will aggregate (and disaggregate) the micro-requests on a regional level and will use a scheduling approach for energy production and consumption based on aggregated requests.

The following modules comprise in the Mirabel system:

- FlexOffers: Allow Prosumers to fully automatically indicate their energy flexibility to their BRP.

- Aggregation: Groups thousands of similar FlexOffers to manage the large amount of data in a distributed hierachical Energy Data Management System.
- Forecasting: Is used to predict demand and supply of renewable energies for the upcoming hours and days.
- Scheduling: Computes to optimal plan to dispatch flexible energy demand and supply to minimize costs for the BRP.
- Negotiation: Finds the optimal price in real-time that maximize the financial benefit for the customer and the profit for the BRP.

The system uses Mirabel Common Data Model – CDM. CDM provides data models in the form of UML class diagrams. CDM enables to interpret data within the Mirabel system. CDM is described in the Mirabel deliverable D2.3 [5].

The following table presents differences and possible synergies between INERTIA CIM and Mirabel Common Data Model.

***Table 4: Differences and possible synergies between INERTIA CIM and Mirabel CDM***

| **Inertia CIM elements** | **Synergies**<br><br>(Mirabel Re-usable information models) | **Differences**<br><br>(absent in Mirabel information models) |
|---|---|---|
| DERs and their aggregation model<br>• Generators<br>• Storages<br>• Loads | Some characteristics of:<br>• Fixed energy profile<br>• FlexEnergySchedule class<br>• Durations (min, max) and dependencies (for DER):<br>• EnergyConstraintProfile class | No DER concept introduced in the model<br><br>No division of DER into Generators, Storages, Loads and no operational properties of these DERs. |
| Aggregator Hubs basic topology | Place<br>• MeteringPoint class | Basic topology |
| Units<br>• Units for measuremed and computed data values | IEC CIM derived classes:<br>• ActivePower<br>• UnitSymbo<br>• UnitMultiplier | Occupancy related units |
| Business related | Classes for negotiating flexibility in energy supply and demand | |

| | | |
|---|---|---|
| • Flexibility<br>• Capacity<br>• Money with currency | • FlexEnergy<br>• EnergyConstraintProfile<br>• TariffConstraintProfile<br>• FlexEnergyState<br>• MeteringPoint<br><br>Offers on a Marketplace and its acceptance is modelled in these classes:<br>• FlexOffer class<br>• FlexOfferAcceptance class<br><br>Class that associates FlexOffer with each level of flexibility - FlexOfferAssignment<br><br>IEC CIM derived classes:<br>• Money, Currency | |

### *3.2.1* Conclusion

Some Mirabel data models have potential to be re-used in INERTIA CIM Business models, models of DER and models of Units. Most of the INERTIA CIM models can be not inspired by Mirabel CDM (Occupancy, User profile, Events, Building, KPIs, GRID topology, Weather).

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the conclusion is: Even though the INERTIA project partner dealing with model of Flexibility was aware of the Mirabel data model of flexibility, the model did not influence the data model design since the approach there is different in some ways.

## *3.3 FIEMSER project*

FIEMSER project focuses on “Use less energy” and “Make more energy locally” and it also provides the necessary conditions and platform for future developments to sell surplus energy.

The main objective of this project is the development of an innovative energy management system for existing and new residential buildings, which pursues the increase of the efficiency of the energy used and the reduction of the global energy demand of the building, but without penalizing the comfort levels of the users. To the achievement of this goal, it will follow two main strategies:

- Minimizing the energy demand from external resources, through the reduction of the energy consumption in the building and the correct management of local generation (heat and electricity) and energy storage equipment to satisfy the energy demand of the building, and even provide the capability to export energy to the utilities when needed.

- Interaction with the building user, in such a way as to increase the consciousness of the consumer of his energy consumption and CO2 emissions, providing hints to make punctual changes in his behaviour without major disruptions of his comfort conditions.

FIEMSER project introduces the Data Model formalized in UML diagrams and describing the object/relational mappings between the data components, which are persisted using the frameworks, such as Hibernate. Following the definition of use cases and data to be exchanged between system parts to perform the FEIMSER processes, the 8 main data categories were identified:

Environmental and contextual data

- WSN-related data
- User preferences
- Resources scheduling data
- Advices
- Energy performance indicators
- Energy-focused BIM (Building Information Model)
- User Access Rights

The following table presents differences and possible synergies between INERTIA CIM and FIEMSER Common Data Model.

***Table 5: Differences and possible synergies between INERTIA CIM and FIEMSER CDM***

| **Inertia CIM elements** | **Synergies**<br><br>(FIEMSER Re-usable information models) | **Differences**<br><br>(absent in FIEMSER information models) |
|---|---|---|
| DERs and their aggregation model<br><br>• Generators | Energy-focused BIM:<br><br>• models of loads, generators and storages associated with the parts of | |

| | | |
|---|---|---|
| • Storages<br>• Loads | the building.<br><br>Models of operation mode and losses. | |
| Building<br>• Spaces and connection between them (building topology) | Energy-focused BIM:<br>• space organization<br>• building partitions, as an extension to gbXML standard. | Relations with the Hub topology. |
| Occupancy flow and prediction<br>• Occupancy models with schedules<br>• Flow model | Resource scheduling:<br>• description of resource planning for the parts of the building associated also with user preferences,<br>• estimation of temperature evolution and,<br>• schedule of device usage. | The occupancy related information. |
| Occupant and procurer profiles (static data)<br>• Occupant preferences | User preferences:<br>• usage profile,<br>• definition of scenes, including comfort set-points and<br>• use of appliances, control rules and energy strategy | |
| Events and RAW<br>• Occupancy changes/unexpected occupancy<br>• Occupant movement<br>• DER related events<br>• Environment/weath er | WSN-related data:<br>• data collected from sensors - logs of control orders sent to devices | Events except the data collected from devices. |
| Environment/weather<br>• Typical weather conditions | Environmental and contextual model:<br>• location, climate zone, shadowing, building | |

| | | |
|---|---|---|
| | orientation, weather forecast | |
| Key Performance Indicators (KPIs) (static and dynamic data)<br>• KPI Etalons<br>• KPI obtained from simulations (energy performance, comfort) | Energy performance indicators:<br>• usage of devices,<br>• consumption or generation of the building partitions<br>• overal building performance. | |
| Business related<br>• Documents, agreements, Price<br>• Flexibility<br>• Capacity | Environmental and contextual model:<br>• energy prices associated with the building partitions | |
| WSN and equipment devices:<br>• taxonomy and description of devices<br>• association of devices to the contextual information, such as location | WSN-related data:<br>• sensors and actuators (locations, charactereistics) | |

### *3.3.1* Conclusion

There is reasonable overlap between the INERTIA and FIEMSER data model, mostly in the areas of models for: DERs, building description, models of devices/equipment, device events, user profiles and forecast information. However, FIEMSER data model is dedicated for saving the energy in the building and has no relation to the any Aggregation elemnent. The building occupancy information is also not covered.

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the conclusion is: FIEMSER data models did not influence the INERTIA CIM since the FIEMSER approach was not focused on occupancy information.

## 3.4 Fenix project

The objective of Fenix project is to enable DER making the EU electricity supply system cost efficient, secure and sustainable through aggregation of DERs into Large Scale Virtual Power Plant (LSVPP).

The Fenix system defines Virtual power plant concept (VPP) and Technical Virtual power plant concept (TVPP) that are designed to achieve project objective. VPP has the capability to flexibly aggregate attributes from a multitude of distributed energy resources (DER) by use of FENIX remote control intelligence. DER comprises renewable power plants, CHP plants, flexible loads and electricity storage devices, connected to distribution networks. DER attributes may encompass commercial inputs, products and/or services or relevant information for network management. DER inputs, products and services may relate to wholesale markets for energy commodities, ancillary network services and, when applicable, valuable market stimulation benefits and electricity generation attributes. The control centre functionalities are defined of an entity aggregating DER-associated products and services for creation of additional commercial value, using intelligence for remote control of flexible DER [6]. Technical virtual power plant (TVPP) creates a single operating profile from a composite of the parameters characterizing each DER. Thus it enables GRID operators to operate aggregated DERs as one Power Plant. Namely, TVPP Services and functions enable local system management for Distribution System Operator (DSO) and providing Transmission System Operator (TSO) system balancing and ancillary services. Commercial VPP (CVPP) defines also contract models to regulate the inter-relationships between TSO, DSO, VPP and DER operators.

The FENIX system addressed following ancillary services:

- Active power / frequency control
- Reactive power / voltage control
- Congestion management
- Reduction of power losses
- Improvement of power quality
- Black start / islanded operation (grid-forming)

There are following models of DER defined:

- Wind turbine generator (WTG) systems,
- Photovoltaic (PV) systems,
- Hydroelectric power (Hydro) stations,
- Combined cooling, heating and power (CCHP) systems,
- Storage systems

The implementation of a VPP requires monitoring and/or control of DER. Fenix BOX (FB) provides proxies for metering, communications and data management capable to provide information to the aggregation. The overview of FENIX system architecture is on Figure 2.

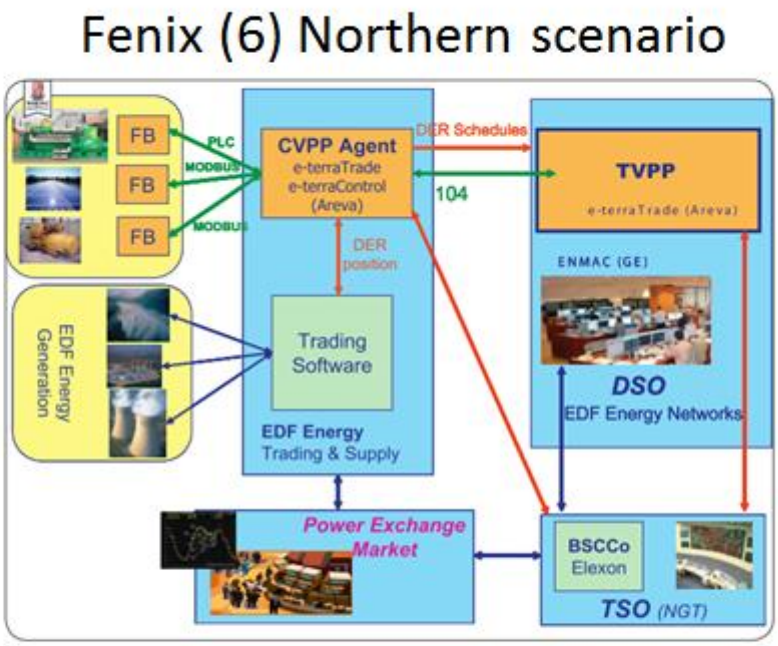



***Figure 2: Overview of FENIX architecture***

Note that there were no reports describing common data structures in FENIX. Therefore we are reflecting possible synergies only (it is not possible to know about the differences since no complete information was analysed) with main concepts mentioned and explained in the publically available Fenix documents [6], [7] and [8]. The possible synergies are following.

***Table 6: Possible synergies between INERTIA CIM and FENIX concepts***

| **Inertia CIM elements** | **Synergies**<br><br>(Fenix Re-usable information models/concepts) |
|---|---|
| DERs and their aggregation model<br><br>• Generators<br><br>• Storages<br><br>• Loads | Concept of VPP, TVPP:<br><br>• DER comprises renewable power plants, CHP plants, flexible loads and electricity storage devices, connected to distribution networks.<br><br>• DER operator: baseline business model |
| Aggregator Hub | DER attributes may encompass |

| | |
|---|---|
| Basic topology | information for network management. |
| Business related<br><br>• Flexibility<br>• Capacity<br>• Price and currency | Concept of Commercial VPP<br><br>• Commercial aggregator: baseline business model<br>• CVPP: FENIX business model<br>• DER attributes may encompass commercial inputs, products and/or services or relevant information for network management.<br><br>TSO baseline business model<br><br>DSO baseline business model<br><br>DER operator: FENIX business model<br><br>Fenix contractual structure |

### *3.4.1* Conclusion

The problem to describe possibly re-usable information elements defined in Fenix is fact, that reports describing data structures in details are not publically available at project web page. Therefore we were limited in reflection of Fenix concepts. The concepts of VPP TVPP and CVPP together with the Fenix business models have potential to be reused in the INERTIA CIM.

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the conclusion is: Fenix concepts did not influence the INERTIA CIM. Instead of using concepts derived from Fenix VPP the flexibility of Aggregator Hubs are defined.

## *3.5 BeyWatch project*

Among others, the following objectives were stated for BeyWatch project:

Study and specify the energy consumption and power demand models

Design and develop the BeyWatch "Agent" platform. This will be an HGi based, sophisticated, yet low-cost monitoring/remote-control platform, integrating both an intelligent electricity metering device, a remote-managed appliances/white-goods controller and a communication module

Design and implement the BeyWatch "Supervisor" system. This will be an open and distributed system/platform for the management and control of energy consumption and power demand at a local/square or urban geographical level.

Study, analyse, evaluate and validate the results, as well as the conduction of case studies and business models for the wider adoption of the platform and the proposed services

Inertia project will spent effort for constructing power demand models for monitoring and remote control platform integrating metering sensor data and remote-management of appliances, management system for consumption and power demand at a local/square or urban geographical level as well as for analysing, defining and evaluating of business models.

BeyWatch's solution to this problem is based on:

- minimizing white goods' and consumer electronic devices' energy consumption

- balancing energy distribution

The D2.2 “End-to-end Platform Specification” [17] provides Quick Reference to the information models. D2.2 defines End-To-End Platform specification. D4.3 describes Middleware adaptation & user preferences profiling mechanism. Due to lack of description of all information models using UML modelling approaches, we were not able to investigate all information elements defined by BeyWatch projects. Therefore we are reflecting only Synergies with information models found in BeyWatch documents in the following.

***Table 7: Possible synergies between INERTIA CIM and BeyWatch models***

| **Inertia CIM elements** | **Synergies**<br><br>(BeyWatch Re-usable information models/concepts) |
|---|---|
| DERs and their aggregation model<br>• Generators<br>• Storages<br>• Loads | State models of:<br>• Washing Machine, Dishwasher, Refrigerator, Electricity Meter<br><br>BeyWatch Devices Common APIs<br><br>PV generators and micro Wind generators mathematical models parameters |
| Events and RAW<br>• Occupancy changes/unexpected occupancy<br>• Occupant movement<br>• DER related events | Device management and event API specification |

| | |
|---|---|
| • Environment/weather | |
| Aggregator Hub<br><br>Basic topology | BeyWatch Supervisor Specifications |
| Business related<br><br>• Flexibility<br><br>• Capacity<br><br>• Price and currency | BeyWatch Supervisor Specifications |

### *3.5.1* Conclusion

The problem to describe possibly re-usable information elements defined in BeyWatch is lack of description of all information models using standard modelling approaches (e.g. UML class diagrams). Therefore we were limited in reflection of BeyWatch informations models. The static models of appliances BeyWatch supervisor specification and Device management and event API specification have potential to be reused in the INERTIA CIM.

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the conclusion is: Even though the INERTIA project partner dealing with model of Operations of devices was aware of the BeyWatch data model of states of devices, the model did not influence the data model design since the approach there is different in some ways.

# 4. Industrial standards analysis

This section presents two industry standards. These standards were analysed before the requirements from INERTIA components were specified (i.e. during phase 1 according to adopted methodology). Two standards were chosen in cooperation with project consortium and analysed during this period. The first chosen standard is known as Common Information Model (CIM) for electricity - IEC 61970-301 and IEC 61968. The second chosen standard is known as Substation configuration language (SSL) – IEC 61850. In case of SSL standard the part for Distributed Energy Resources was analysed - IEC 61850-7-420.

## *4.1 CIM in electricity*

While CIM in computing is open standard that defines how managed elements in an IT environment are represented as a common set of objects and relationships between them, CIM in electricity represent common set of objects and relationships between them for the IT environment of the electric power industry. The CIM in electricity is defined by International Electrotechnical Commission (IEC). It aims to allow application software to exchange information about the configuration and status of an electrical network. The definition of this CIM is maintained as a UML model (UML model is not suitable for INERTIA CIM see the Gaps and threats at the end of this section). The UML model (of IEC CIM) defines a common vocabulary and basic ontology for aspects of the electric power industry. The central package within the CIM is the Wires model. The Wires model describes the basic components used to transport electricity. Core packages of the CIM are defined in IEC 61970-301. They focus on the needs of applications for electricity transmission including energy management system, SCADA, planning and optimization. The IEC 61970-501 and 61970-452 standards define an XML format for network model exchanges using RDF. The IEC 61968 series of standards extend the CIM to meet the needs of electrical distribution, where related applications include: Distribution management system, outage management system, planning, metering, work management, geographic information system, asset management, customer information systems and enterprise resource planning.

### *4.1.1* IEC 61970-301

Standard IEC 61970-301 covers the core package of the CIM. The following part describes CIM defined in IEC 61970-301 that is of potential interest for INERTIA CIM. For more information about IEC 61970-301 please see INERTIA internal SoTA document.

The class dependencies in the CIM general package indicate that the central classes are Wires, Core and Resource (most of the classes are related to them).

**Domain package**

The domain package defines data types that are used in other packages. It contains data types for the relevant units in the domain of electricity power (e.g. for measurements, costs, time etc.).

The model of Units has overlap with Units for measurements used in INERTIA CIM.

**Core package**

The Core package models core of Power System Resource. For that it specifies relations among equipment with their connectivity, geographical regions, region's substations and voltage levels and schedules (irregular and regular). The overlap with the INERTIA CIM is limited, since equipment (DER) modelled on the lowest level is connected with building space while aggregated DER can be described by the location in terms of region and/or Grid topology. The NonConformLoads entities might correspond to the representation of the Local Hubs of Inertia.

In addition if there is need to model specific location information according to some coordinate system the classes to use are Location. PositionPoint and CoordinateSystem that are associated to any PowerSystemResource might be an interesting source.

**The Production**

The Generation (Production) package connects schedules, wires, costs curves with classes for various generators (hydro, thermal, wind, nuclear). The information about generators are further specified by UML models (defining e.g. fuel, emission curves etc.).

**LoadModel package**

The LoadModel package relates energy and load areas, their sub-areas and group of loads with schedules.

The inspiration source for the INERTIA CIM is how schedules and Load areas/sub-areas are connected.

**ControlArea package**

The ControlArea package models information about power system resources, generation units and measurements.

The information models for generation units and measurements can be inspiriting source for the INERTIA CIM.

**CIM tool**

CIM UML model exists as an Enterprise Architect project file. The experience has shown that exchanges between different UML tools are not perfect [9]. There is specific tool for processing the CIM UML file called CIMTool [10]. CIMTool is an open source tool that supports the Common Information Model (CIM) standards used in the electric power industry. It is potentially useful for other semantic applications too.

CIMTool supports following functions [10]

- read and merge CIM and local UML models in XMI form
- browse models and check inconsistencies
- generate equivalent OWL ontologies
- create and edit profiles
- generate XML schemas, OWL and RDFS ontologies for profiles

- validate instances against profiles (including very large CIM/XML instances)

### *4.1.2* IEC 61968

IEC 61968 is CIM extension for Distribution management domain (also called DCIM). The Distribution management domain covers all aspects of management of utility electrical distribution networks. A distribution utility will have some or all of the responsibility for monitoring and control of equipment for power delivery, management processes to ensure system reliability, voltage management, demand-side management, outage management, work management, automated mapping and facilities management [11].

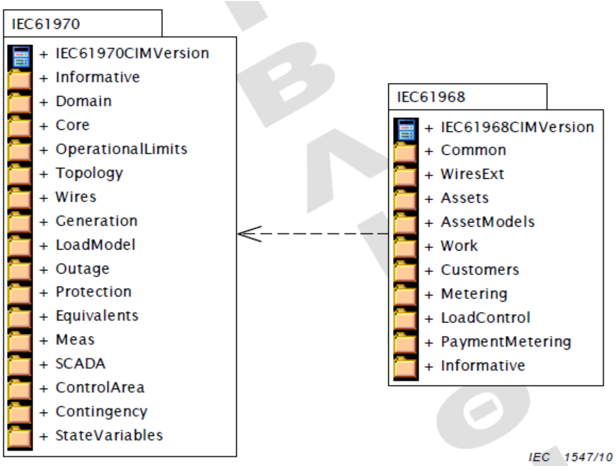



***Figure 3: Packages defined in IEC 61970 and IEC 61968 [12]***

There are 14 parts of IEC 61968 (many are still under development [13]). Some parts are further divided into specific sub-parts. Part of the standard IEC 61968-11 specifies the CIM Extensions for Distribution.

#### 4.1.2.1 IEC 61968-11

IEC 61968-11 is published IEC standard that covers CIM for Distribution. It is also known as DCIM. The main extension added by IEC 61968-11 are related to Asset Management, Customer Management, Meter Data and Work force Management

DCIM exists as Enterprise Architect file and can be also processed by CIMTool (for more info about CIMTool see [10]). The following part describes packages and classes DCIM defined in IEC 61968-11 that are of potential interest (can serve as inspiration source) for INERTIA CIM. For more information about IEC 61968-11 please see see INERTIA internal SoTA document.

**DCIM load model**

The DCIM load model is defined using LoadResponseCharacteristic class, EnergyConsumer class (subclass of CIM Equipment class). EnergyConsumer can be associated with meter through ServiceDeliveryPoint. This model covers single-phase load. For three-phase model the single-phase model is connected (through CIM Terminal class) to the ConnectivityNode.

**DCIM location**

DCIM provides class Location. The DCIM Location class can be used in relation with resource and asset (from IEC CIM).

**Payment metering package**

Payment metering package provided by DCIM defines classes that enable payment metering system to exchange data about financial transactions between customer and service provider. The core class specifying most of such information is Transaction. It is further specified by e.g. CustomerAccount class and PricingStructure class. The package also defines classes for DCIM auxiliary agreement model.

**Common package**

DCIM Common Package contains the classes that support distribution management in general. The IdentifiedObject class is the most abstract one and it has subclasses such as Document, Location, Organisation, TimePoint, ActivityRecord. It contains also class UserAttributes.

**Customers package**

DCIM package Customers contains the core information classes that support customer billing applications. The classes Customer (subclass of Organisation class) and CustomerAgreement (subclass Document) are related into classes about Tariff, ServiceCategory and ServiceLocation. The enumerator for CustomerKind is defined (defining e.g. windMaschine, commertialIndustrial etc.)

**Metering package**

DCIM package Metering introduces classes that are needed for enterprise integration and of metering systems and other enterprise systems. The main class introduced here is MeterAsset. It is related to classes MeasurementValue, Location, EndDeviceEvent and others DCIM and CIM classes.

**LoadControl package**

DCIM package LoadControl is extention of Metering package and contains the information classes that support specialised applications such as demand-side management using load control equipment. These classes are generally related with point where service is delivered to the customer.

**PaymentMetering package**

DCIM package PaymentMetering is extention of Metering package and contains the information classes that support specialised application such as payment metering. These classes are generally associated with the collection and control of revenue from the customer for a delivered service.

### *4.1.3* Conclusion

**Opportunities**

As concluded in SoTA, the overall IEC CIM UML model is complex. It covers the whole domain of power systems in details. For the INERTIA CIM, the reusability of the whole package of norms defining CIM for power systems is low. The promising approach for INERTIA might be use CIM as a starting point, using namespaces to maintain traceability and concentrate mostly on parts IEC 61970-301 and IEC 61968-11.

We might focus on the model of Meter and Units used in the electrical power industry, Production/Generation model (including information object models for specific generators, schedules as well as emission curves etc.), Control area/region model, Energy consumer, Load model, Documents and Contract model and possibly on Equipment model.

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the conclusion is: however we were aware of IEC CIM its classes does not influence the INERTIA CIM specification.

**Gaps and threats**

CIM is not well suited for the INERTIA project. The aim of CIM is to standardize the information model at the level of the Transmission Operator or Distribution Operator enterprise applications but it is not intended for the real time monitoring and control of devices which is the scope of INERTIA.

The INERTIA CIM will cover some aspects that are out of the scope of the IEC CIM. The IEC CIM does not specify capacities and flexibilities of DER (in INERTIA DER model loads are considered as well) and it does not cover the term prosumer. The occupancy prediction and flow model as well as user profile associating user preferences with DERs is also out of the scope of the IEC CIM.

There is threat when trying to fully integrate INERTIA CIM with CIM for electric power industry. The threat comes from the fact that CIM for electric power industry is very complex, since it contains a lot of classes their attributes and relationship between classes. Thus, the integration of information models can be time consuming task and it can significantly slow down project research activities or focus these activities on researching aspects of CIM for electric power industry. UML models are not suitable representation for INERTIA CIM, since partners agreed to use XML data for exchanging data between components. XSD schemas proofed to be the appropriate basis for definition of data types for XML. The XSD schemas can be straightforwardly used by all component developers to be able to correctly manipulate with data defined in INERTIA CIM (see also Step 6 Proposal of Common Information Model of adopted methodology in Section 2).

## *4.2 IEC 61850-7-420*

IEC 61850-7-420 is based on the conceptual class model introduced in IEC 61850 series of norms. The following describes IEC 61850 Conceptual class model:

**SERVER** - represents the external visible behaviour of a device. All other ACSI models are part of the server.

Any server has two roles: to communicate with a client (most service models in IEC 61850 provide communication with client devices) and to send information to peer devices (for example, for sampled values)

**LOGICAL-DEVICE** (LD)/Intelligent Electric Device (IED) - contains the information produced and consumed by a group of domain-specific application functions; functions are defined as LOGICAL-NODEs.

**LOGICAL-NODE** (LN) - contains the information produced and consumed by a domain-specific application function, for example, overvoltage protection or circuit-breaker.

**DATA** - provide means to specify typed information, for example, position of a switch with quality information and timestamp, contained in LOGICAL-NODEs.

The following logical nodes comprise the base for IEC 61850-7-420. The meaning of the Logical nodes is described in the fig. 5.

- DER unit Controllers
  - comprise the LD/IED for DER control
  - e.g. DRCT, DRCS, DRCC and FSEQ (see fig. for )
- Internal Parameters
  - internal status of the DG
  - e.g. DCIP; DFCL, DSTK, DFPM; DPVM, DPVA, DPVC, DTRC; DCHC, DCTS, DCHB
- Grid Connection Units
  - structure of (micro)grid
  - CSWI and XCBR are required for every DG connection
  - XFUS model a fuse and it can be likened to a circuit breaker for one-time use
  - converters: ZRCT and ZINV
  - XSWI or CSWI feed the output to the converter

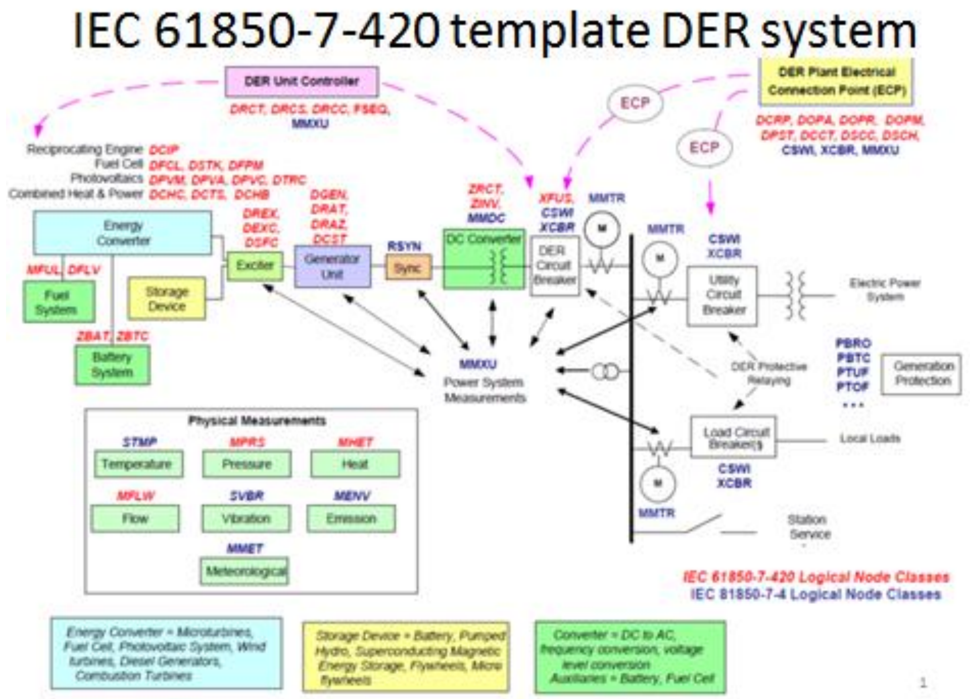



***Figure 4: IEC 61850-7-420 template of DER system providing template configuration of operable system with DGs and storage system based on IEC 61850 (source [14])***

The part 7-420 also provides DER system configuration for specific DGs (e.g. for PV generator, Wind turbines).

**IEC CIM versus Substation Configuration Language**

IEC CIM and Substation Configuration Language (SCL – norm IEC61850) are developed in parallel under different working groups of IEC TC 57. Both have the ability to exchange model and configuration information between different equipment or tools and use XML for storage, there are lot of differences between both the standards.

Differences between IEC CIM and SCL [15] and INERTIA CIM:

- IEC CIM is based on UML (inheritance). SCL representation is sequential or hierarchical INERTIA CIM is based on signatures of interfaces, XSD schemas, DB schemas and Semantic models.

- IEC CIM approach gives emphasis on the equipment inheritance and its interconnection SCL starts from a functional point of view INERTIA CIM gives emphasis on flexibility model, occupancy and flow models containing structure of Local Hub, devices with energy profiles and occupant profile comprisinsing comfort and operational characteristics (description of relaiton with DERs/devices)

- IEC CIM is broadly applied to enterprise integration (EMS, DMS, Planning, Energy Markets and Metering)
  SCL is limited to exchange of data within substation equipment and tools
  INERTIA CIM is defines interaction with data storage service of the IAM for the INERTIA components

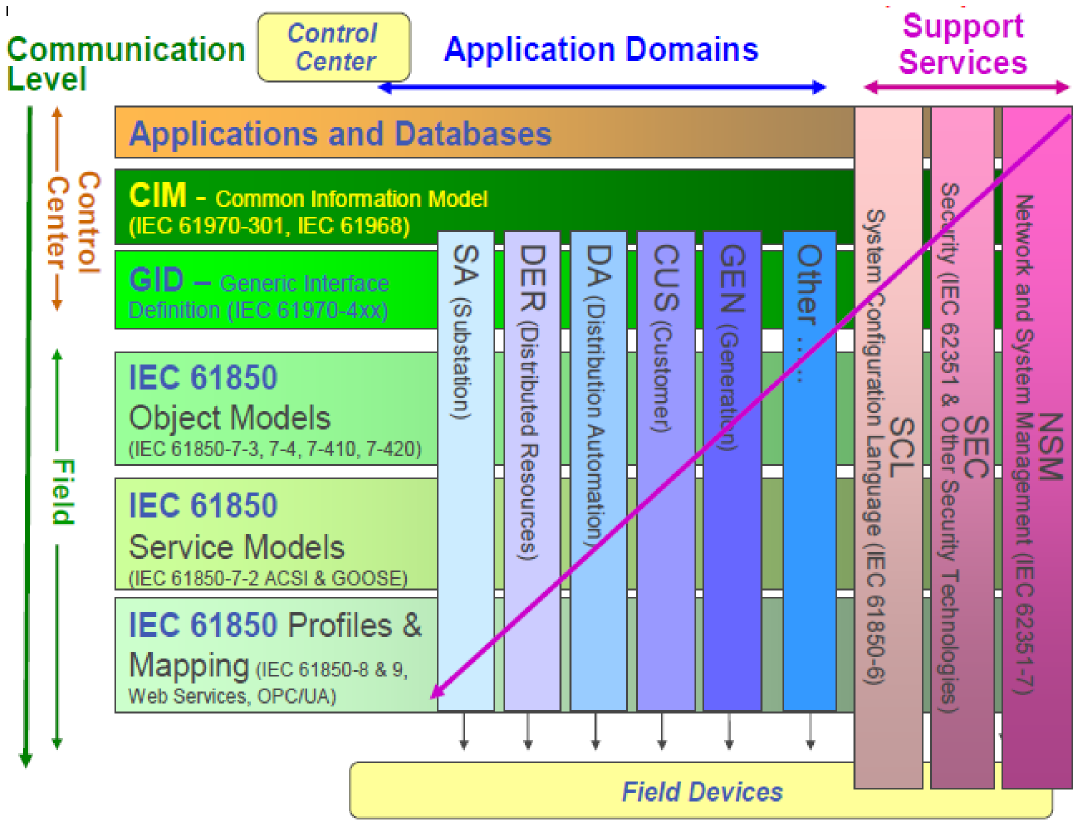



***Figure 5: IEC 61850 Models and the Common Information (CIM) Model (source [14])***

### *4.2.1* Conclusion

IEC 61850 defines Substation Configuration Language that enables to operate (monitor and control) sub-station. The sub-station corresponds to server in the norm. The server enables to operate devices in the sub-station. The devices are modelled from function point of view and this construct of device is called Logical Devices (called also Intelligent Electrical Devices) from predefined elementary devices called Logical Nodes. The norm defines data classes for a lot of Logical Nodes. Part 7-420 defines data classes for DERs (e.g. DGs, storage systems). The part 7-420 also provides Template DER system which depicts typical configuration of LNs for operable DER system based on LNs defined in this part of the norm and LNs from part 7-4 of the norm. The part 7-420 also provides DER system configuration for specific DGs (e.g. for PV generator, Wind turbines). These system configurations can be operated (monitored and controlled) by the GRID operator. They enable to hide complexity of DGs and storage systems under one element (Logical Device). Thus, they easy the operation of complex industry DGs and storage systems that are designed to feed the GRID with the electric power and thus implement interfaces based on IEC 61850. The part 7-420 does not consider Loads.

Since the INERTIA system will enable to monitor operation of DER that is based on the occupancy prediction and flow in the building. The information on prediction of (near real-time) electric power consumption is derived by processing (interpretation and aggregation) of DER data and occupancy prediction and flow data. The INERTIA system will present this information (prediction) to operators (facility and grid). The operators will be able to use this information for the appropriate action based on the situational context (e.g. status of GRID, price of electricity in the GRID, status of DGs and local storage system etc.). The INERTIA system will enable operator to do controlling actions. The local DGs and Storage Systems used in the INERTIA pilot applications are modelled under DERs in INERTIA will not necessary implement interfaces defined in IEC 61850 family of norms. Thus the definition of the LDs, LNs and their data attributes can not be reused in the INERTIA information models (there are not devices supporting it). However, the attributes describing information objects for DGs and storage systems (LNs specification) in the part 7-420 can be reusable (i.e. can serve as an inspiration source for INERTIA DER modelling task T2.2).

Note, in time when INERTIA task T2.1 is finishing and a current version of INERTIA CIM is ready the conclusion is: however we were aware of IEC 61850-7-420 and the profiles of some overlapping LNs it did not influence the INERTIA CIM specification.

# 5. Analysis of requirements to INERTIA CIM

The analysis of requirement to INERTIA CIM is provided in this section. As described in the methodology adopted (step 2, 3, 4 and 5 of phase 2 – see section 2), the requirements were collected in form of INERTIA component's interface specifications. These specifications consist of interface name that incorporate whether it is importing or exporting one and specification of INERTIA specific data types used in these interfaces. It is based on the template form that can be found in the Methodology. Such specification were then analysed from the perspective of CIM. Thus it was aimed at definition of IAM interfaces reflecting component interfaces needs and INERTIA specific Data types.

There are some exceptions to this methodology:

- The requirements of component for KPI calculation and DER calculation on IAM and CIM will be fully analysed in the forthcoming INERTIA reports D2.3.2 and D2.4.2 respectively. However, the initial analysis of requirements based on current reports D2.3.1 and D2.4.1 is presented here (but does not follow the methodology described above).

- The requirements of User Profiling component, INERTIA Ambient User Interface and INERTIA Aggregator Analytics were provided with pleny insight information into the model used so this information is also presented here.

- The requirements of Occupancy prediction and flow component have influence on the BIM model. The requirements on BIM model were defined by partner responsible for Occupancy prediction and flow component. The requirements were analysed aiming at proposal of hierarchy of BIM concepts and attributes needed. The INERTIA specific data types are defined in the INERTIA report D2.2, therefore, to avoid duplicity, there is link to concrete data type specification in form of table number (used in D2.2) presented here.

- The LinkSmart event type specification was provided in a textual form. This form is used here.

In most cases the mirror interfaces of IAM to the component interfaces are expected and these IAM interfaces are first part of INERTIA CIM. The component specific data types will be specified as XSD schemas. These XSD schemas are second part of INERTIA CIM. Since IAM will provide repository services the database schemas are third part of the CIM. The forth part of CIM are semantic models that enable to put together data from heterogeneous sources, enrich dynamic data with static context etc. Semantic models will be fully defined in INERTIA report D3.2, nevertheless the analyses and proposal of CIM in this report is reflecting some of their aspects as well as needs of date stored in the DB tables to be annotated to these semantic models.

Since the architecture of the INERTIA framework is divided into the Local Control Hub and Aggregator Control Hub and components of these two frameworks serve different purposes the IAM is defined for both Local Hub and Aggregator frameworks. As there are Local Control Hub IAM and Aggregator Control Hub IAM the definition of CIM is analysed and specified for these two components separately. Thus this section is split into Local Control Hub CIM and Aggregator Control Hub CIM.

## *5.1 Local Control Hub CIM*

The Local Control Hub CIM (referred as LH CIM or by CIM in the following) serves as a commonly shared vocabulary by to Local Control Hub INERTIA framework components. This commonly shared vocabulary of LH CIM contains Local Hub IAM interfaces serving for Local Control Hub components and INERTIA specific data types used in these interfaces in form of XSD schemas. At last LH CIM defines DB schemas of tables and relevant annotations to semantic models for storing the data coming from these components and retrieving and serializing data that are requested from LH IAM by these components.

To be able to identify the LH CIM elements, the requirements to LH IAM (serving as a repository service) form the LH components were collected in the predefined template form (see Annex 1). These requirements specify the LH component interfaces and the INERTIA specific data types. The collected requirements in this format were then analysed from the perspectives of LH CIM elements (interfaces, XSD schemas, DB table schemas, Semantic models).

The requirements from LH components on LH IAM and their analysis from the perspective of LH CIM elements is in the following.

### *5.1.1* LinkSmart requirements

Based on D1.3 LinkSmart will be supported by device discovery mechanisms responsible for retrieving the related service interfaces for physical devices including discovery of semantic devices by providing the relevant semantic device aggregation according to the real presence of physical devices. IAM will support this process by providing required context information (see INERTIA deliverable D3.1). Based on D3.1 requirements on IAM here is list of respective IAM interfaces and specification of INERTIA data types used in these interfaces that will support LinkSmart with the context information.

#### 5.1.1.1 Data models for LinkSmart

Spatial structure data model put together location model (represented as reference to location model) with the data about DERs.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| spatialStructureType | locationModelRef | String | Location Model References – ID of Area |
| | spaceWithDER | spaceWithDERsType | Space with respective DERs and other characteristics |

DER data type (data view) specification for LinkSmart

| Type name | Attributes | Type | Description |
|---|---|---|---|
| spaceWithDERsType | spaceRef | String | Space identifier |

| | | | |
|---|---|---|---|
| | derIDs | Array of String | List of DERs |
| spaceWithDERsType | zoneRef | String | Zone identifier |

DER model is specified by the reference to the instance of DER semantic model, reference to Location semantic model, states of DER, energy profile and relevant IOT devices

| Type name | Attributes | Type | Description |
|---|---|---|---|
| derModelType | derModelRef | String | Der model identifier |
| | locationRef | String | Location identifier(e.g. space) |
| | derStates | Array of String | List of DER states in a commonly shared (between IAM and Linksmart) format determined by the semantic model (see D3.1) |
| | energyProfile | energyProfileType | Energy profile of the device |
| spaceWithDERsType | iotDeviceIDs | Array of String | Listo of IDs of IOT devices belonging to the semantic DER |

Each device is specified by energy profile that put together states of the device and respective energy consumption to this state.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| energyProfileType | derState | String | DER state |
| | energyNominalPower | powerType | Watt (J/s) power in the state specified |

The second role of the LinkSmart middleware is to publish the events form the semantic devices. LinkSmart contains event manager for this purpose. All INERTIA components can access real time events by listening LinkSmart event manager. The role of IAM regarding the INERTIA events is to store the events published by LinkSmart. The INERTIA framework components need these historical events enhanced with context information (such as BIM static data, DER static data). The storing of events in IAM will save the dynamic data in the DB tables (e.g. values of sensor) and the entry will contain semantic annotation to the semantic model (e.g. space in the BIM semantic model). Using this annotation, event data can be enriched by any information form semantic model.

**LinkSmart event description – requirement on linkSmartEventType**

The event specification is designed for LinkSmart IoT devices and applications with the sending and receiving of measurement data and state changes in IoT devices as the main purpose. Not having a well defined event format has proved to make it difficult to integrate LinkSmart applications and leverage event-driven behavior in Linksmart IoT applications. Querying measurement data without metadata or a common format is also quite difficult.

**Metadata**

"EventType" is used to the describe if the event is e.g. a state change in a property of an IotDevice, an alarm, an action being initiated or a system message (e.g. event manager shutting down).

"EventId" is used to keep track of the event in the publish-subscribe system. If not supplied, one will be assigned by the Event Manager.

"Topic" is not mandatory, the subscription event filtering in Event Managermay be on topic or content.

"Timestamp" is mandatory. For possible future EM functionality using historical events, this timestamp will be used. It is the time when the event was generated.

"Comment" is a human-readable description of the event and is not meant for automated processing, but for information like "This is a test!"

The "Source" tag is mandatory, as is the "ID" attribute. This attribute should uniquely identify the event source. This could be the LinkSmart description if this is globally unique, or the PID, or an EPCglobal id. (In the case of LinkSmart sensor/actuator proxies, the ID should be sufficient identification to allow the event recipient to make actuator calls to the proxy.) The "Source" tag is open, so any application or IoTDevice could use tags with identification that is useful in the context of that application. There are some suggested tags, however (none of which are mandatory):

- "Project" - to distinguish events meant to be used in a specific context ("SEEMPubS", "Cnet Office Demo", "Polito Test") sent through the same Event Manager.
- "Location" is used to specify physical location.

- “ObjectID” refers to an identity specified in an ontology or similar model (which may differ from the Source/@ID attribute).

- “ProcessID” identifies the process context of the event, a part in or step of a process model (e.g. business, industrial, manufacturing)

**Content**

The “Content” contains an “Observation” that describes the changed state of an “ObservableProperty” of the “Source”.

The Observation has a “Value” (e.g. “37.5”, “1572”, “True”, “On”, “Patient X admitted” or a complex structure in a format like XML or JSON).

The Observation has a “Timestamp”, which is the time that the Value is valid for. It also has an “ObservationTimestamp” for the time this measurement/observation/forecast/revision/estimate was made. For “normal” sensors, the “ObservationTimestamp” is omitted and is assumed to be equal to “Timestamp”. That should cover energy consumption sensors, weather forecasts (and updates thereof), expected energy consumption forecasts as well as future revisions of the estimated average temperature year 30000 BC.

ObservableProperty specifies which property of the event source the observation belongs to, e.g “InternalTemperature”, “CurrentPowerConsumption”, “IsOpen”, ”PowerState”. This should be unique for a specific event source. If left out, the source can be assumed to send events for only one property.

ObservablePropertyType is a classification of the ObservableProperty, e.g. SI base quantities (physical quantity) such as "Temperature", "Mass", "Luminous intensity". The purpose is to allow the recipient to understand more about the event content without previous knowledge about the event source. It could be just a string, e.g. "Temperature", “PowerConsumption” or a reference to a specific model or ontology. It could also be left out altogether if the observable property type (and perhaps also its unit and data type) is already specified in the ontology.

“Properties” is a way to allow non-application specific components to store data about the observation.

### 5.1.1.2 Interfaces for LinkSmart

**The IAM interface for storing events from LinkSmart**

| IAM Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| setLinkSmartEvent | linkSmartEvent | linkSmartEventType | Data structured based on LinkSmart Event descritpion |

**The IAM interfaces providing information to LinkSmart**

| IAM Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getLocationModel | locationRef | String | Location Reference in form of ontology URI of concrete location |
| | recursive | Boolean | The switch to decide, if the service should return the full model of embedded locations or only the plain description of the location specified by locationRef without embedded structure |
| return | spatialStructure | spatialStructureType | The hierarchic spatial structure including related equipment and the corresponding zone obtained from spatial semantic model |

| IAM Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getDeviceLocationModel | deviceRef | String | The ontology URI of concrete device, for both DER and IOT device. |
| | recursive | Boolean | The switch to decide, if the service should return the full model of embedded locations or only the plain description of the location specified by deviceRef without embedded |

| | | | |
|---|---|---|---|
| | | | structure. |
| return | spatialStructure | spatialStructureType | The XML structure in predefined format describing the hierarchic spatial structure including related equipment. |

| IAM Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getZoneDERs | zoneRef | String | zoneRef The ontology URI of concrete zone. |
| return | dersList | List of String | The list of DERs (their IDs) |

| IAM Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getDERModel | derRef | String | derRef The ontology URI of concrete DER device. |
| return | derModel | derModelType | The XML structure in predefined format containing the full description of particular DER device including all its properties. |

#### 5.1.1.3 Impact of LinkSmart requirements on CIM

The specification of IAM exporting interfaces and respective data types used in these interfaces (described above) is part of CIM vocabulary. All the supporting context data for LinkSmart are obtained querying semantic models (specified initially in D3.1 and fully in D3.2). The data type for events used in the IAM interface serving for manipulating with LinkSmart events together with the DB schema specification designed for storing the Linksmart Events can be found in chapter 5. This DB schema contains also additional fields to be able to associate data objects of other INERTIA components (such as User profiling, Occupancy prediction and flow component) with the Event record. Some of the parameters describing LinkSmart Events will enable to bind data with semantic models (Location, ObjectID).

For the event format, it is assumed that the CIM Ontologies will include models for some key elements. The EventType is one of them, but there is aslo Source/Location and Observation/ObservablePropertyType. Based on the values of these properties (ObservablePropertyType can contain "TEMPERATURE", "OCCUPANCY" and ObservableProperty e.g. "IndoorTemp", "DetectedID", "NoOfPersons") the events can be distinguished into environmental (thermal, visual), and operational and occupancy related, thus filtered based on the calling parameters for requesting INERTIA components.

### *5.1.2* Occupancy prediction and flow component requirements

The INERTIA data types are defined in the INERTIA report D2.2. To avoid duplicity, the link is to the table number (used in D2.2) is presented in case of INERTIA data types is used in the interface.

#### 5.1.2.1 Data models of Occupancy prediction and flow

The Occupancy data models used in the method specified below are defined in the INERTIA report D2.2. The reference to the specific table in the D2.2 is provided in the definition of methods.

However, the conceptualization of Local hub and Building model was provided separately and is described in the following.

**Local Hub and BIM model**

As defined by Occupancy prediction and flow component developers, to be able to monitor human presence and movement within the building parts, the following granularity of objects is needed:

- Facility: Referring to single Local Hub, to be handled as a unique entity by the Aggregator e.g. CERTH premises.
- Building/Area: a Facility/L.C.H. could have different buildings. Could also represent the term "area" mentioned above (e.g. parking area).
- Sub-Area: A group of spaces within a building directly associated with each other (eg. the same floor within a building).
- Space: A room or explicitly defined "space" (e.g. corridor, stairs, entrance hall etc.)
- Zone: Some spaces could be divided in different zones, in terms of HVAC and lightning (eg. a room could have 2 separate zones of air-conditioning or 4 different zones of lights)

Using this Local Hub and BIM model the building infrastructures description have to be accessible. Thus, a spatial reference to position DERs, sensors (with their orientation in space), actuators, occupatnts etc. need to be modelled. Furthermore, there is a need to add energy-related characteristics such as consumption of HVAC and Lighting (but also wall/window/roof/floor materials as they are in gbXML standard [16]) that could help DER modeling in defining building's thermal characteristics. Devices should be allocated to the lowest level of above hierarchical schema as possible e.g. a light or PC should be assigned to a zone (or a space, if no zones are available).

#### 5.1.2.2 Interfaces for Occupancy prediction and flow

The importing and exporting interfaces of Occupancy prediction and flow (OCC), which are intended to import data from IAM or export data to IAM (respectively), are presented in this section. The interfaces are described by their name, by the importing/exporting information, by the format of the information and the component what is the destination (in case of OCC exports) or source (in case of OCC import) of these information. Then the signature of the interface is presented described by the interface name and parameters specified by XSD. XSD specification of the parameters enables to clearly identify data types needed. If the data type is specific for INERTIA the reference to the D2.2, where the type is defined, is provided. In most of the following interface description there are also parts providing comments and examples that clarify the interface description. Under the description of all OCC I/O interfaces there is conclusion providing analysis of the requirements of these interfaces on the INERTIA CIM. Note the interfaces are associated with numbers that are referred in the conclusion for convenient reading (importing interfaces numbers 1-10, exporting interfaces numbers 1-6).

**OCC importing interfaces descriptions**

**1 Import of Historical occupancy data**

The signature of Occupancy prediction and flow (OCC) importing interface for Historical occupancy data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getLocationModel | dateFrom | dateTime | Date and time from |
| | dateTo | dateTime | Date and time to |
| | timeStep | Float | Duration of the time step |
| | spatialID | string | could refer to any level of the spatial granularity proposed, from a building/area to the lowest level of a space or zone |
| return | numberOfOccupants | List of int | Exact number of occupants detected per date, time and spatialID. The return value will be given |

| | | | |
|---|---|---|---|
| | | | according to the lowest spatial granularity available for the specified place (e.g. per zone, space, sub-area). |

## 2 Import of Historical occupants location data

The signature of OCC importing interface for Historical occupant's location data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getHistoricalOccupantLocation | dateFrom | dateTime | Date and time from |
| | dateTo | dateTime | Date and time to |
| | timeStep | Float | Duration of the time step |
| | occupantID | string | Identifier of occupant |
| return | spatialID | string | SpatialID of occupant's location per date and time. It will be given according to the lowest spatial granularity available for the specified place. See the example below for the structure of requested spatialID |

## 3 Import of Real-Time occupancy

The signature of OCC importing interface for Real-Time occupancy data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getRealTimeOccupancy | spatialID | spaceID | ID of space |
| return | ccupantNumber | int | Latest actual number of Occupants |

| | | | |
|---|---|---|---|
| | | | detected. |

***Comments:***

- Returned data could be the latest value of a table storing historical values or could be stored at a separate table, maintaining only the latest (near Real-Time) values – the second approach could be more efficient than seeking in a large table.
- The return value will be given according to the lowest spatial granularity available for the specified place (e.g. per zone, space, sub-area)
- The aggregated number of occupants per space will not be returned but instead calculated (e.g. Space1 = space_1_zone_1 + space_1_zone_2 = 6+2 = 8).

## 4 Real-Time occupants location

The signature of OCC importing interface for Real-Time occupant's location data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getRealTimeOccupantLocation | occupantID | string | Identifier of occupant |
| return | SpatialID | string | Latest SpatialID of occupant's location.<br><br>Comment: Returned data could be the latest value as stored in the historical values table or could be stored at a separate table, maintaining only the latest (near Real-Time) values. |

## 5 Spatial information of the actual location of the sensors

The signature of OCC importing interface for spatial information of the actual location of the sensors:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| getSensorLocation | spatialID | string | Identifier of space |
| return | sensorLocation | Building Information Model/ gbXML | SensorID, sensorType, sensor location coordinates per each sensor located within the place specified by spatialID. |

## 6 Spatial typical or dynamic scheduling

The signature of OCC importing interface for spatial typical or dynamic scheduling:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getSpatialSchedule | spatialID | string | Spatial identifier |
| | dateTimeFrom | dateTime | Date and time from |
| | dateTimeTo | dateTime | Date and time to |
| return | activeEvents | List of event | List of active events scheduled for the place specified by the spatialID. The returned data per each event will be stateType, startTime, endTime and numOfParticipants, as defined in table 13 of D2.2, as well as type of the event (typical or dynamic) along with the related parameters defined in tables 15 and 16 of D2.2. |

***Comments:***

- stateType and periodicity will defined by enumerations
- stateType values: presence or absence. It defines presence or absence from the space specified by the spatialID.

## 7 Import of Occupant typical or dynamic scheduling information

The signature of OCC importing interface for Occupant typical or dynamic scheduling:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getSpatialSchedule | occupantID | string | Occupant identifier |
| | dateTimeFrom | dateTime | Date and time from |
| | dateTimeTo | dateTime | Date and time to |
| return | activeTasks | List of task | List of active tasks scheduled for the specified occupant or group of occupants. The returned data per each task will be spatialID, stateType, startTime and endTime, as defined in table 14 of D2.2, as well as type of the task (typical or dynamic) along with the related parameters defined in tables 15 and 16 of D2.2. |

***Comments:***

- stateType and periodicity will defined by enumerations
- stateType values: presence or absence. It defines presence or absence from the space specified by the spatialID.

# 8 Import of Open Reference Models

The signature of OCC importing interface for open reference model data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOpen ReferenceModels | typeOfBuilding | BuildingType | Type of building |
| | typeOfSpace | SpaceType | Type of space |
| return | openReferenceModels | List of openRefernceModelTypes | INERTIA open reference models are specified in 5.3 of D2.2. |

# 9 Import of INERTIA Occupancy and Flow model

The signature of OCC importing interface for INERTIA Occupancy and Flow model data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOverallInertia OccupancyAndFlowModel | LCH_ID | string | Local hub identifier |
| | dateTimeFrom | dateTime | Date and time from |
| | dateTimeTo | dateTime | Date and time to |
| return | occAndFlowModels | List of OverallOccupancy AndFlowModel | INERTIA Occupancy and Flow Models (Overall) created within the specified period. Model parameters are specified in section 5.2.1 of |

| | | | |
|---|---|---|---|
| | | | D2.2 |

***Comments:***

- LCH stands for Local Control Hub
- INERTIA model will be initialized using open reference models and typical schedules. After initialization, it be will be updated on a regular basis based on sensor observations. Various versions of the model will be stored for future reference and evaluation. Each version of the model will have a datetime stamp of its creation, so that the latest or a previous model(s) could be asked.

## 10 Import of INERTIA Occupancy and Flow Model (Individual)

The signature of OCC importing interface for INERTIA Occupancy and Flow model data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getIndividualINERTIAOccupancy AndFlowModel | LCH_ID | string | Local hub identifier |
| | dateTimeFrom | dateTime | Date and time from |
| | dateTimeTo | dateTime | Date and time to |
| return | occAndFlowModels | List of IndividualOccupancyAndFlowModel | INERTIA Occupancy and Flow Models (Individual) created within the specified period. Model parameters as specified in section 5.2.2 of D2.2 |

***Comments:***

- LCH stands for Local Control Hub
- INERTIA model will be initialized using open reference models and typical schedules. After initialization, it be will be updated on a regular basis based on sensor observations. Various versions of the model will be stored for future reference and evaluation. Each version of the model will have a datetime stamp of its creation, so that the latest or a previous model(s) could be asked.

### OCC Export interfaces description

#### 1 Export of Real-Time occupancy per space

This OCC method will export Real-Time occupancy per space, in terms of actual number of occupants per space

The signature of OCC interface for exporting Real-Time occupancy per space data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| exportRealTimeOccupancyPerSpace | dateTimeStamp | dateTime | Date and time stamp |
| | spatialID | String | Identifier of space |
| | numberOfOccupants | int | Number of occupants in a given time |
| return | operationSuccess | boolean | Information whether the operation was successful or not |

#### 2 Export of Real-Time individual occupant’s location

This OCC method will export Real-Time individual occupants location, provided by RFID or other occupant identification mechanisms (for example, provide exact positioning of a person owning a specific RFID card)

The signature of OCC interface for exporting Real-Time individual occupant’s location data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| exportRealTimeOccupantLocation | dateTimeStamp | dateTime | Date and time stamp |
| | occupantID | String | Identifier of |

| | | | |
|---|---|---|---|
| | | | occcupant |
| | spatialID | String | Identifier of space |
| return | operationSuccess | boolean | Information whether the operation was successful or not |

## 3 Export of INERTIA Occupancy and Flow Model (Overall)

This OCC method will export INERTIA Occupancy and Flow Model (Overall) for specified Local Control Hub. This model will be constantly updated (e.g. on a daily basis or even more often) to reflect changes that may occur on building's occupancy behaviour (e.g. people changing working places, new people or spaces added etc.) in order to provide input for the day-ahead supply demand operations or for User Profiling purposes.

The signature of OCC interface for exporting Overall INERTIA Occupancy and Flow Model data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| exportOverallInertia Occupancy<br><br>AndFlowModel | dateTimeStamp | dateTime | Date and time stamp |
| | LHC_ID | String | Identifier of Local Control Hub |
| | overallOccAndFlow | OverallOccupancy AndFlowModel | INERTIA Occupancy and Flow Model (Overall) as specified in section 5.2.1 of D2.2 |
| return | operationSuccess | boolean | Information whether the operation was successful or not |

# 4 Export of INERTIA Occupancy and Flow Model (Individual)

This OCC method will export INERTIA Occupancy and Flow Model (Individual) for specified Occupant. This model will be constantly updated (e.g. on a daily basis or even more often) to reflect changes that may occur in order to provide input for the day-ahead supply demand operations or for User Profiling purposes.

The signature of OCC interface for exporting Individual INERTIA Occupancy and Flow Model data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| exportIndividualI InertiaOccupancy<br><br>AndFlowModel | dateTimeStamp | dateTime | Date and time stamp |
| | occupantID | String | Identifier of occupant |
| | overallOccAndFlow | individualOccupancy AndFlowModel | INERTIA Occupancy and Flow Model (Individual) as specified in section 5.2.2 of D2.2: |
| return | operationSuccess | boolean | Information whether the operation was successful or not |

# 5 Export of Spatial Occupancy and Flow prediction export

This OCC method will export Spatial Occupancy and Flow prediction (e.g. per building/space/zone).

The signature of OCC interface for exporting Spatial Occupancy and Flow prediction data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| exportSpatialOccu pancy<br><br>AndFlowPrediction | spatialID | String | Spatial identifier |

| | | | |
|---|---|---|---|
| | predictionDate | Date | Prediction date |
| | timeFrom | Time | Time from |
| | timeTo | Time | Time to |
| | occupancyGranularity | OccupancyValue Type | Occupancy granularity<br><br>exact number of occupants, portion of maximum number of occupants (per sub-area, space, zone) or occupancy ranges (empty, low-occupied etc) |
| | predictionTimeStep | Int | Prediction time step |
| | predictedNumberOfOc cupant | | Predicted number of occupants, portion or occupancy range (according to the value of occupancyGranul arity) per timestep and spatialID. The return value will be given according to the lowest spatial granularity available for the specified place (e.g. per zone, space, sub-area). Defined in table 19 of D2.2 (example in table 20 of D2.2). |
| | requestDateTimestam p | dateTime | Request datetime stamp refers to the request sent to our component asking for the |

| | | | |
|---|---|---|---|
| | | | prediction. |
| | exportDateTimestamp | dateTime | Export datetime stamp refers to our component's reply to the prediction request. |
| | requestSource | string | Request source is the entity that requested the prediction (e.g. Local Agent, Holistic Flexibility component). |
| Return | operationSuccess | boolean | Information whether the operation was successful or not |

## 6 Export of Occupant Location and Flow prediction

This OCC method will export Occupant Location and Flow prediction.

The signature of OCC interface for exporting Occupant Location and Flow prediction data:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| exportOccupantLocationAndFlowPrediction | occupantID | String | Identifier of Occupant |
| | predictionDate | Date | Prediction date |
| | timeFrom | Time | Time from |
| | timeTo | Time | Time to |
| | predictionTimeStep | Int | Prediction time step |
| | PredictedLocation | string | Predicted location (spatialID) of occupant per timestep. The return value will be given according to the lowest spatial |

| | | | |
|---|---|---|---|
| | | | granularity available for the specified place. Defined in table 20 of D2.2 (example in table 22 of D2.2). |
| | Request DateTimestamp | dateTime | Request datetime stamp refers to the request sent to our component asking for the prediction. |
| | exportDateTimestamp | dateTime | Export datetime stamp refers to our component's reply to the prediction request. |
| | requestSource | string | Request source is the entity that requested the prediction (e.g. Local Agent, Holistic Flexibility component). |
| Return | operationSuccess | boolean | Information whether the operation was successful or not |

#### 5.1.2.3 Impact of OCC importing and exporting interfaces on CIM

The importing and exporting interfaces of OCC, which are intended to import data from IAM or export data to IAM (respectively), influence the IAM interfaces and IAM hybrid storage system. There are data types specific for INERTIA used in these interfaces. These specific data types have to be parsed and stored by IAM in its hybrid repository. The hybrid repository consists of SQL tables and ontologies. The definition of these INERTIA specific data types (used in the interfaces) together with DB schema definition and ontology structure definition define INERTIA CIM. In general the dynamic data (i.e. data changing over time) are stored in DB tables whereas the static data (i.e. data not changing over time) are stored in the ontologies (as instances of ontology concepts). The data from DB tables are annotated to the ontologies. In case the identification of relevant aggregated items (such as sub-spaces of building, buildings belonging to one control hub) needs to be done, then query to ontology selects it. Using the annotations in DB tables, all relevant data from DB tables (such as historical occupancy) are further selected.

**OCC IMPORTING interfaces**

The following table provide the mapping between OCC importing interfaces (getters) and interfaces of IAM that are intended to serve them. Thus these IAM interfaces are first part of CIM vocabulary definition that needs to be known by the OCC component (but also other component) to be able to use it. The names of the IAM interfaces begin with "OCC_" and continue with whole name of OCC interface name to be able to easier distinguish the bindings in the implementation.

***Table 8: Mapping between Occupancy Prediction and Flow interfaces (retrieval) and IAM interfaces***

| Nr | OCC interface | IAM interface |
|---|---|---|
| 1. | OCC.getHistoricalOccupancy | IAM.OCC_importHistoricalOccupancy |
| 2. | OCC.getHistoricalOccupantLocation | IAM.OCC_importHistoricalOccupantLocation |
| 3. | OCC.getRealTimeOccupancy | IAM.OCC_importRealTimeOccupancy |
| 4. | OCC.getRealTimeOccupantLocation | IAM.OCC_importRealTimeOccupantLocation |
| 5. | OCC.getSensorLocation | IAM.OCC_importSensorLocation |
| 6. | OCC.getSpatialSchedule | IAM.OCC_importSpatialSchedule |
| 7. | OCC.getOccupantSchedule | IAM.OCC_importOccupantSchedule |
| 8. | OCC.getOpenReferenceModels | IAM.OCC_importOpenReferenceModels |
| 9. | OCC.getOverallInertiaOccupancyAndFlowModel | IAM.OCC_importOverallInertiaOccupancyAndFlowModel |
| 10. | OCC.getIndividualINERTIAOccupancyAndFlowModel | IAM.OCC_importOverallInertiaOccupancyAndFlowModel |

Note, in the following the names of IAM interfaces are used without part "IAM.".

OCC_importHistoricalOccupancy and CIM

This method will filter and serialize the historical occupancy data belonging to the space specified. First all relevant spaces of BIM ontology are identified using provided space ID (e.g. if the ID refers to the building, all spaces belonging to this building are considered). Then all entries of DB table that are annotated to this spaces and fits to the other constraints provided by the method input selected and serialized in the form specified by examples. The BIM ontology contains the model of facilities and buildings. BIM ontology specification can be found in D3.2. The specification of DB table is in the section 6.

OCC_importHistoricalOccupantLocation and CIM

This method will filter and serialize the historical occupant location data from DB table using the constraints provided by the method input. The location is provided in at the space level (lowest level in BIM model). The specification of DB table depends on specification of occupant movement and presence events in the LinkSmart events.

OCC_importRealTimeOccupancy and CIM

This method will filter and serialize the near-real time occupancy data belonging to the space specified. First all relevant spaces of BIM ontology are identified using provided space ID (e.g. if the ID refers to the building, all spaces belonging to this building are considered) Based on the example provided, when space is separated into more zones these zones have to be considered in the occupancy calculation. The entries of near real time occupancy will be saved in DB table. These entries will contain annotation to the building spaces. Thus the occupancy of relevant spaces inferred from space ID provided by the method input will be selected and serialized in the form specified by examples. The BIM ontology contains the model of facilities and buildings. BIM ontology specification can be found in D3.2. The specification of DB table is in the section 6.

OCC_importRealTimeOccupantLocation and CIM

This method will filter and serialize the near-real time occupant location data. Using OccupantID provided as input by the interface, the entries of near real time occupant space will be selected from DB table. Querying BIM ontology this space will be enhanced with BIM context information i.e. info about Zones, Sub-areas, building, facility) and serialized into the form specified by examples. The BIM ontology contains the model of facilities and buildings. BIM ontology specification can be found in D3.2. The specification of DB table containing near-real time location of occupants is in section 6

OCC_importSensorLocation CIM

This method will filter and serialize the data about sensor spatial location. Using spatialID, provided as input by the interface, the IAM will obtain all relevant sensors and their types querying BIM ontology and Sensor ontologies. The specific coordinates will be most likely obtained querying sensor specific DB tables containing further (dynamic) data about sensors (since the coordinates can easily change in time). The BIM ontology contains the model of facilities and buildings. The sensor ontology is part of DER ontology. BIM and DER ontology specifications can be found in D3.2. The specification of DB table containing near real time location of occupants is in section 6.

OCC_importSpatialSchedule and CIM

This method will filter and serialize the spatial typical or dynamic schedule for the specified space. Thus, the schedules relevant for space ID and time period provided by the method inputs will be selected and serialized in the form specified by examples. The dynamic scheduling will be stored in DB table while (most likely) the typical schedules will be stored in the BIM ontology. The BIM ontology specification can be found in D3.2. The specification of DB table containing dynamic scheduling is in section 6.

OCC_importOccupantSchedule and CIM

This method will filter and serialize the occupancy schedule data from ontology using the constraints provided by the method input. The ontology OCCUPANCY_TYPICAL_SCHEDULE contains the typical occupancy patterns for specific occupant roles (occulantGroupId). The ontology will be queried; requested structures will be retrieved and serialized in the form specified below. For ontology specification can be found in D3.2.

OCC_importOpenReferenceModels and CIM

This method will filter and serialize the open reference model data as they are ospecified in section 5.3 of D2.2. The open reference model data will be selected from ontology using the constraints provided by the method input (building type and space type). The BIM ontology contains the open reference model data for specific building type and space type (e.g. research building, IT lab). The ontology will be queried; requested structures will be retrieved and serialized in the form specified in section 6. Ontology specification can be found in D3.2.

OCC_importOverallInertiaOccupancyAndFlowModel and CIM

This method will filter and serialize the Occupancy and Flow Model data as they are specified in section 5.2.1 of D2.2. The Occupancy and Flow Model data will be selected by querying the IAM repository using the constraints provided by the method input (ID of local control hub and time period). The typical schedule and open reference model that are part of the model data will be most likely stored in the BIM ontology. Although the historical (dynamic) data (containing time stamps) will be stored in DB table and annotated to ontology. The IAM repository will be queried, then requested structures will be retrieved and serialized in the form specified in section 6. The ontology specification can be found in D3.2.

OCC_importIndividualINERTIAOccupancyAndFlowModel and CIM

This method will filter and serialize the Individual Occupancy and Flow Model data as they are specified in section 5.2.2 of D2.2. The Individual Occupancy and Flow Model data will be selected by querying the IAM repository using the constraints provided by the method input (Occupant and time period). The individual occupancy and flow model will be stored in DB table and annotated to ontology (most likely by user role). The IAM repository will be queried; requested structures will be retrieved and serialized in the form specified in section 6. The ontology specification can be found in D3.2.

**OCC EXPORTING interfaces**

The following table provides the mapping between OCC exporting interfaces and interfaces of IAM that are intended to serve them (thus the importing IAM interfaces serving to OCC). The naming convention is the same as for importing OCC interfaces. There will be one extra interface provided by IAM that is relevant for storing and updating Local Hub and BIM model.

***Table 9: Mapping between Occupancy Prediction and Flow interfaces (storage) and IAM interfaces***

| Nr | OCC interface | IAM interface |
|---|---|---|
| 1. | OCC.exportRealTimeOccupancyPerSpace | IAM.OCC_exportRealTimeOccupancyPerSpace |
| 2. | OCC.exportRealTimeOccupantLocation | IAM.OCC_exportRealTimeOccupantLocation |

| | | |
|---|---|---|
| 3. | OCC.exportOverallInertiaOccupancyAndFlowModel | IAM.OCC_exportOverallInertiaOccupancyAndFlowModel |
| 4. | OCC.exportIndividualInertiaOccupancyAndFlowModel | IAM.OCC_exportIndividualInertiaOccupancyAndFlowModel |
| 5. | OCC.exportSpatialOccupancyAndFlowPrediction | IAM.OCC_exportSpatialOccupancyAndFlowPrediction |
| 6. | OCC.exportOccupantLocationAndFlowPrediction | IAM.OCC_exportOccupantLocationAndFlowPrediction |
| 7 | | Boolean IAM.setLocalHubandBIMmodel(String ID, LocalHubandBIMmodel lh_bim) |

Note, in the following the names of IAM interfaces are used without part "IAM.".

OCC_exportRealTimeOccupancyPerSpace and CIM

This method is used to store the actual number of occupants in certain time in concrete location. The IAM will store the data in the database DB_NAME described below. The database column spatialID contains the semantic annotation to the instance in space and location ontology. The spatial object refers to the context entity of concrete location in the application model.

OCC_exportRealTimeOccupantLocation and CIM

This method is used to store the actual location of occupants. The IAM will store the data in the database described in section 6. The database column spatialID contains the semantic annotation to the instance in space and location ontology. The spatial object refers to the context entity of concrete location in the application model.

OCC_exportOverallInertiaOccupancyAndFlowModel and CIM

This method is used to store the Overall INERTIA Occupancy and Flow Model data as described in D2.2 section 5.2.1. The IAM will store the data in the database DB_NAME described below. The database column LHC_ID contains the semantic annotation to the instance in space and location ontology.

OCC_exportIndividualInertiaOccupancyAndFlowModel and CIM

This method is used to store the Individual INERTIA Occupancy and Flow Model data as described in D2.2 section 5.2.2. The IAM will store the data in the database DB_NAME described below. The database column RoleID contains the semantic annotation to the instance in Occupant ontology.

OCC_exportSpatialOccupancyAndFlowPrediction and CIM

This method is used to store the Spatial Occupancy and Flow prediction data as defined in D2.2 table 19 and 20. The IAM will store the data in the database

DB_NAME described below. The database column RoleID and spatialID contains the semantic annotation to the instance in BIM and Occupant ontology respectively.

OCC_exportOccupantLocationAndFlowPrediction and CIM

This method is used to store the Occupant Location and Flow prediction data as defined in D2.2 table 20 and 22. The IAM will store the data in the database DB_NAME described below. The database column RoleID and spatialID contains the semantic annotation to the instance in BIM and Occupant ontology respectively.

#### 5.1.2.4 Impact of BIM model on INERTIA CIM

Any INERTIA comment will be requesting data related to locations/building parts with above-described granularity from IAM. The location/building parts are considered as static data (no updates, or rare updates are expected). To be able to respond to the requests in terms of the INERTIA CIM, the following list of concepts need to be modelled in the IAM ontology:

- Location, LocalHub, Facility, Building, Area, Sub-Area, Zone, Space

Following semantic relations enable to model the Location/Building parts at all level of required granularity with regards to the INERTIA Architecture. The Architecture consider Local control hub as an element that can be monitored and controlled on the local level. Therefore aggregation of local DERs is needed. Thus for the local control hubs the relations *hasSubSpace* (transitive relation), *hasZone* will be used. The relation *hasSubSpace* is transitive, thus the aggregation of instances is implicitly supported (i.e. if $1^{st}$ floor hasSubSpace meeting office and research building hasSubSpace $1^{st}$ floor then research building hasSubSpace meeting office). The semantic relation used for Global control hub to identify Local control Hubs of required localisation preferences (i.e. set of aggregated DERs in Thessaloniki) will be called *hasSubLocation*. The specification of BIM ontology can be found in chapter 5.

The requests from INERTIA components will specify location elements by URIs (IDs). These URIs will refer to the instances of Location instances of the location/building part of ontology. Based on the semantic relations presented above the inference mechanism can find all sub-spaces context to them (e.g. devices that are part of spaces) and data connected with these sub-spaces (e.g. data form database tables about DER consumption and occupancy in specific buildings etc.). Note, further specification of the Location and space ontology can be found in D3.2.

### *5.1.3* User Profiling Component requirements

The INERTIA User Profiling approach reflects the need to minimize or eventually eliminate required Occupant Interaction and thus overall Occupant Disturbance during building operations under normal everyday conditions or Demand Response events. The occupant will not have to explicitly define specific operational profiles, instead these will be defined by continuously monitoring user control actions and also reactions (corrective control actions) to specific automated control operations. The whole approach will define the preferences of specific Users or groups of Users. The detailed analysis of the modelling approach of the specific model is out of the scope of this document (will be provided as part of the Deliverable D2.2 - Tertiary Local Control Hubs consumer's flow modelling & profiling). In the following, the main requirements of the User Profiling component reflecting the role of the INERTIA IAM are presented.

The high level categorization is delivered between the different types of occupants. So the User Profiling related data model is divided in (Users_With_ID, Groups_of_Users, Stereotypes). Based on this analysis the detailed attributes for each type of occupant are defined.

Different types of comfort related parameters are examined within INERTIA and thus different types of data will be handled within INERTIA IAM. A high level review of the comfort parameters examined in INERTIA is provided:

#### 5.1.3.1 Data models describing User Preferences

**Visual comfort**

The [lux] measurement value is considered as the main parameter for the extraction of the visual preferences of the INERTIA occupants/groups of occupants. The role of the Visual Comfort related Profiling mechanism is to evaluate the overall preferences on different conditions and thus to define through an implicit mechanism, the limits of the visual comfort characteristics.

- minLum: The low limit that a type of occupant (Individual Occupant, Groups of Occupants, Stereotypes) can accept
- maxLum: The upper limit accepted from a type of occupant
- CurrentLum: The real time measurement within the area examined.
- AverageLum: A statistic measurement as defined through the algorithmic approach
- weighted_Dyn: A complex structure that defines the visual related historical events, referred to the specific type of occupant.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| VisualUserPreferences | maxLum | Luminance | The parameter/value that set the upper boundaries of the visual comfort type examined. A predefined luminance value is considered. |
| | minLum | Luminance | The parameter/value that set the lower boundaries of the visual comfort type examined. A predefined luminance value is considered. |

| | | | |
|---|---|---|---|
| | CurrentLum | Luminance | The parameter/value that set the current value of the visual comfort type examined |
| | AverageLum | Luminance | The parameter/value that set the average value of the comfort type examined. A predefined performance indicator is considered. |
| | Weighted Dynamics | Env Dynamics | The Historical database of environmental conditions, related to the comfort parameter examined |

**Thermal comfort**

The Predicted Mean Vote (PMV) model stands among the most recognized thermal comfort models. Adaptive models of thermal comfort consider occupants as dynamically interacting with their environment, and controlling their thermal comfort by means of clothing or window-opening and closing. Following a similar approach, as the one already provided on the Visual Preferences model, the initial parameters the User profiling model are expressed:

- minPMV: The low limit that a type of occupant (Individual Occupant, Groups of Occupants, Stereotypes) can accept
- maxPMV: The upper limit accepted
- maxPMV: The upper limit accepted from a type of occupant
- CurrentPMV: The real time measurement within the area examined.
- AveragePMV: A statistic measurement as defined through the algorithmic approach
- weighted_Dyn: A complex structure that defines the thermal related historical events, referred to the specific type of occupant.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| ThermalUserPreferences | maxPMV | PMV | The parameter/value that set the upper boundaries of the thermal comfort type examined. A predefined luminance value is considered. |

| | | | |
|---|---|---|---|
| | minPMV | PMV | The parameter/value that set the lower boundaries of the thermal comfort type examined. A predefined luminance value is considered. |
| | Current | PMV | The parameter/value that set the current value of the thermal comfort type examined |
| | PMV | PMV | The parameter/value that set the average value of the thermal comfort type examined.. |
| | Average | Env Dynamics | The Historical database of environmental conditions, related to the thermal comfort parameter examined |

The role of the weighed dynamics (weighted_Dyn) is to express the historical environmental events related to a specific type of occupant. The environmental events will be triggered by the differentiation on the environmental conditions. The types of events are delivered as a complex structure with the specific data fields to be expressed at the relevant Sessions.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| EnvironmentalWeightedDynamics | EnvEvent | Envtimeseries | List of events triggered by environmental conditions |
| | Weight1 | Enum. | Weighted factors based on the analysis of historical environmental conditions (user related conditions) |

The LinkSmart middleware publish events in the format that contains type of events. These events will be stored in IAM in form of DB table records. There will be also DB table for recording weighed dynamics that will contain foreign key of event record. Thus it is possible to associate these dynamic weighting with the LinkSmart events and further retrieve them if needed.

Similarly Occupant ID and group ID can be associated with the record of LinkSmart Event, as there are also fields for identifiers of occupants and occupant group in the DB table for Middleware events.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| EnvironmentalEventsDynamics | Id | String | hh:mm:ss |
| | Time | Time | |
| | Number of Occupants | Integer | [PMV, luminance] |
| | KPIref | Float | |

From all the data fields of the INERIA events, the main related data to be handled by the User Profiling mechanism (set the wieghted_Dyn complex data type) are:

- Id: The Id of the Event
- time: Time period of the event
- Number of Occupants: (Only for the groups of occupants Data Model)
- Space: Space area of the event (Only for the Occupants_With_Id Data Model)
    - **lux** for visual comfort analysis
    - **PMV** for thermal comfort analysis
- Weight1, Weight2, etc.: express some statistical parameters related to the specific types of historical events as extracted from the algorithmic approach. These statistical analysis data are not referred on the whole examination of the environmental events but reflect groups of events.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| ControlDynamics | ControlActions | ControlAction | List of event triggered by the control actions on respective devices |
| | Weights | Enum. | Weighted factors based on the analysis of historical control actions |

| | | | |
|---|---|---|---|
| | | | (Groups of occupants reference) |

| Type name | Attributes | Type | Description |
|---|---|---|---|
| ControlActions | Id | String | The sequence number of event |
| | Time | Time | Time period of the event |
| | DER_Id | String | The ID of the DER triggered by a control event. |
| | Control Act | String | The respective Control Action on the actuator, triggered by the group of users |
| | Number of Occupants | Integer | Number of Occupants on each state |

**Operational comfort (Only for the Occupant_with_ID/ Stereotype Data Model)**

By this term, we mention the operation of DERs that is totally independent from the environmental conditions. This list expresses the preferences on the operational appliances as examined within the INERTIA project. These preferences are mainly defined through the operational status of the devices.

Refrigerator (Stereotype Users)
- Set point. The preferred temperature set-point (initial setting)
- Weighted_Dyn. A statistical analysis referred to the potential of using the device when the occupant is in the near area

Electric Vehicles (Individual Users only)
- Set point. The average value of SoC
- Time_Out. The average value at the end of charging process.
- Weighted_Dyn. A complex structure that refers to the historical charging processes.

Dishwasher (Stereotype Users)
- Set point. The value of operational cycle preferred.
- Time_Out. The average value at the end of operational process.
- Weighted_Dyn. A complex structure that refers to the historical operations of the dishwasher.

Water Heater (Individual Users only)
- Set point. The average value of the operational temperature.
- Weighted_Dyn. A complex structure that refers to the historical operations of the Water Heater.

Th Operational comfort data types are defined in D2.2 (section 7.1.3). Other smart devices will be defined in forthcoming deliverables mainly in the report D2.4.2.

Based on the above mentioned analysis of the INERTIA types of Occupants and the comfort parameters to be examined as part of the User Preferences model, the main User Profiling Component Requirements from IAM are provided in a structural way.

**Individual Id User/ Stereotype data models details**

The following diagram presents the main parts of the Individual profiling modelling approach. The whole dataset is retrieved from the database when the learning process is considered and also the review schema is provided in the database when the total learning process is over. In addition this data-schema is requested from User Profiling Component by IAM when the estimation of the Utility Function value is needed. As this data model reflects the role of the Individual User and the stereotype a detailed definition of the specific roles is provided:

**Individual Id User:** A user with a specific ID {RFID ID}. The role of the user is dependent on the specific devices that have been initially set through the Interface.

**Stereotype:** The INERTIA Stereotypes are considered for specific areas {kitchenette, bathroom} where the patterns of usage are common. The environmental related events are handled by a common way, as defined in INERTIA process, and the whole differentiation is delivered for the specific devices in these areas {water heater, refrigerator, dishwasher, other smart appliances}.

The whole analysis is delivered for the Individuals with ID, as the stereotype role is triggered for the specific devices as above mentioned.

The main details are considered:

- Id: The RFID_Id of the Occupant_With_ID or the ID of the Stereotype reference {kitchenette ID, etc..}
- Role: The role of the occupant (defined by the explicit profiling mechanism - Information Stored in IAM)
- Artefacts: A list of DERs (Unique Id) that the occupant uses (defined by the explicit profiling mechanism - Information stored in IAM)
- Spaces: A list of spaces that the occupant uses (defined by the explicit profiling mechanism- Information stored in IAM)
- Explicit Preferences: This is a list of preferences as defined by the explicit mechanism. These are defined as:
- LumPref: A unique index as defined by the explicit mechanism
- PMVPref: A unique index as defined by the explicit mechanism
- OperPref: A list of preferences on different devices

- Operational Preferences are further handled by taking into account the selection of DERs from the external UI.
- Environmental Preferences: There are three main categories of preferences as extracted from State of the Art analysis (visual, thermal, operational).
- The specific fields have been analysed in the introductory reference of the User Profiling Component
- Control actions: Is a list of historical events connected to the Individual User.
- Control Actions (Id, time, DERid). This type of information will be defined by the INERTIA event mechanism.
- Id: The id of the actuator event
- time: The time of the event
- DERid: This part does not refer only to the controlled DER but also to spatial characteristics (space of DER) and to the different types of control actions that could be potentially delivered to the specific DER (This is actually part of the DER Modeling)
- Weigth1, Weigth2, Weigth3. These represent different statistics on events that are useful for the User profiling mechanism. They provide statistical parameters for the control actions of the Users and are refined during a continuous learning procedure.

Based on the mentioned data model of the Individual User, a list of request/receive methods are defined.

#### 5.1.3.2 Interfaces for individual occupant

The initial methods are defined in order to handle the different types of events as examined in Middleware specs.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOccupantEvents | fromTime | dateTime | |
| | toTime | dateTime | |
| | eventTypeID | string | |
| | occupantID | String | |
| return | events | Array of Events | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| correlateEventToOccupant | eventID | String | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| assignOccupantWeightToEvent | weightID | String | |
| | weightValue | Float | |
| | eventID | String | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOccupantWeightsForEvent | weightID | String | |
| | eventTypeID | String | |
| | occupantID | String | |
| return | occupantWeightsForEvent | Array of Float | |

Apart from the analysis of INERTIA events, some parameters are handled by the User Profiling Model. The following methods define some settings related to the visual preferences of the Occupant with Id

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantMinLuminance | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantMaxLuminance | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantCurrentLuminance | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantAverageLuminance | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

While the update of the luminance parameters are defined in a clearly distinct way, the request of these parameters is delivered with a single method

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOccupantLuminance | occupantID | String | |
| return | luminance | Luminance | |

The additional analysis is also delivered for the thermal related aspects as examined.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantMinPMV | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantMaxPMV | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantCurrentPMV | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantAveragePMV | value | Float | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOccupantPMV | occupantID | String | |
| return | pmvValue | PMV | |

While the previous analysis reflects the environmental related aspects, the operation of the INERTIA devices is handled by the following methods, related to the specific INERTIA Devices. This analysis is considered only for the devices, assigned to a specific user (Individual User with Id).

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantEV | newValue | EV | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOccupantWaterHeater | newValue | WaterHeater | |
| | occupantID | String | |
| return | operationSuccess | Boolean | |

The previous approach, related to operational devices will be delivered for the final list of the INERTIA devices. **An extra set of methods should be handled in order to manage the external preferences. As these methods will be further examined as part of the Ambient User Interface analysis, a short reference on these methods is provided at this part of the work:**

- **readUserProfile**: The role of this method is to get the individual preferences as set through the Ambient UI.

As a consequence to the Individual User analysis, the respective analysis is delivered for the stereotypes as examined in INERTIA project.

#### 5.1.3.3 Interfaces for Stereotypes occupant

The stereotypes events are handled when the ID of the occupants on the specific areas {kitchenette, bathroom} is not known. Actually the whole methods approach remains the same, while the only differentiation is delivered for the specific operational devices, as examined in the INERTIA project {dishwasher, water heater, refrigerators etc..}

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getStereotypesEvents | fromTime | dateTime | |

| | | | |
|---|---|---|---|
| | toTime | dateTime | |
| | eventTypeID | string | |
| | stereotypeID | String | |
| return | events | Array of Events | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| correlateEventToStereotypes | eventID | String | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| assignStereotypesWeightToEvent | weightID | String | |
| | weightValue | Float | |
| | eventID | String | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getStereotypesWeightsForEvent | weightID | String | |
| | eventTypeID | String | |
| | stereotypeID | String | |
| return | stereotypesWeighsForEvent | Array of Float | |

Apart from the analysis of INERTIA events, some parameters are handled by the User Profiling Model. The following methods define some settings related to the visual preferences of the Occupant with Id

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesMinLuminance | value | Float | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesMaxLuminance | value | Float | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesCurrentLuminance | value | Float | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesAverageLuminance | value | Float | |
| | stereotypeID | String | |

| | | | |
|---|---|---|---|
| return | operationSuccess | Boolean | |

While the update of the luminance parameters are defined in a clearly distinct way, the request of these parameters is delivered with a single method

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getStereotypesLuminance | stereotypeID | String | |
| return | luminanceValue | Luminance | |

The additional analysis is also delivered for the thermal related aspects as examined.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesMinPMV | value | Float | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesMaxPMV | value | Float | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesCurrentPMV | value | Float | |
| | stereotypeID | String | |

| | | | |
|---|---|---|---|
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesAveragePMV | value | Float | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getStereotypesPMV | stereotypeID | String | |
| return | pmvValue | PMV | |

While the previous analysis reflects the environmental related aspects, the operation of the INERTIA devices is handled by the following methods, related to the specific INERTIA Devices. This analysis is considered only for the devices, assigned to a specific user (Individual User with Id.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesWaterHeater | newValue | WaterHeater | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesRefrigeratorSetpoint | newValue | Refigerator | |
| | stereotypeID | String | |

| | | | |
|---|---|---|---|
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateStereotypesDishwasher | newValue | Dishwasher | |
| | stereotypeID | String | |
| return | operationSuccess | Boolean | |

#### 5.1.3.4 Data models describing Group of Users

The **group of Users** mechanism is considered in a space with the following conditions:

- When the Number of Occupants > 1 ,
- The Number of Occupants> Number of Known_Id Occupants

Then the preferences within the area are expressed as part of a grouping mechanism. The data model for the profiling is expressed:

The main parts of the model are defined:

- Roles: Defined at an initial stage on settings parameters
- Space: The specific area examined (defined by initial settings)
- Artefacts: The list of DERs within the areas (through BIM Model)
- Preferences: The preferences of the group of occupants (Same data model for Visual and Thermal Comfort parameters as above)
- Control Actions. The control actions by the group of occupants (Same data model as the previous one - Occupants_ with_ Id)

The group of occupants analysis is delivered for the specific areas, where the role of the group is known {researcher's offices, administrative staff offices} and the whole analysis is related to the environmental conditions in the specific area. The total number of occupants will be handled in this scenario analysis {this is the core differentiation from stereotypes analysis} as different patterns are going to be examined in the algorithmic approach.

Based on the above mentioned requirements for the User Profiling Component, the following Table provides the list of methods needed for the extraction of the related to the User Profiling Component information.

#### 5.1.3.5 Interfaces for groups of occupants

As in the Individual Users/Stereotypes analysis, a series of methods for the interconnection of the Groups of Users with the specific events is provided:

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getGroupEvents | fromTime | dateTime | |
| | toTime | dateTime | |
| | eventTypeID | string | |
| | groupID | String | |
| return | groupEvents | Array of Events | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| correlateEventToGroup | eventID | String | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| assignGroupWeightToEvent | weightID | String | |
| | weightValue | Float | |
| | eventID | String | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getGroupWeightsForEvent | weightID | String | |
| | eventTypeID | String | |
| | groupID | String | |
| return | groupWeights ForEvent | Array of Float | |

The groups of occupants analysis mainly refers to the interaction of the Users with the environmental related devices. Thus the data model provide reflects on the operation of HVAC and lighting DERs.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupMinLuminance | value | Float | |
| | groupID | String | |
| return | operationSucc ess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupMaxLuminance | value | Float | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupCurrentLuminance | value | Float | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupAverageLuminance | value | Float | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getGroupLuminance | groupID | String | |
| return | luminanceValue | Luminance | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupMinPMV | value | Float | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

For the thermal comfort parameters of the group of occupants the additional methods are defined.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupMaxPMV | value | Float | |
| | groupID | String | |
| return | operationSucc | Boolean | |

| | | | |
|---|---|---|---|
| | ess | | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupCurrentPMV | value | Float | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateGroupAveragePMV | value | Float | |
| | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getGroupPMV | groupID | String | |
| return | pmvValue | PMV | |

#### 5.1.3.6 Impact of User preferences requirements on INERTIA CIM

**Individual User Model on INERTIA CIM**

The alignment of the User Profiling requirements with the respective structure of INERTIA IAM is further analysed. The INERTIA common information model will define the high level domain model comprising the basic elements as examined in the INERTIA User Profiling Mechanism.

**Visual comfort model and INERTIA CIM:** The measurements and corresponding units described in Unit XSD schemas and further in Unit semantic model will be part of the INERTIA CIM model. The specification of the unit model will be done in a unit-related ontology. The measurements stored in DB table (specification depends on the format of events) are expected to have annotations to the unit-related ontology (the ontology definition can be found in D3.2).

The general limits describing user preferences are also expected to be specified in the respective user preference part of ontology, since they are considered as static data (the ontology definition can be found in D3.2).

**Thermal comfort models and INERTIA CIM:** The initial parameters/limits as well as the above outlined structure of weighed dynamics will be reflected in the INERTIA CIM to be able to support framework components when analysing historical events related to the specific occupant. They will be specified in form of ontology (see D3.2), since they are considered as static data. The dynamic data describing user comfort will be annotated to these static data, thus (based on component demand) extendable with the situational context from ontology. The DB schema for dynamic data is proposed in Chapter 5.
**Operational comfort and INERTIA CIM:** Operational comfort preferences are considered as dynamic data, thus saved in IAM in DB tables. The annotation of entries in these table to the DER ontology will enable to enhance data with situational context of DERs (e.g. energy profile of a appliance). The DB schema for dynamic data of Operational comfort is proposed in Chapter 5 and the final DER ontology will be described in the deliverable D3.2 (initial version can be found in D3.1).

#### 5.1.3.7 Impact of Group user model on INERTIA CIM

In case data about group of used will be stored and retrieved the above outlined schema will serve for data serialisation of the IAM interfaces based on CIM. The static data such as user roles, spaces (BIM model) DER static data (e.g. energy profiles) typical preferences (templates) will be specified in the respective ontology (defined in D3.2). The dynamic data will be stored in DB table. There will be annotations of these table entities to the ontology (e.g. URI of space, URI of user role) that will enable to enhance them (the data stored in the DB tables). The same approach is delivered for the additional types of User Preferences as examined within the INERTIA project (**Visual, Thermal, Operational** Comfort).

#### 5.1.3.8 Impact of User Preferences on INERTIA CIM

The mapping between User Preferences (UP) Interfaces and IAM Interfaces is in the following table. Many specific UP interfaces were unified since they are operating with the same data types and semantically this data do not differs much (e.g. occupantID and stereotypeID are basically the same but first is for known occupant while second is for unknown occupant). Also the specific UP updates of user preferences values can be unified into one IAM interface that uses the data structure describing all user preferences.

***Table 10: Mapping between User UP interfaces of and IAM interfaces***

| UP interface | IAM interface |
|---|---|
| getOccupantEvents<br><br>getStereotypesEvents | UP_getOccupantEventsOrStreotypes |
| correlateEventToStereotypes | UP_correlateEventToOccupantOrStr |

| | |
|---|---|
| correlateEventToOccupantOrStreotypes | eotypes |
| assignOccupantWeightToEvent<br><br>assignStereotypesWeightToEvent | UP_assignOccupantWeightToEventOrStreotypes |
| getOccupantWeightsForEvent<br><br>getStereotypesWeightsForEvent | UP_getOccupantWeightsForEventOrStreotypes |
| updateOccupantMinLuminance<br><br>updateStereotypesMinLuminance | UP_updateOccupantOrStereotypeMinLuminance |
| updateOccupantMaxLuminance<br><br>updateStereotypesMaxLuminance | UP_updateOccupantOrStereotypeMaxLuminance |
| updateOccupantCurrentLuminance<br><br>updateStereotypesCurentLuminance | UP_updateOccupantOrStereotypeCurrentLuminance |
| updateOccupantAverageLuminance<br><br>updateStereotypesAverageLuminance | UP_updateOccupantOrStereotypeAverageLuminance |
| updateOccupantMinPMV<br><br>updateStereotypesMinPMV | UP_updateOccupantOrStereotypeMinPMV |
| updateOccupantMaxPMV<br><br>updateStereotypesMaxPMV | UP_updateOccupantOrStereotypeMaxPMV |
| updateOccupantCurrentPMV<br><br>updateStereotypesCurrentPMV | UP_updateOccupantOrStereotypeCurrentPMV |
| updateOccupantAveragePMV<br><br>updateStereotypesAveragePMV | UP_updateOccupantOrStereotypeAveragePMV |
| updateOccupantEV | UP_updateOccupantEV |
| updateOccupantWaterHeater<br><br>updateStereotypesWaterHeater | UP_updateOccupantOrStereotypeWaterHeater |
| updateStereotypesRefrigeratorSetpoint | UP_updateOccupantOrStereotypeRefrigeratorSetpoint |

| | |
|---|---|
| updateStereotypesDishwasher | UP_updateOccupantOrStereotypeDishwasher |
| getOccupantLuminance<br><br>getStereotypesLuminance | visualPreferencesType UP_getOccupantOrStereotypeLuminance(<br><br>String occupantID<br><br>) |
| getOccupantPMV<br><br>getStereotypesPMV | thermalPreferences UP_getOccupantOrStereotypePMV(<br><br>String occupantID) |
| getGroupEvents | UP_getGroupEvents |
| correlateEventToGroup | UP_correlateEventToGroup |
| getGroupWeightsForEvent | UP_getGroupWeightsForEvent |
| updateGroupMinLuminance | UP_updateGroupMinLuminance |
| updateGroupMaxLuminance | UP_updateGroupMaxLuminance |
| updateGroupCurrentLuminance | UP_updateGroupCurrentLuminance |
| updateGroupAverageLuminance | UP_updateGroupAverageLuminance |
| updateGroupMinPMV | UP_updateGroupMinPMV |
| updateGroupMaxPMV | UP_updateGroupMaxPMV |
| updateGroupCurrentPMV | UP_updateGroupCurrentPMV |
| updateGroupAveragePMV | UP_updateGroupAveragePMV |
| getGroupPMV | thermalPreferences UP_getGroupPMV(String occupantID) |
| getGroupLuminance | visualPreferencesType UP_getGroupLuminance(String occupantID) |

### *5.1.4* Ambient User Interface requirements analysis

The input/output interfaces of Ambient User Interface (UI) which manipulate with the information about user profiles from the Information Access perspective are analysed in the following. These input/output interfaces are specified by the signatures of their call. The specific data types are specified by XSD schema. These interfaces are analysed with the intention to map them into the IAM interfaces. The whole analysis is based on the different types of INERTIA Ambient User Interfaces as have already been provided in the INERTIA Reference Architecture document (Deliverable D1.3 INERTIA Framework Architecture including functional elements and technical specifications). Based on the role and the functionality of the respective architecture elements, the main INERTIA User Interfaces, as provided in the DoW are:

- **for the occupants:** smartphones to visualize high-level information concerning the energy consumption of area, in order to increase the occupants' awareness on how they may affect (and reduce) the energy. In addition, the User Occupants should provide the additional individual parameters

- **for groups of occupants:** public screens to visualize high-level information concerning the energy consumption of area, together with contextual information. The analysis is based on a location based structure and this reflects the requirements through IAM

- **for the manager of the building**, who is responsible for its energy consumption: pc to visualize real-time (aggregated and correlated) information on the energy consumption of the building, in order to have real-time information to apply suitable energy strategies to reduce the consumption and negotiate the price of the energy with the energy provider.

The holistic data model is based on the different types of end users in the premises examined, and the functionality already define in the specifications document. Based on this categorization, a detailed overview of the Ambient UIs data model is considered.

#### 5.1.4.1 Data models describing Individual User Interface

The initial selection of the data elements consist of the data from the analysis of individual preferences. From individual occupants, where the Id is considered as known, different types of data are considered. The specific data structure is then provided.

The real time information is provided through the respective component element (monitoring component) as part of the INERTIA Local Hub Multi-Agent Architecture. The **Historical energy measurements**, together with some User related Performance metrics are provided through INERTIA IAM. The different types of metrics, are provided through the analysis of the INERTIA types of events (environmental events, control actions events). No computational effort is considered, as the whole estimation is delivered through the User Profiling mechanism and the additional information is stored in IAM.

- Environmental Events

- Min-Max Value of Preferred environmental conditions (Temperature-Luminance)

- Average Value of environmental conditions(Temperature- Luminance)

- Most common accepted environmental conditions(Temperature-Luminance)
- Control Events
- Total number of control actions events (frequency of manual control actions per type of comfort)

In addition, the **external settings about preferences** (comfort related preferences, operation related preferences) are also available. Moreover, the **occupancy schedules** as defined from the User Interface are handled within INERTIA IAM. As an extra functionality, the historical estimation, in a personalized way, is provided through the Interface. The total estimation of energy consumption is based on the available in IAM data.

#### 5.1.4.2 Interfaces of external settings about preferences

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertUserProfile | Role | Role | |
| | Array of DER Ids | String array | |
| | Array of Space IDs | String array | |
| | TemperaturePref | Float | |
| | LuminancePref | LuminancePref | |
| | EV_time_in | dateTime | |
| | EV_time_out | dateTime | |
| | EV_Charging_Level | Float | |
| | Hot_Water_Preferences | Float | |
| return | id | String | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateUserProfile | Role | Role | |
| | Array of DER Ids | String array | |

| | | | |
|---|---|---|---|
| | Array of Space IDs | String array | |
| | TemperaturePref | Float | |
| | LuminancePref | LuminancePref | |
| | EV_time_in | dateTime | |
| | EV_time_out | dateTime | |
| | EV_Charging_Level | Float | |
| | Hot_Water_Preferences | Float | |
| return | operationSuccess | Boolean | |

| **Method name (import/export)** | **Attributes** | **Type** | **Description** |
|---|---|---|---|
| readUserProfile | occupantID | String | |
| return | userProfile | ExtPreferences | |

The ExtPreferences data type has been defined as part of the Deliverable D.2.

#### 5.1.4.3 Interfaces for historical energy measurements

| **Method name (import/export)** | **Attributes** | **Type** | **Description** |
|---|---|---|---|
| getOccupantHistorical DERData | occupantID | String | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| | DER_ID | String | |
| | timestep | Int (num of seconds) | |

| | | | |
|---|---|---|---|
| return | ocupantHistDER | Array of Float | |

**User related Performance metrics:** The additional methods are available in the User Profiling reference analysis

**Occupancy schedules:** The reference is based on the occupancy related settings as provided in **Occupancy flow and prediction** data model (OCC. *getOccupantSchedule*)

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOccupantSchedule | occupantID | String | |
| | dateFrom | dateTime | |
| | dateTo | dateTime | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| return | occupantSchedule | Schedule | |

The additional method is provided in order to store the initial parameter for the schedules of occupants

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertOccupantSchedule | occupantID | String | |
| | dateFrom | dateTime | |
| | dateTo | dateTime | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| return | scheduleID | String | |

The Schedule datatype has been defined in the D2.2

#### 5.1.4.4 Data models for Location Based UI (Groups of Occupants)

The initial selection of the data elements consist of the data related to the specific area examined. The specific data structure is provided. The as mentioned functionality remains the same apart from the need to establish the individual preferences. Thus the main data models needed from IAM are presented.

- Environmental Events related Metrics
- Min-Max Value of Preferred environmental conditions (Temperature-Luminance)
- Average Value of environmental conditions(Temperature- Luminance)
- Most common accepted environmental conditions(Temperature-Luminance)
- Control Events related Metrics
- Total number of control actions events (frequency of manual control actions per type of comfort)
- Historical measurement of energy consumption
- For a predefined period, the energy consumption estimation for the respective devices, area.
- Historical occupancy data
- For a predefined period, the occupancy data estimation for the respective area.

Thus, the respective data model for the location based analysis is provided.

#### 5.1.4.5 Interfaces for historical measurement of energy consumption (Groups of Occupants)

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getGroupHistoricalDERData | groupID | String | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| | DER_ID | String | |
| | timestep | Int (num of | |

| | | | |
|---|---|---|---|
| | | seconds) | |
| return | groupHistoricalDER | Array of Float | |

**Groups of Users related Performance metrics:** The additional methods are available in the User Profiling reference analysis

**Schedules:** The reference is based on the occupancy related settings as provided in **Occupancy flow and prediction** data model (OCC.*getSpatialSchedule*)

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getSpatialSchedule | groupID | String | |
| | dateFrom | dateTime | |
| | dateTo | dateTime | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| return | spatialSchedule | Schedule | |

The additional method is provided in order to store the initial parameter for the schedules of groups of occupants {settings from the AUI of the INERTIA project}

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertSpatialSchedule | groupID | String | |
| | dateFrom | dateTime | |
| | dateTo | dateTime | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| return | scheduleID | string | |

**Historical occupancy data:** The reference is based on the occupancy related settings as provided in **Occupancy flow and prediction** data model (OCC.getHistoricalOccupancy)

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getHistoricalOccupancy | groupID | String | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| | timestep | Int (num of seconds) | |
| return | historicalOccupancy | Array of Float | |

**Role of Group:** The role of occupants {researchers, administrative staff} in the specific area examined is provided

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertGroupRole | groupID | String | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getGroupRole | groupID | String | |
| return | groupRole | string | |

#### 5.1.4.6 Impact of Occupants Ambient User Interface on INERTIA CIM

The Ambient User interface component (UI) has both importing and exporting interfaces which are affecting CIM based interfaces of IAM. Their impact on INERTIA CIM is analysed in the following.

The following table provides mapping between UI importing interfaces (getters) and interfaces of IAM that are intended to serve them. Thus these IAM interfaces are first part of CIM vocabulary definition that need to be known by the UI component (but also other component) to be able to use it. The names of the IAM interfaces begin with “UI_” and continue with whole name of UI interface name to be able to easier distinguish the bindings between them in the implementation.

As the several INERTIA User Interface have main commonalities, a per category view of the data model requirements is provided. The holistic analysis remains the same, providing the main input/output methods of the Ambient User Interface Components to IAM

***Table 11: Mapping between exporting Individual User Interfaces (retrieval) of UI and IAM interfaces***

| Nr | UI interface | IAM interface |
|---|---|---|
| 1 | *readUserProfile* | UI_readUserProfile |
| 2 | *getOccupantHistoricalDERData* | UI_getGroupHistoricalDERData |
| 3 | *getOccupantSchedule* | OCC_exportIndividualInertiaOccupancyAndFlowModel |

***Table 12: Mapping between Individual UI Interfaces (storage) and IAM interfaces***

| Nr | UI interface | IAM interface |
|---|---|---|
| 1 | *insertUserProfile* | UI_insertUserProfile |
| 2 | *updateUserProfile* | UI_updateUserProfile |
| 3 | *insertOccupantSchedule* | OCC_importIndividualInertiaOccupancyAndFlowModel |

***Table 13: Mapping between Location Based UI Interfaces (retrieval) and IAM interfaces***

| Nr | UI interface | IAM interface |
|---|---|---|
| 1 | *getGroupHistoricalDERData* | UI_getGroupHistoricalDERData |
| 2 | *getSpatialSchedule* | OCC_importSpatialSchedule |
| 3 | *getHistoricalOccupancy* | OCC_importHistoricalOccupancy |
| 4 | *getGroupRole* | UI_getGroupRole |

***Table 14: Mapping between Location Based UI Interfaces (storage) and IAM interfaces***

| Nr | UI interface | IAM interface |
|---|---|---|
| 1 | *insertSpatialSchedule* | OCC_exportSpatialSchedule |

| | | |
|---|---|---|
| 2 | *insertGroupRole* | UI_insertGroupRole |

#### 5.1.4.7 Data models for Facility Manager UI (Facility Manager)

The selection of the data elements consist of the data related to the specific role of the facility manager as examined in the INERTIA project. The related functionality is further analysed.

- BIM Model (Areas and DERs in the areas)
- Environmental Events related Metrics
- Min-Max Value of Preferred environmental conditions for occupants-groups of occupants.
- Average Value of environmental conditions for ID_occupants-groups of occupants
- Control Events related Metrics
- Total number of control actions events (frequency of manual control actions per type of comfort)
- Historical measurement of energy consumption
- For a predefined period, the energy consumption estimation for the respective devices, area, occupant etc..
- Occupancy related data. As stored from the Occupancy profiling component
- Occupancy prediction data. As stored from the Occupancy profiling component
- Occupancy related schedules data. As defined from Single Occupants/ Groups of occupants in a specific location.
- Operational mode for the building. Operational settings for the different types examined in INERTIA (Green Operation, Cost efficient Operation, Sustainable Operation, On Vacation Operation)
- CO2 emissions rate. (Predefined value)
- Price value (Dynamic price policy)
- Logging mechanism for abnormal situations in the infrastructure.
- DER static information based on the specific DERs available in CERTH

The respective data are available in IAM and the role of the Ambient User Interface is to request these data and present them through a visual interface. The reverse functionality, the storage of Facility Manager settings in IAM is further provided.

- Facility Manager schedules data. The facility Manager will provided additional schedules for specific areas in the infrastructure
- Operational mode for the building. Operational settings for the different types examined in INERTIA (Green Operation, Cost efficient Operation, Sustainable Operation, On Vacation Operation)
- CO2 emissions rate. Define the CO2 emissions rate
- Price value. Define the reference Price Values.
- Historical measurement energy consumption. Define the time period
- For a predefined period, the energy consumption estimation for the respective devices, area, occupant etc.

Based on the above mentioned functionalities of the facility manager, the respective methods are provided.

The get.BIM method is a core method in order to define, in a visual way, the premises examined. The additional attributed of this method will be further defined.

The role of the facility manager is to manage the whole operation of the premises examined and thus a list of static parameters {energy price, $CO_2$ rates} is defined:

#### 5.1.4.8 Interfaces Facility Manager User Interface

**Energy price, $CO_2$ rates**

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertTariff | description | String | |
| | value | Float | |
| | effectiveFrom | long (second offset) | |
| | effectiveTo | long (second offset) | |
| return | id | String | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateTariff | tariffID | string | |
| | description | String | |
| | value | Float | |
| | effectiveFrom | long (second offset) | |
| | effectiveTo | long (second offset) | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getTariffs | | | |
| return | tariffs | Array of Tariff | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateEmissions | newValue | Float | |
| return | operationSuccess | Boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getEmissions | | | |
| return | emissions | float | |

In addition, the Facility Manager will need to handle the operation {direct control operation} on different types of the Devices {HVAC} and thus the different DER parameters, as defined in the DER Modelling part of the work are further provided.

**DER static parameters**

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getlistofDER | spaceID | String | |
| return | DERs | List of DER datatype | |

Apart from the static information, a list of energy measurement, occupancy measurement aspects will be handled by the facility manager. The core methods are further analysed

**Energy measurement data**

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getHistoricalDERData | DER_ID | String | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| | timestep | Int (num of seconds) | |
| return | historicalDER | Array of Float | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getHistoricalDataForSpace | spaceID | String | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| | timestep | Int (num of seconds) | |
| return | histDatForSpace | Array of Float | |

**LoggingData method**

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getLoggingData | spaceID | String | |
| | timeFrom | dateTime | |
| | timeTo | dateTime | |
| return | logs | Array of Float | |

#### 5.1.4.9 Data models related to Occupancy

The main functionalities have been defined, and thus the additional methods are available from the previous analysis (occupancy flow and prediction model, Individual/ Groups User Interface Model):

- **Occupancy related data**. As stored from the Occupancy profiling component
- **Occupancy prediction data**. As stored from the Occupancy profiling component
- **Occupancy related schedules data**. As defined from Single Occupants/ Groups of occupants in a specific location.

The additional methods are mentioned:

- getHistoricalOccupancy
- getHistoricalOccupantLocation
- getRealTimeOccupancy
- getRealTimeOccupantLocation
- getSpatialSchedule
- getOccupantSchedule
- insertSpatialSchedule

#### 5.1.4.10 Data models for Operational modes of the Infrastructure

The additional types of operational modes are defined in the INERTIA project

(Operational Mode ID: 1--> Green Operation

Operational Mode ID: 2--> Cost efficient Operation

Operational Mode ID: 3--> Green Operation

Operational Mode ID: 4--> On Vacations Operation

Operational Mode ID: 0--> DR operation

)

As the below mentioned methods are initially defined at this part of the work, a list of additional data types is provided in order to cover the main aspects examined.

**Data models related operational mode selection**

These data types define the different aspects to be handled as part of the operational mode selection.

| Type name | Attributes | Type | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| OperationalInterval | DERId | ID | The ID of the DER for which the interval is given. |
| | IntervalList | Array of Interval | Reference to the intervals associated with this DER.<br><br>Temperature: oC<br><br>Humidity: %<br><br>Luminance: lux |

| Type name | Attributes | Type | Description |
|---|---|---|---|
| Interval | IntervalID | ID | The ID of the Interval. |
| | OperationalModeId | ID | The ID of the operational mode this interval refers to for this DER |
| | isOn | Boolean | Whether this is an on or an off interval |
| | from | dateTime | The starting point in time for the operation. |
| | to | dateTime | The ending point in time for the operation |

Operational mode specification is in the following.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| OperationalMode | OpModeId | ID | The ID of the operational mode. |
| | Description | String | Textual description. |
| | isActive | Boolean | Whether this operational mode is currently active |

#### 5.1.4.11 Interfaces related to Operational modes

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertOperationalMode | OpMode | OperationalMode | |
| return | Id | String | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateOperationalMode | OpMode | OperationalMode | |
| return | operationSuccess | boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| deleteOperationalMode | OpModeId | String | |
| return | operationSuccess | boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getListOfDERsForOpMode | OpModeID | String | |
| return | DERs | Array of DER | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getOpIntervalforDERForOpMode | DERid | String | |
| | OpModeId | String | |

| | | | |
|---|---|---|---|
| return | opIntervalForDED | Array of OperationalInterval | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| setOpIntervalforDER<br><br>forOpMode | DERid | String | |
| | OpModeId | String | |
| | IntervalList | Array of OperationalInterval | |
| return | operationSuccess | boolean | |

### 5.1.4.12 Impact of Facility Manager User Interface on INERTIA CIM

The list of additional data types with the aspects defined above will be reflected in the XSD schemas of CIM as well as in the DB tables that will enable to store these kinds of data.

The mirroring interface to the interfaces defined for Facility Manager Component will be part of CIM definition. The mapping between Facility Manager Interfaces UI and IAM interfaces is provided in the following.

***Table 15: Mapping between Individual User Interfaces (retrieval) of UI and IAM interfaces***

| Nr | UI interface | IAM interface |
|---|---|---|
| 1 | *getHistoricalDERData* | UI_getHistoricalDERData |
| 2 | *getHistoricalDataForSpace* | UI_getHistoricalDataForSpace |
| 3 | *getHistoricalOccupancy* | OCC_importHistoricalOccupancy |
| 4 | *getHistoricalOccupantLocation* | OCC_importHistoricalOccupantLocation |
| 5 | *getRealTimeOccupancy* | OCC_importRealTimeOccupancy |
| 6 | *getRealTimeOccupantLocation* | OCC_importRealTimeOccupantLocation |
| 7 | *getSpatialSchedule* | OCC_importSpatialSchedule |

| | | |
|---|---|---|
| 8 | *getOccupantSchedule* | OCC_importIndividualInertiaOccupancyAndFlowModel |
| 9 | *getTariffs* | UI_getTariffs |
| 10 | *getEmissions* | UI_getEmissions |
| 11 | *getListOfDERsForOpMode* | *UP_getListOfDERsForOpMode* |
| 12 | *getOpIntervalforDERForOpMode* | tbd |
| 13 | *get.BIM* | getLocationModel |
| 14 | *get.DERModeling* | getDERModel<br><br>getDeviceLocationModel |
| 15 | *getGroupPMV* | thermalPreferences UP_getGroupPMV(String occupantID) |
| 16 | *getGroupLuminance* | visualPreferencesType UP_getGroupLuminance(String occupantID) |
| 17 | *getUserpPMV* | thermalPreferences UP_getOccupantOrStereotypePMV(<br><br>String occupantID) |
| 18 | *getUserLuminance* | visualPreferencesType UP_getOccupantOrStereotypeLuminance(<br><br>String occupantID<br><br>) |
| 19 | *getLoggingData* | UI_getLoggingData |

In addition, the output methods are provided for the storage of UI related Information to IAM.

***Table 16: Mapping between Individual User Interfaces (storage) of UI and IAM interfaces***

| **Nr** | **UI interface** | **IAM interface** |
|---|---|---|
| 1 | *insertOperationalSchedules* | UI_insertOperationalSchedules |

| | | |
|---|---|---|
| 2 | *insertOperationalSchedules* | UI_insertOperationalSchedules |
| 3 | *insertOperationalSchedules* | UI_insertOperationalSchedules |
| 4 | *insertOperationalSchedules* | UI_insertOperationalSchedules |
| 5 | *updateOperationalSchedules* | UI_updateOperationalSchedules |
| 6 | *insertSpatialSchedule* | OCC_exportSpatialSchedule |
| 7 | *insertOperationalMode* | UI_insertOperationalMode |
| 8 | *updateOperationalMode* | UI_updateOperationalMode |
| 9 | *deleteOperationalMode* | UI_deleteOperationalMode |
| 10 | *setOpIntervalforDERforOpMode* | UI_setOpIntervalforDERforOpMode |

### *5.1.5* KPI requirements (Energy Performance Modelling component)

#### 5.1.5.1 Data models describing KPIs

The KPI requirements are coming from their definition in the INERTIA deliverable D2.3.1. This deliverable contains the KPIs categorisation together with KPIs their description, formulas for their calculation, measurement units, granularity and other aspects not mentioned here (they are not relevant for CIM).

The structure of KPI definition consisting from their categorisation, individual KPIs, granularity as well as the measurement units need to be directly reflected in the CIM schema for IAM I/O operations.

KPI categorisation as defined in D2.3.1:

- Energy Related KPIs
- Flexibility Related
- Business Related KPIs
- Comfort Related
- Grid related Operation KPIs

The categorisation of KPIs together with the granularity of KPIs provides a basic schema upon which the CIM schema for results of KPI calculations is build. The granularity of KPIs also influences the information model for measurement, occupancy and user profile data, since they are used for the calculation of KPIs with a given granularity. The other aspects of the INERTIA KPIs that has impact on CIM schema is the granularity of each KPI. Choosing the granularity of the energy consumption KPI, the information model need contain references structured as following:

Time reference

- monthly
- daily
- minutes (for DR operation)

Space reference

- per room
- per zone
- total per MV

Business reference

- High Price period
- Low Price period
- per Device
- per Operation

Occupancy reference

- per occupant
- per group of occupants
- total

Based on the initial analysis of formulas and required granularities of all KPIs the data structures containing energy consumption records per minutes (while time has to be associated to/categorised to the High price period and Low price period), rooms/zones, per role/business activity (i.e. group of user) per occupant, per device and per operation have to be accessible using IAM interface. There is a need to be able to represent RES energy production of local hub and per individual RES, while the Time granularity remains the same as for energy consumption data. Thus these data structures will be part of CIM definition. The data from measurement will be serialised into the form based on this structures and sent to the requesting component responsible to the calculation. The measurements in form of Events will come from Linksmart and will be stored with the appropriate context references (i.e. references to spaces and roles) in IAM. Thus these granularity requirements influence representation of events that will be produced by LinkSmart (simply they need to contain all the references in some form).

#### 5.1.5.2 Interfaces related to KPIs

Using the KPI schema the IAM will be able to parse result from the calculation sent from the component responsible for the calculation (i.e. Local & Global

Energy Performance Modelling component) and store it into DB repository. The following interface specification is expected on the IAM side to be able to store the result of KPI calculation.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| importKPI | spatialIDRef | string | URI of spatial element for which the KPI was calculated |
| | derIDRef | string | URI of der/der cluster for which the KPI was calculated |
| | occupantGroupIDRef | string | URI of occupant group or role for which the KPI was calculated |
| | occupantIDRef | string | URI of individual occupant for which the KPI was calculated |
| | timeFrom | datetime | Date and time from which the KPI was calculated |
| | timeto | datetime | Date and time to which the KPI was calculated |
| | KPI | KPItype | KPI type based on the KPI definition in D2.3.1 |

Providing inputs into calculation of KPIs is another task in which IAM (as common repository for the INERTIA platform) is involved. An illustrative example of the IAM interfaces supporting calculation of one chosen KPI (energy consumption) is described in the INERTIA deliverable D2.3.1. All dedicated IAM interfaces supporting calculations of KPIs with the respective inputs will be analysed in details in the INERTIA deliverable D2.3.2. These interfaces will (mainly) serve to the Local & Global Energy Performance Modelling component.

#### 5.1.5.3 Impact of KPI Requirements to CIM

As mentioned above, the dedicated IAM interfaces supporting calculations of KPIs with the respective inputs will be analysed in details in the INERTIA deliverable D2.3.2. These interfaces will (mainly) serve to the Local & Global Energy Performance Modelling component.

The IAM interface for storing the results of KPIs is proposed above. The DB tables storing KPIs will contain annotation to the static data based on the granularity requested (e.g. KPI value for the whole building or for specific space will refer to these BIM elements in the ontology).

### *5.1.6* Multi Agent Based Local Hub Gateway requirements

#### 5.1.6.1 Data models for Local Hub Gateway:

The flexibility model defines the possible historic and forecast consumption and production per device per local hub per duration.

The specification enables to distinguish:

- consumption of energy of the device per some time – e.g. dishwasher01 will consume 0,5 kWh in next 30 minutes
- generation of energy per device per time – e.g. PV panel will produce 0,5 kWh in next 1 hour
- possible lower consumption of energy per device per some time – e.g. dishwasher01 will lower consumption to 0,3 kWh in next 30 minutes
- possible higher consumption of energy per device per some time – e.g. dishwasher01 will raise consumption to 0,8 kWh in next 30 minutes

If the system will use the flexibility then KPI can be influenced. The comfort is reflected in the flexibility model.

- if system will lower the consumption of dishwasher01 then user will have to wait longer and his/her comfort will be lower
- if system will raise the consumption of dishwasher01 then more energy is consumed for the cleaning process – his/her comfort will be higher

This is modelled by real numbers expressing negative effect and positive effect of the flexibility.

The specification allows different granularities of data, such as seconds inside local hubs and minutes or hours on aggregator or above levels.

Aggregation of record can be achieved, albeit with the loss of detailed sub-windows flexibility information. This is in line with the current view on privacy sensitive information.

The flexibility is described also by device id and hub and/or other higher level hierarchical information. It will depend on the types of aggregation required.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| Forecast Profile | start time | time stamp | start of period |
| | end time | time stamp | end of period |
| | consumption/production | float ((K)Wh) | energy consumed or produced |
| | sample/granularity | Time duration | integration period for observation |
| | positive flexibility | float ((K)Wh) | upward potential per sample |
| | negative flexibility<br>positive comfort cost<br>negative comfort cost | float ((K)Wh)<br>float<br>float | downward potential per sample<br>impact on comfort<br>impact on comfort |

### 5.1.6.2 Interface related to Local Hub Gateway

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| insertFlexibilityForecst | Device ids | List of string | name/identification of device |
| | Forecast profile | Forecast profile type | Forecast profile for Local hub in terms of flexibility |

### 5.1.6.3 Impact on CIM

The forecast profile interfaces will be reflected in the LH IAM, thus the LH IAM will provide mirror interface serving Local Hub Gateway component. The name of this LH IAM interfaces will be part of CIM vocabulary specification.

The profile records specification (like the one described by the parameters above) will be part of CIM vocabulary. The specification in the CIM will be done using XSD schemas. The data of these types will be stored in the hybrid repository of IAM in form of DB tables. The schema definition that enables storing and retrieving such data will be reflected in the CIM vocabulary too. The device identifier will refer to the instance of device identifier in the DER semantic model.

### *5.1.7* Local Hub DER Flexibility Modelling requirements

#### 5.1.7.1 Data models for Local Hub DER Flexibility

The Local Hub flexibility is defined by the amount energy that can be increased and decreased respect to the estimated energy consumption or generation. The reference energy profile is the predicted one and represents the expected energy consumption and generation for a certain time period with the given time step size. The reference energy profile contains one energy value per time step in the profile period. The flexibility is given by the upwards and downwards flexibility profiles. The Upwards flexibility gives the amount of energy that can be increased respect to the expected energy profile and contains one energy value for each time step in the profile period. In the same way, the downwards flexibility indicates how much energy can be reduced respect to the expected profile and contains one energy value for each time step in the profile.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| LocalHubFlexibility | Id_LocalHub | String | Identification of the Local Control Hub |
| | TimeStamp | Timestamp | Date & Time of the flexibility creation |
| | Type | String Enumeration | Type of profile:<br>- Demand Flexibility<br>- Generation Flexibility |
| | Flexibility_Profile | FlexibilityProfile | Flexibility profile |

| Type name | Attributes | Type | Description |
|---|---|---|---|
| FlexibilityProfile | TimeStamp_Begin | Timestamp | Date & Time at which the data set starts. |
| | TimeStamp_End | Timestamp | Date & Time at which the data set ends. |

| | | | |
|---|---|---|---|
| | TimeStamp_Step | Seconds | Time length between two individual steps |
| | Energy_Values | Array of Float | Set of energy values one per step kwh (generation or demand depending on the flexibility type) |
| | Up_Flexibility_Values | Array of Float | Set of upwards flexibility values one per step (kwh)<br><br>Represents the amount of energy in each step that can be increased (generation or demand depending on the flexibility type) |
| | Down_Flexibility_Values | Array of Float | Set of downwards flexibility values one per step (kwh)<br><br>Represents the amount of energy in each step that can be decreased (generation or demand depending on the flexibility type) |

#### 5.1.7.2 Interface for Local Hub DER Flexibility

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| setLocalHubFlexibility | localHubFlexibility | LocalHubFlexibility | Contains information about the flexibility provided by a Local Control Hub |

#### 5.1.7.3 Impact Local Hub DER Flexibility Modelling on CIM

The mirroring interface to the interface setLocalHubFlexibility will be defined in the CIM vocabulary and thus provided by Local Hub IAM. The data type Local hub flexibility with local hub flexibility profile will be part of XSD schema definition of CIM. The data about flexibility are stored in the IAM repository in the DB tables designed for the needs of Multi Agent Based Holistic Flexibility component. Thus, the IAM mirroring interface will retrieve data from IAM repository based on the request from DER Flexibility Modelling

#### 5.1.7.4 Impact of DER model requirements on CIM

Data required for simulation including DER configuration are based on the formulas defined in the INERTIA deliverable D2.4.1. These formulas use monitoring and controlling parameters of device types. These formulas enable to simulate operation aspects (e.g. energy consumption) of devices belonging to DER. These models for monitoring and controlling of devices influence the information model of CIM as following: static data are modeled in CIM as DER semantic model and dynamic data (operation of devices) are modeled as DB tables that are designed to store events coming from the LinkSmart middleware. Some of the aspects of DER model are reflected in the information coming from other INERTIA components. For instance, the User interfacing component uses DER statuses in the user preferences model (operational preferences).

Providing inputs into DER calculation is expected task of IAM. An illustrative example of the use of IAM interfaces supporting calculation of one chosen DER model (Lighting device model parameters) is described in the INERTIA report D2.4.1. The dedicated IAM interfaces supporting inputs needed for the calculations of DER states will be fully defined in the INERTIA final report D2.4.2 (INERTIA Multi-dimensional DER flexibility models).

## 5.2 Aggregator Control Hub CIM

The Aggregator Control Hub CIM (referred as Aggregator CIM in the following) serves as a commonly shared vocabulary by to aggregator control HUB INERTIA framework components. This commonly shared vocabulary of CIM needs to define IAM interfaces serving for Aggregator control hub components INERTIA specific data types used in these interfaces in form of XSD schemas. At last Aggregator CIM defines DB schemas of tables and relevant annotations to semantic models for storing the data coming from these components and retrieving and serializing data that are requested from IAM by these components.

To be able to identify the Aggregator CIM elements, the requirements to Aggregator IAM (serving as a repository service) from the aggregator components were collected in the predefined template form (see Annex 1). These requirements specify the Aggregator component interfaces and the INERTIA specific data types. The collected requirements in this format were then analysed from the perspectives of CIM elements (interfaces, XSD schemas, DB table schemas, Semantic models).

The requirements from Aggregator components on Aggregator IAM and their analysis from the perspective of Aggregator CIM elements are in the following.

### *5.2.1* Prosumers Control & Optimization Component requirements

#### 5.2.1.1 Data models describing Prosumers Control & Optimization

The flexibility model defines the possible historic and forecast consumption and production per local hub per duration.

The forecast flexibility profile form Local Hub MAS Based Local Hub Gateway is used in the interface of Prosumers Control & Optimization Component.

#### 5.2.1.2 Interfaces for Prosumers Control & Optimization

| Method name (import) | Attributes | Type | Description |
|---|---|---|---|
| insertFlexibilityForecst | Local hub id | string | name/identification of Local Control Hub |
| | Device ids | List of string | name/identification of device |
| | Forecast profile | Forecast | Forecast profile for Local hub in terms |

| | | | |
|---|---|---|---|
| | | profile type | of flexibility |

#### 5.2.1.3 Impact of Prosumers Control & Optimization on CIM

The above described interfaces will be reflected in the Aggregator IAM, thus the Aggregator IAM will provide mirror interface serving Aggregator Prosumers Control & Optimization Component. The name of this Aggregator IAM interfaces will be part of CIM vocabulary specification.

| Nr | ProsCO interface | IAM interface |
|---|---|---|
| N/A | *insertFlexibilityForecst* | ProsCO_insertFlexibilityForecst |

The profile records specification need to be part of Aggregator CIM vocabulary (specified in Multi Agent Based Local Hub Gateway requirements on Local Hub CIM). The method will store these flexibility data relavant to Local hub. The methods getDRResponseStatistics and getDRRequestStatistics data type specified in INERTIA Aggregator Analytics will be filled by these records (flexibility forLocal Hub ID). The specification in the Aggregator CIM will be done using XSD schemas. The data of these types will be stored in the hybrid repository of IAM in form of DB tables. The LH and device identifier will refer to the instance of LH in the BIM semantic model and device identifier in the DER semantic model respectively.

### *5.2.2* Grid Coordinator& DR Activation Component requirements

#### 5.2.2.1 Data models describing Grid Coordinator& DR Activation

There are two types of information that are needed by the Grid Coordination and Demand Response Activation component.

The first one is related to the information received fork the DSO with the definition of the demand response requested to the aggregator. This information consists of a power reduction or increase both in demand or generation for a certain Medium Voltage supply point. This information is given as a power profile that may vary with time.

The DSO request is mainly defined by:

- Start time: the time when the demand or generation request starts.
- End time: the time when the demand or generation request ends.
- Time step: duration of each time step for which the demand or generation modification values are requested

- Request power values: For each time step in the request the power value to be increased/decreased
- MV supply point: the identifier of the MV supply point where modification in demand or generation is required
- Request type: The request type indicates if the modification values are for increasing or decreasing generation or demand

The following tables show with a greater detail level the data model for the DSO Demand Response requests.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| DSODispatch | Id_Dispatch | String | Identification of the profile |
| | Id_Issuer | String | Identification of the DSO control hub source |
| | Id_Receiver | String | Identification of the aggregator / local control hub destination |
| | Id_Type | String Enumeration | Type of profile:<br>- Dispatch Demand Up<br>- Dispatch Demand Down<br>- Dispatch Generation Up<br>- Dispatch Generation Down |
| | TimeStamp | Timestamp | Date & Time of the message. |
| | Energy_Profile | EnergyProfile | Energy profile |

| Type name | Attributes | Type | Description |
|---|---|---|---|
| EnergyProfile | Id_Supply | String | Identification of the supply point at MV level. |

| | | | |
|---|---|---|---|
| | TimeStamp_Begin | Timestamp | Date & Time at which the data set starts. |
| | TimeStamp_End | Timestamp | Date & Time at which the data set ends. |
| | TimeStamp_Step | Seconds | Time length between two individual steps. |
| | Energy_Values | Array of Float | Set of energy values one per step.<br><br>The value will always be a positive float value in the agreed measurement unit (kW-h, MW-h,…) and interpreted accordingly to the profile type. |

The aggregator needs to map flexibility information related to Local Control Hubs to flexibility information related to Medium Voltage supply points. This is because for network operation, the information needed is at Medium Voltage supply point where secondary transformers lower the voltage from MV to LV.

Each local control hub in the portfolio of the aggregator is connected to one Medium Voltage supply point in the electric distribution network. And one Medium Voltage supply point may have many Local Hubs connected to it.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| MV2LCHMapping | Id_Mapping | String | Identification of the mapping |
| | TimeStamp | Timestamp | Date & Time when the mapping was generated |
| | Mapping_Values | Array of MappingValue | Array containing the Local Control Hubs belonging to each MV supply point |

| Type name | Attributes | Type | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| MappingValue | Id_Supply | String | Identification of the supply point at MV level. |
| | Id_LocalHubs | Array of String | List of Local Control Hub IDs belonging to the MV supply point |

#### 5.2.2.2 Interfaces for Grid Coordinator& DR Activation

Storing of dispatch signal

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| setDSODispatch | DSO_Dispatch | DSODispatch | Information about the Demand Response dispatch request as received from the DSO Control Hub |

Read mapping between supplier and Local hubs

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getMV2LCHMapping | | | Obtains the latest mapping in the IAM<br><br>The return type is **MV2LCHMapping** |

#### 5.2.2.3 Impact on CIM

Both interfaces setDSODispatch and getMV2LCHMapping of this component will be reflected in the aggregator IAM, thus aggregator IAM will provide mirror interfaces serving Grid Coordinator& DR Activation Component. The names of these aggregator IAM interfaces will be part of CIM vocabulary specification.

The INERTIA specific data types (DSODispatch, EnergyProfile, MV2LCHMapping, and MappingValue) that are specified above will be part of CIM vocabulary. Their specification in the CIM will be done using XSD schemas. The data of these types will be stored in the hybrid repository of IAM in form of DB tables. The schema definition of that enable storing and retrieving such data will be reflected in the CIM vocabulary too. The LH identifier will refer to the instance of LH in the BIM semantic model.

### *5.2.3* Aggregator Analytics requirements

The role of the Aggregator Analytics Component is provided in the DoW as the DER Flexibility Analysis, Aggregation and Forecasting Module.

#### 5.2.3.1 Data requirements of Aggregator Analytics

As the role of the tool is to analyze historical related aspects, the total performance will be based on the overall data storage in the **Aggregator IAM**. The more the available data, the optimal data mining techniques for the extraction of useful knowledge is. Due to storage database limitations, we are going to handle the operational analysis in a hybrid form.

A categorization of the available analysis is provided based on the different operational states of the Aggregator:

- During a DR event (information will further handled during DR events). By the term of DR event we handle both the Demand Requests as managed by the D.S.O. and the Aggregator's Settings as managed through the UI of the Aggregator (Specific DR requests on the portfolio)
    - DR Event characteristics triggered by the D.S.O. (time-stamping reference). These data are stored in IAM by Grid Coordinator& DR Activation Component. Or DR requests as provided from the Aggregators UI.- stored in IAM by Grid Coordinator& DR Activation Component (thus requirements on data models are analysed and defined for this component).
    - Aggregator’s initial selection of Local Hubs portfolio Energy Consumption & Flexibility (time – series of aggregated amount of flexibility enrolled for the time period of the DR signal). These Data are stored in Aggregator IAM by MAS Prosumers Control & Optimization Component (thus requirements on data models are analysed and defined for this component).
    - Per Local Hub: Initial selection of **DERs and spaces (as handled by Occupancy prediction and flow component through the BIM Model)** Energy Consumption & Flexibility for the time period of the DR event. Thus the whole events parameters will be considered during the DR event time-period. (Demand Flexibility enrolled for each Local Hub for the time period of the DR signal) These data are provided by MAS Local Hub Gateway.
    - Aggregator’s final contribution of the respective LHs in the requested DR event (time – series of aggregated data Energy Consumption & Flexibility). These data will be stored in IAM by Prosumers Control &Optimization Component.
    - Local Hubs final contribution of DER’s Flexibility in the requested DR event (time – series of per DER related energy data (Energy Consumption& Flexibility)). These data are provided by MAS Local Hub Gateway

- A list of KPIs related to the Aggregators, handled in IAM either for the Aggregator's performance or the Local Hub performance. These data are provided by MAS Local Hub Gateway and Prosumers Control &Optimization Component.

**Visual analytics tool**

The Visual analytics tool is a data mining tool with the main functionality to analyze the historical data as stored in Aggregator's IAM.

A set of get methods is defined as part of the data model work of the Visual Analytics tool. The insert/update methods are defined by the respective methods of the above mentioned Components.

***The respective data needed {energy consumption and flexibility} will be handled at the time granularity permitted from the needs of storage {max: 1 minute, min: 4-5 seconds.}***

- For the market analysis or the multi-criteria analysis (parametric analysis). During this operation there is no need for the extraction of the continuous data as this may lead to storage limitations. The overall analysis will be mainly based on the market characteristics examined.
    - Aggregated data about energy consumption as defined from the functionality of the Prosumers Control & Optimization Component
    - Aggregated data about demand flexibility as defined from the functionality of the Prosumers Control & Optimization Component
    - Extrapolation of the energy consumption & demand flexibility data to the respective DERs of the portfolio.

The energy measurement data will be handled as part of this approach. The initially enrolled data are out of the scope of this analysis as we want to hide from the Aggregator's view, the internal analysis of each Local Hub.

This information will be provided as batches of data to be available in Aggregator IAM for a specific time period. {Defined by the limitations of data storage unit}. The aggregation of data will be delivered in time periods of 5/10/15 minutes (LH Consumption and measured flexibility- Extrapolation of these data to the respective DERs - These types of events are available in LH level and the aggregation of data is provided from the Gateway Component for each LH).

***If a storage limitation is arising, then we can handle this issue by throwing away or aggregating the historical data. This is an operation handling internally in IAM, based on the total data storage performance.***

For each Demand Response signal as provided to the Aggregator, at least the on-going list of data types should be delivered in INERTIA Aggregator IAM for further analysis. The role of the Control & Optimization Component, Local Hub Gateway is to handle the below mentioned information based on the functionality provided on sequence flows diagrams.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| DRResponseStatistics | LocalHub | ItemResult | The local hub that the statistics refer to |
| | EnergyConsumptionForecasted | MetricOnTime[] | The overall energy consumption as forecasted by the model. |
| | EnergyConsumptionActual | MetricOnTime[] | The overall energy consumption as measured. |
| | FlexibilityForecasted | MetricOnTime[] | Flexibility as forecasted |
| | FlexibilityActual | MetricOnTime[] | Flexibility as measured |
| | EnergyConsumptionForecastedPerSpace | Segment[] | List of spaces. Each segment contains as its members a list of MetricOnTime, each one containing EnergyConsumptionForecasted totals for the space. Each segment has value = spaceId, name = space description. |
| | EnergyConsumptionActualPerSpace | Segment[] | List of spaces. Each segment contains as its members a list of MetricOnTime, each one containing EnergyConsumptionActual totals for the space. Each segment has value = spaceId, name = space description. |
| | FlexibilityForecastedPerSpace | Segment[] | List of spaces. Each segment contains as its members a list of MetricOnTime, each one containing FlexibilityFoprecasted totals for the space. Each segment has value = spaceId, |

| | | | |
|---|---|---|---|
| | | | name = space description. |
| | FlexibilityActualPerSpace | Segment[] | List of spaces. Each segment contains as its members a list of MetricOnTime, each one containing FlexibilityActual totals for the space. Each segment has value = spaceId, name = space description. |
| | EnergyConsumptionForecastedPerDER | Segment[] | List of DERs. Each segment contains as its members a list of MetricOnTime, each one containing EnergyConsumptionForecasted for the DER. Each segment has value = DERId, name = DER description. |
| | EnergyConsumptionActualPerDER | Segment[] | List of DERs. Each segment contains as its members a list of MetricOnTime, each one containing EnergyConsumptionActual for the DER. Each segment has value = DERId, name = DER description. |
| | FlexibilityForecastedPerDER | Segment[] | List of DERs. Each segment contains as its members a list of MetricOnTime, each one containing FlexibilityFoprecasted for the DER. Each segment has value = DERId, name = DER description. |
| | FlexibilityActualPerDER | Segment[] | List of DERs. Each segment contains as its members a list of MetricOnTime, each |

| | | | |
|---|---|---|---|
| | | | one containing FlexibilityActual for the DER. Each segment has value = DERId, name = DER description. |

The additional data types are provided in order to further describe the complex structure as mentioned:

Metric on Time is used to represent time series of values for requested metrics, e.g. consumption per day for last week.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| MetricOnTime | Value | float | The value of metric for a specific point in time |
| | Date | dateTime | The point in time for which a value was estimated |

Segment is used to represent slices of data for segmentation metrics, e.g. showing DERs with high, low and average consumption for last week.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| Segment | Name | string | Name of the segment |
| | Value | float | Value of the segment |
| | Members | ItemDetail | List of items belonging to the segment |

A signal data-type is considered for the analysis of different types of signals to be handled within the INERTIA project.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| SignalType | SignalTypeId | ID | The ID of the signal type. |
| | Description | String | Textual description. |

| | | | |
|---|---|---|---|
| | Parameters | Array of ItemDetail | List of parameters for signal type |

An indicative list of data signals are provided, based on the above mentioned description of the data type.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| SignalTypeID | ID1 | Duration, Energy Profile | Grid - Dispatch Demand Up |
| | ID2 | Duration, Energy Profile | Grid - Dispatch Demand Down |
| | ID3 | Duration, Energy Profile | Grid - Dispatch Generation Up |
| | ID4 | Duration, Energy Profile | Grid - Dispatch Generation Down |
| | ID5 | | Market- Ancillary services market |

As mentioned, different types of parameters need to be handled and this analysis is based on the ID of the signal type. Thus, and as is not clear the parameter settings a generic type of "Array of ItemDetail" is mentioned.

High-Level Strategy of Aggregator Control Hub is described as following.

| Type name | Attributes | Type | Description |
|---|---|---|---|
| HighLevelStrategy | ID | string | Identifier of High-Level Strategy of Aggregator control Hub |
| | name | String | Name of High-Level Strategy |
| | description | string | Description of High-Level Strategy |

#### 5.2.3.2 Interfaces related Visual analytics tool

The Visual analytics tool needs (at least) the above mentioned dataset for a concrete analysis of the historical data. The whole analysis is delivered, taking into account the additional operational modes:

- During a DR event, data management
- During non DR event, data management

**During a DR event**

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getAllDREvents<br><br>ForAggregator | AggregatorID | String | |
| | From | dateTime | |
| | to | dateTime | |
| return | drEventsForAggregator | Array of DSODispatch | |

Note for type DSODispatch: please add description of type for display purpose

| Method name<br><br>(import/export) | Attributes | Type | Description |
|---|---|---|---|
| getAllDRStatistics<br><br>ForAggregator | DSODispatchID | String | |
| return | drStatistics<br><br>ForAggregator | Array of DRResponseStatistics | |

Apart from the DSO requests as provided from the System Operator, a list of DR settings are delivered from the Aggregator, through the Visual Interface of the DSS tool. These detailed settings are further stored in IAM for further evaluation.

| Method name | Attributes | Type | Description |
|---|---|---|---|

| (import/export) | | | |
|---|---|---|---|
| getStatisticsForAllDRRequestsFromAggregator | AggregatorID | String | |
| | From | dateTime | |
| | to | dateTime | |
| return | drStatisticsForRequestsForAggregatro | Array of DRRequests | |

Note for type DREquest: This type is part of the work performed in T4.2 - Grid Coordination & DR Activation Module, and should therefore be included in D2.4.2 INERTIA Multi-dimensional DER flexibility models. It is a model of what a DR signal includes. Here, It is only used as the return type for method *getStatisticsForAllDRRequestsFromAggregator.*

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getStatisticsForDRRequest | DRRequestID | String | |
| return | statisticsForDRRequests | Array of DRResponseStatistics | |

Apart from the detailed aspects (energy consumption and flexibility during the DR period) a list of KPIs will be also handled and further stored in INERTIA IAM (Handled and stored by Control & Optimization Component, Local Hub). These KPIs will be further retrieved from the Visual Analytics tool for further exploitation. These KPIs may refer to a specific Aggregator or specific signal as examined.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getKPIsForAggregator | AggregatorID | String | |
| | From | dateTime | |
| | to | dateTime | |
| return | KPIsForAggregator | Array of ItemDetail | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getKPIsForDRSignal | SignalID | String | |
| return | KPIForDRsignal | Array of ItemDetail | |

As the specific datatype retrieved is not clear, an ItemDetail view is provided.

Item details are used to represent item details to be retrieved/shown when drilling down.

**During Non DR event Operation**

As already mentioned, the need to store the portfolio related aspects during DR operation does not cover the full functionality of the VA tool. There is a need to analyse the whole operation of the portfolio, and thus there is a need to store the energy related data during continuous operation. The aggregation of data types in this case scenario is considered (e.g. every 15 minutes) for storage issues. In the following method, where the data stored in Aggregator IAM are retrieved from the Visual Analytics tool, a time step attribute is provided, which will be further defined by the functionality of the VA tool.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getDRStatisticsPerTimestep | AggregatorID | String | |
| | timeStep | Long (translated to number of seconds) | |
| | From | dateTime | |
| | to | dateTime | |
| return | responseStats | Array of DRResponseStatistics | |

Note: for DRResponseStatistics returned by this method, fields referring to space statistics are NOT filled in

Up until now, we have mentioned the need to handle different types of information in order to extract useful statistics about the operation of the portfolio, either during the DR signals operation or during the normal portfolio conditions.

Thus, we have not covered the extra functionality of the VA tool. The scope of the VA is to operate different what if scenarios (market based or grid based scenarios with different parameter settings) and then to store the results from the above mentioned analysis in IAM. The scope of this functionality is to store detailed analytics in order to further handle them during the real time operation. As mentioned at the sequence flow analysis, there is a need for a quick decision (through the VA tool) analysis during DR signals operation. Thus, and in order to have a quick response during real time operation, there is a need to handle additional types of what if scenario events and to retrieve them when needed.

Apart from the get methods as already provided, some extra data, reflecting the analysis from the Visual Analytics tool are handled through Aggregator's IAM. The Business related aspects are not examined as part of the project and thus a what-if market related scenarios will be handled through the Visual Analytics tool. In addition some, grid related what if scenarios will be also examined as part of the functionality of the Visual Analytics tool. Thus, the results from this analysis will be further stored in IAM. A list of KPIs parameters (objective function of the market operation), the list of Local Hubs-DERs that will further contribute on the what-if scenario (Id of LH/DER, Flexibility provided, Energy Consumption data) are stored for further exploitation

A method to store the list of LHs (only the IDs) based on the signal type and the specific parameters as examined is provided.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| storeCandidatesForDRSignal Response | localHubList | Array of ItemResult | |
| | signalTypeId | String | |
| | parameterList | Array of ItemDetail | |
| return | operationSuccess | boolean | |

The reverse method is to get the list of the Candidate Local Hubs For the specific DR Signal examined (either actual DR signal or the "what if scenarios" as examined). The detailed data types to be retrieved from the INERTIA IAM are provided. As the list of parameters to be examined is not binding please keep in mind that the fields of (duration, energyLowerBound, energyUpperBound) may differentiate in the final CIM format.

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| getCandidateLocalHubsForDRSig nalResponse | SignalTypeID | String | |

| | | | |
|---|---|---|---|
| | duration | Long (corresponds to seconds) | |
| | energyLowerBound | Float | |
| | energyUpperBound | Float | |
| return | candidates | Array of ItemResult | |

Signature of Interfaces for high level strategies

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| storeHighLevelStrategy | highevelStrategy | Strategy | High-Level Strategy of Aggregator control Hub |
| Return | operationSuccess | boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| updateHighLevelStrategy | highevelStrategy | Strategy | High-Level Strategy of Aggregator control Hub |
| Return | operationSuccess | boolean | |

| Method name (import/export) | Attributes | Type | Description |
|---|---|---|---|
| readHighLevelStrategy | ID | string | Identifier of High-Level Strategy of Aggregator control Hub |

| | | | |
|---|---|---|---|
| | strategy | highevelStrategy | |

The above mentioned analysis of the Visual Analytics related methods are not finalized. A set of additional methods related to the internal operation of the VA tool will be handled {*get.KPIs_related_to_VA_analytics, store.KPIs_related_to_VA_analytics* }.

#### 5.2.3.3 Impact of INERTIA Aggregator Analytics on CIM

The reviewed list of Visual Analytics related methods are provided in the following table. At a first point, the input VA methods are defined in order to get access on the information stored in Aggregator IAM. Also type DRResponseStatistics need to be incorporated into CIM XSD schemas, since they provide the required view to the flexibility data produced and stored by Prosumer Control & Optimalisation component in IAM. Note, DRRequest type is part of the work performed in T4.2 - Grid Coordination & DR Activation Module, and will therefore be included in D2.4.2 INERTIA Multi-dimensional DER flexibility models.

***Table 17: Mapping between Visual Analytics interfaces (retrieval) and IAM interfaces***

| Nr | VA interface | IAM interface |
|---|---|---|
| 1 | *getAllDREventsForAggregator* | VA_getAllDREventsForAggregator |
| 2 | *getAllDRStatisticsForAggregator* | VA_getAllDRStatisticsForAggregator |
| 3 | *getStatisticsForAllDRRequestsFromAggregator* | VA_getStatisticsForAllDRRequestsFromAggregator |
| 4 | *getStatisticsForDRRequest* | VA_getStatisticsForDRRequest |
| 5 | *getKPIsForAggregator* | VA_getKPIsForAggregator |
| 6 | *getKPIsForDRSignal* | VA_getKPIsForDRSignal |
| 7 | *getDRStatisticsPerTimestep* | VA_getDRStatisticsPerTimestep |
| 8 | *getCandidateLocalHubsForDRSignalResponse* | VA_getCandidateLocalHubsForDRSignalResponse |
| 9 | *getHighLevelStrategy* | VA_getHighLevelStrategy |

The output VA methods are provided for the storage of VA related Information to IAM.

***Table 18: Mapping between Visual Analytics interfaces (storage) and IAM interfaces***

| Nr | VA interface | IAM interface |
|---|---|---|
| 1 | *storeCandidatesForDRSignalResponse* | VA_storeCandidatesForDRSignalResponse |
| 2 | *storeHighLevelStrategy* | VA_storeHighLevelStrategy |
| 3 | *updateHighLevelStrategy* | VA_updateHighLevelStrategy |

# 6. Inertia CIM specification

Based on the analysis from chapter 4 the elements of CIM are defined here. These elements are defined for all INERTIA components that are interacting with IAM. The definition of CIM for each of such component consists of IAM interface name, XSD schemas defining data types used in these interfaces, DB schemas for storing the incoming data and specification of annotation of DB records to semantic models. Thus the structure of CIM description for each INERTIA component looks as following:

- IAM interfaces
- XSD schemas
- DB schemas
- Related semantic models

The IAM interfaces in many cases mirror the components interfaces. In these cases the names of IAM (mirroring) interfaces are provided, since the signature is analogous to the signatures of component interfaces. In the cases where specific IAM interfaces are defined the signatures are provided.

The full technical documentation to XSD schemas can be found in the Annex II.

Note, the semantic models are fully defined in the report D3.2, while initial semantic models can be also found in the report D3.1.

Note that besides the CIM for each INERTIA component interacting with the IAM, there is also proposal of conceptualisation for BIM model and DER model as they are static models that will bind data from these components together, will support the aggregation process as well as enable to enrich the dynamic data stored in IAM with context data needed.

Similarly to the previous section about analysis of requirements to CIM, this section (CIM definition) is divided into two parts.

- Local control Hub CIM
- Aggregator Control Hub CIM

These two CIMs are defined for Local Hub IAM and Aggregator Control Hub IAM respectively.

## *6.1 Local control Hub CIM*

### *6.1.1* LinkSmart middleware in CIM

**Local Hub IAM interfaces**

Exporting interfaces

- getLocationModel
- getDeviceLocationModel
- getZoneDERs
- getDERModel

Importing Interface

- setLinkSmartEvent

**XSD schemas**

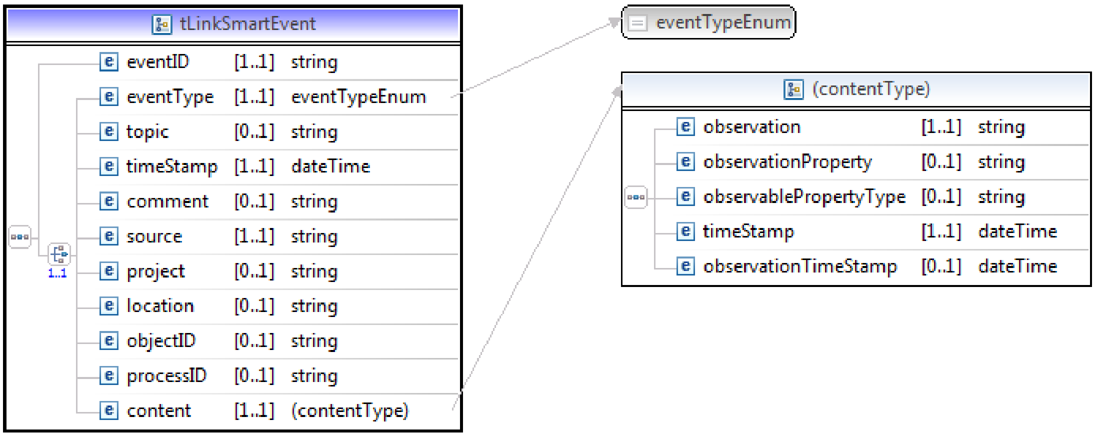



***Figure 6: LinkSmart event schema***

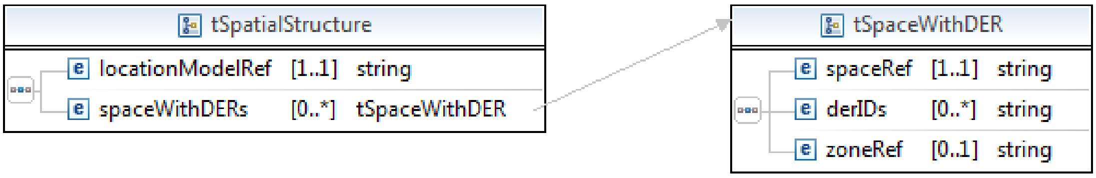



***Figure 7: Spatial structure schema for LinkSmart***

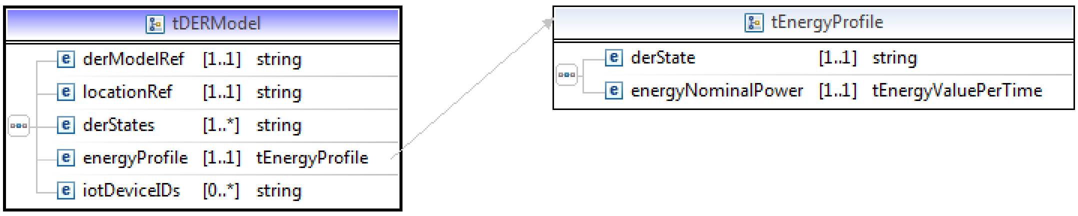



***Figure 8: DER model schema for LinkSmart***

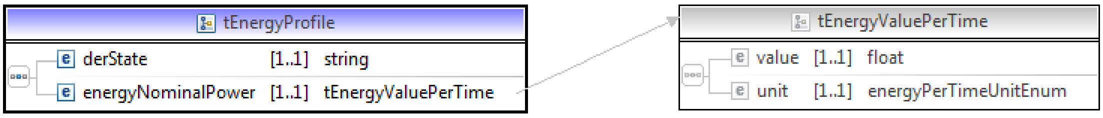



***Figure 9: Energy profile schema for LinkSmart***

## DB tables

DB table schemas for storing LinkSmart events:

SQL specifications:

```
CREATE TABLE ls_event
(
    entryID int NOT NULL PRIMARY KEY,
    eventID (255) NOT NULL,
    eventType (255)
    occupantID (255)
    groupID (255)
    topic (255)
    dateTimeStamp datetime NOT NULL DEFAULT GETDATE(),
    comment (255)
    source (255) NOT NULL,
    project (255)
    location (255)
    objectID (255)
    processID (255)
);
CREATE TABLE ls_event_content
(
    entryID int NOT NULL PRIMARY KEY,
    lsEventID int NOT NULL FOREIGN KEY,
    observation (255) NOT NULL,
```

```
        observationProperty (255),
        observablePropertyType (255),
        dateTimeStamp datetime NOT NULL DEFAULT GETDATE(),
        observationDTStamp datetime
);
```

**Related semantic models**

The DB table fields Location, ObjectID from DB schema for storing events contain the annotation to BIM semantic model and DER semantic model respectively.

The values of attributes in case of spatial structure XSD schema and DER model XSD schema will be obtained by querying BIM and DER semantic models. The proposal conceptualisation of these semantic models is proposed in this report, while prototype of these semantic models can be found in the INERTIA report D3.1 while and the full specification will be defined in the INERTIA report D3.2.

### *6.1.2* Occupancy prediction and flow in CIM

**Local Hub IAM interfaces**

CIM vocabulary for OCC components contains following names of services designed for OCC requests:

IAM exporting interfaces (serving for OCC imports)

- IAM.OCC_importHistoricalOccupancy
- IAM.OCC_importHistoricalOccupantLocation
- IAM.OCC_importRealTimeOccupancy
- IAM.OCC_importRealTimeOccupantLocation
- IAM.OCC_importSensorLocation
- IAM.OCC_importSpatialSchedule
- IAM.OCC_importOccupantSchedule
- IAM.OCC_importOpenReferenceModels
- IAM.OCC_importOverallInertiaOccupancyAndFlowModel
- IAM.OCC_importOverallInertiaOccupancyAndFlowModel

IAM importing interfaces (serving for OCC exports)

- IAM.OCC_exportRealTimeOccupancyPerSpace
- IAM.OCC_exportRealTimeOccupantLocation
- IAM.OCC_exportOverallInertiaOccupancyAndFlowModel
- IAM.OCC_exportIndividuallInertiaOccupancyAndFlowModel
- IAM.OCC_exportSpatialOccupancyAndFlowPrediction
- IAM.OCC_exportOccupantLocationAndFlowPrediction
- Boolean IAM.setLocalHubandBIMmodel(String ID, LocalHubandBIMmodel lh_bim)

Note, that other INERTIA component will use IAM exporting services providing real time occupancy data (i.e. IAM.OCC_importRealTimeOccupancy, IAM.OCC_importRealTimeOccupantLocation) that were calculated and exported by OCC component.

**XSD schemas**

XSD schemas corresponding to overall occupancy and Flow models (defined in D2.2).

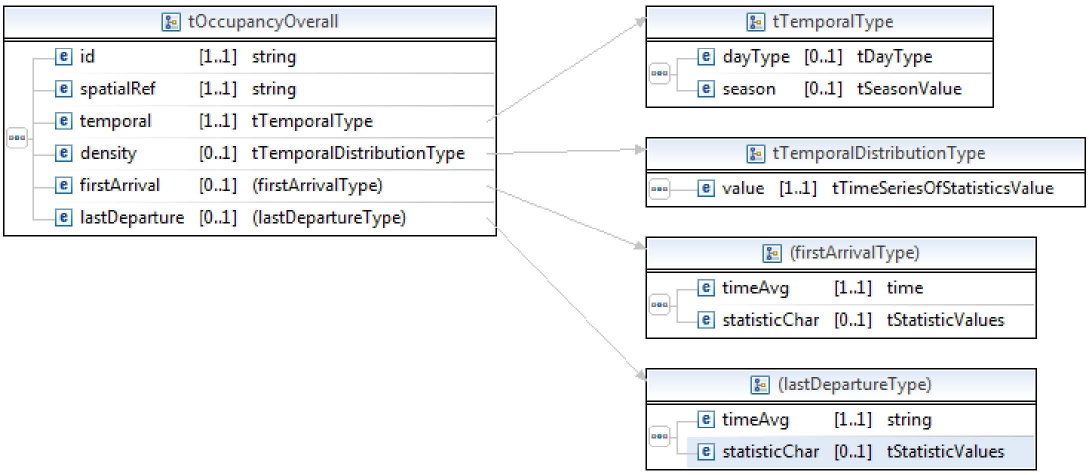



***Figure 10: Schema of INERTIA Occupancy Model Overall***

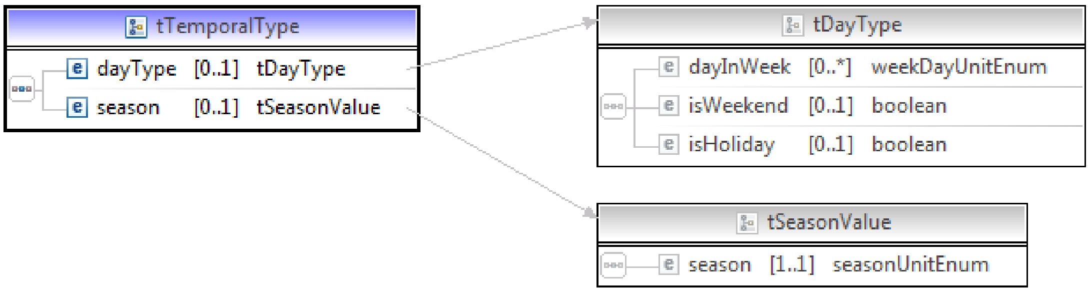



***Figure 11: Temporal type schema with day type and season elements***

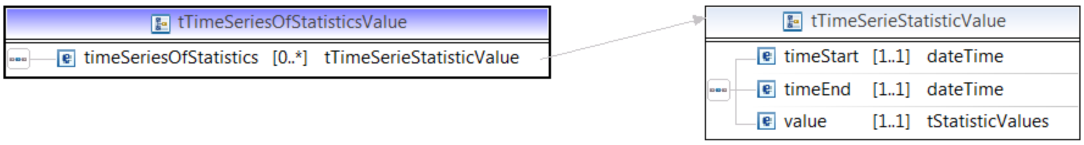



***Figure 12: Time series of statistics schema***

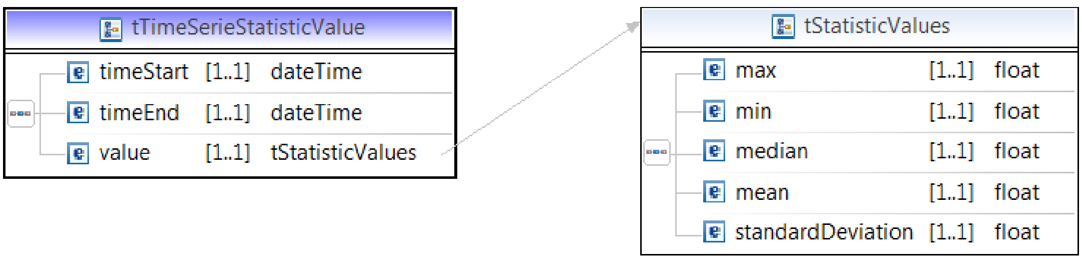



***Figure 13: Statistics schema***

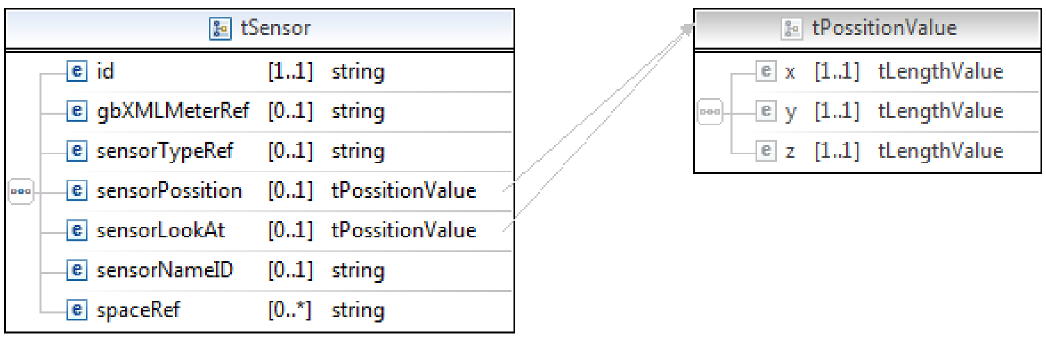



***Figure 14: Sensor location XSD schema***

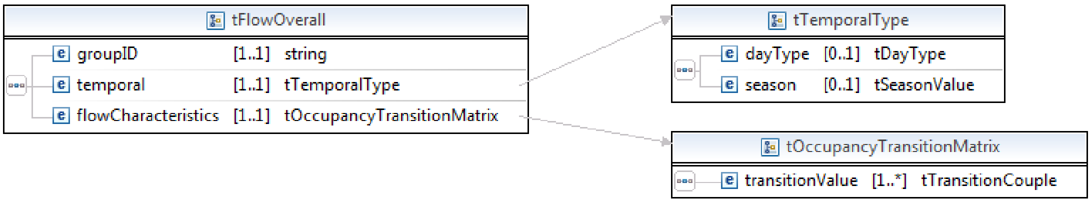



***Figure 15: Overall flow type schema***

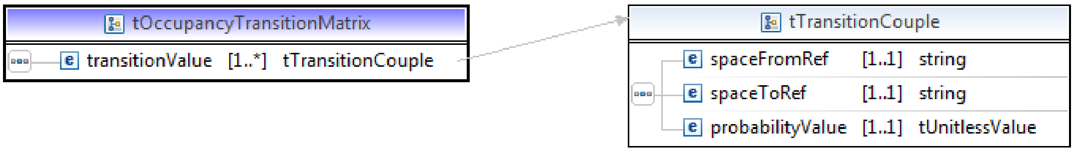



***Figure 16: Occupancy transition matrix schema***

XSD schemas corresponding to individual occupancy and Flow models (defined in D2.2).

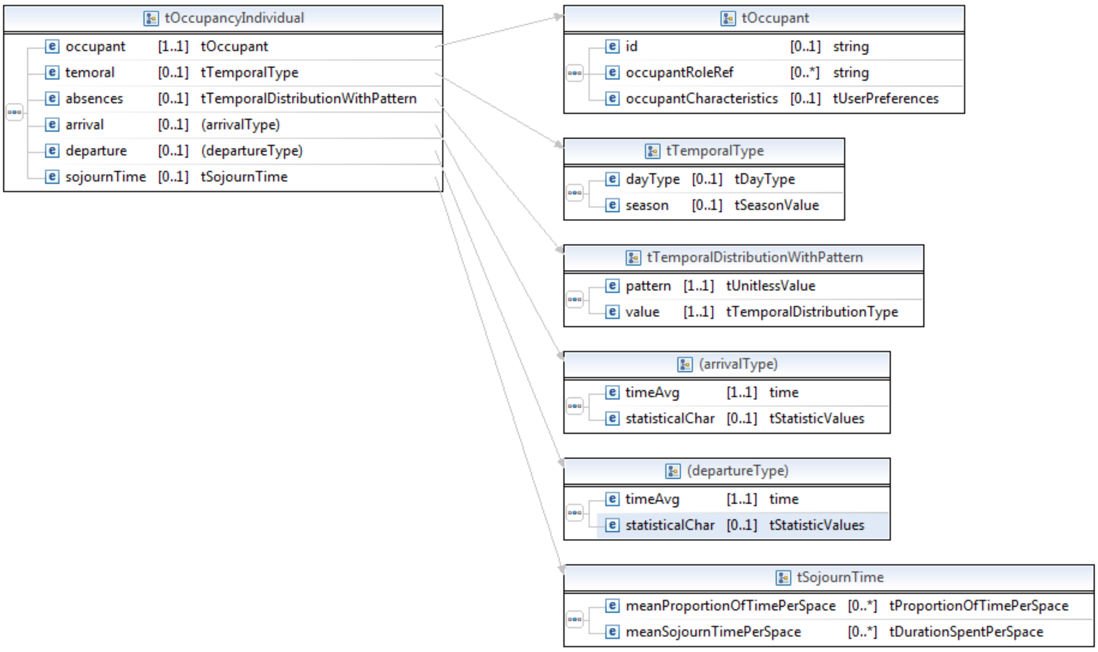



***Figure 17: Schema of INERTIA individual occupancy model connecting this model to User Preferences model defined in INERTIA User Interface module***

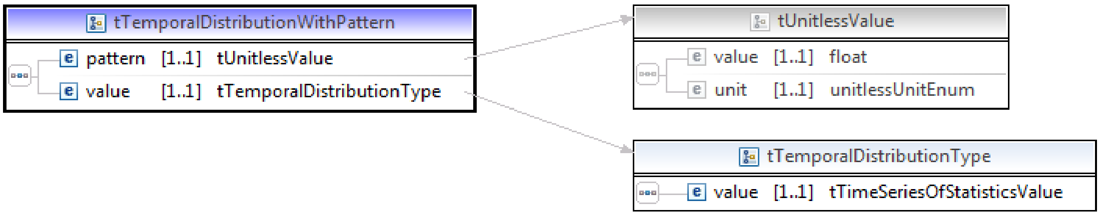



***Figure 18: Temporal distribution with pattern schema***

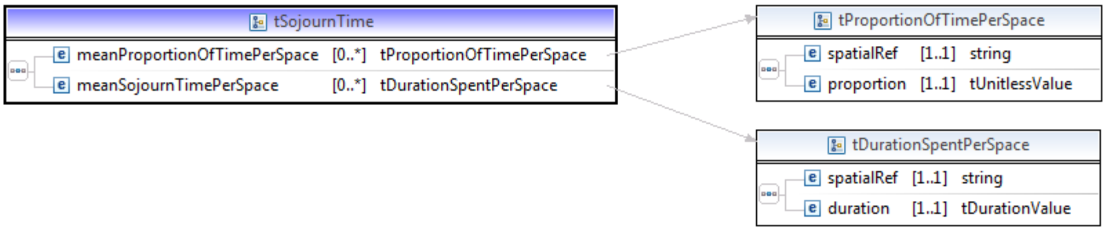



***Figure 19: Sojourn type schema***

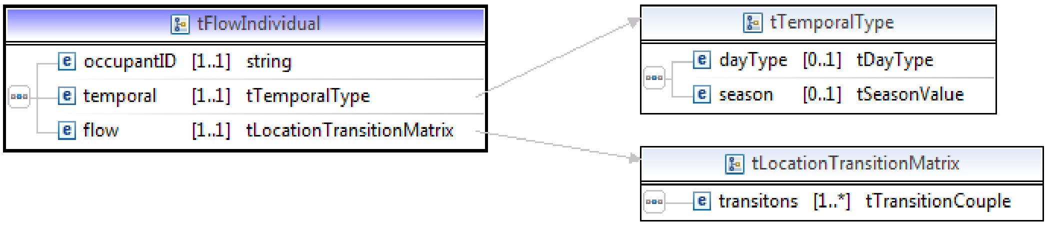



***Figure 20: Individual flow model schema***

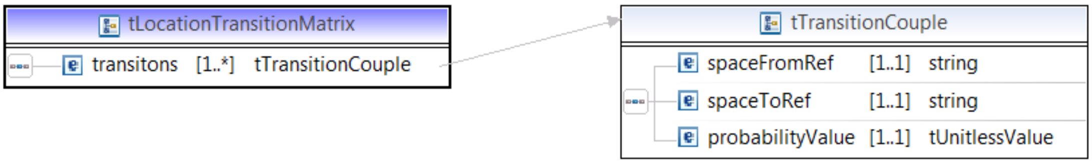



***Figure 21: Location transition matrix schema***

XSD schemas corresponding Local Hub model and BIM model hierarchy and its binding with model of IoT sensors.

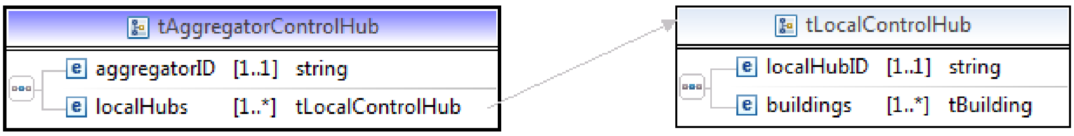



***Figure 22: Aggregator control hub schema containing local control hubs***

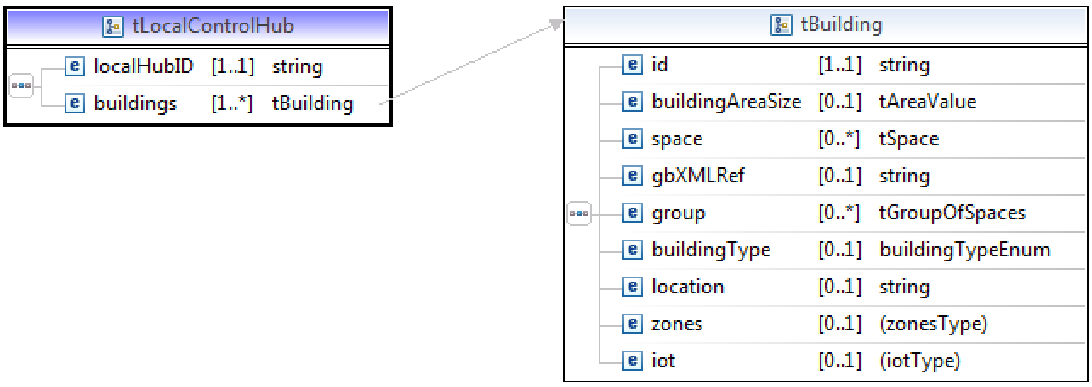



***Figure 23: Local control hub schema containing buildings***

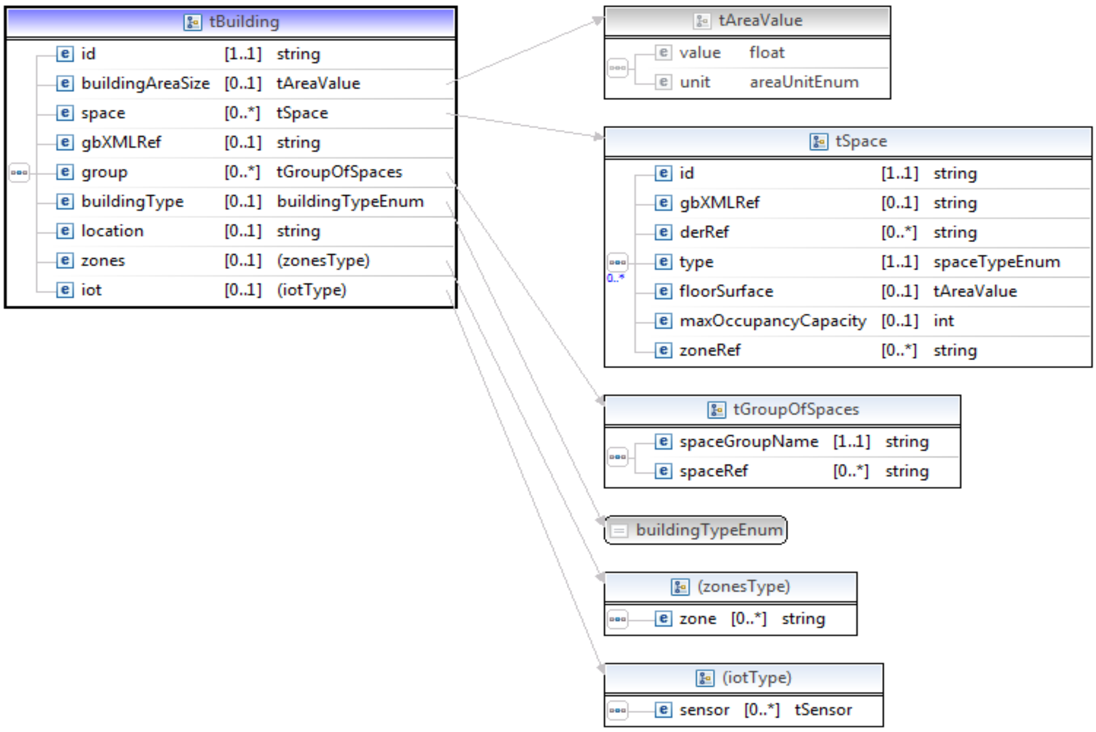



***Figure 24: Building schema containing spaces, zones, sensors and reference to gbXML model***

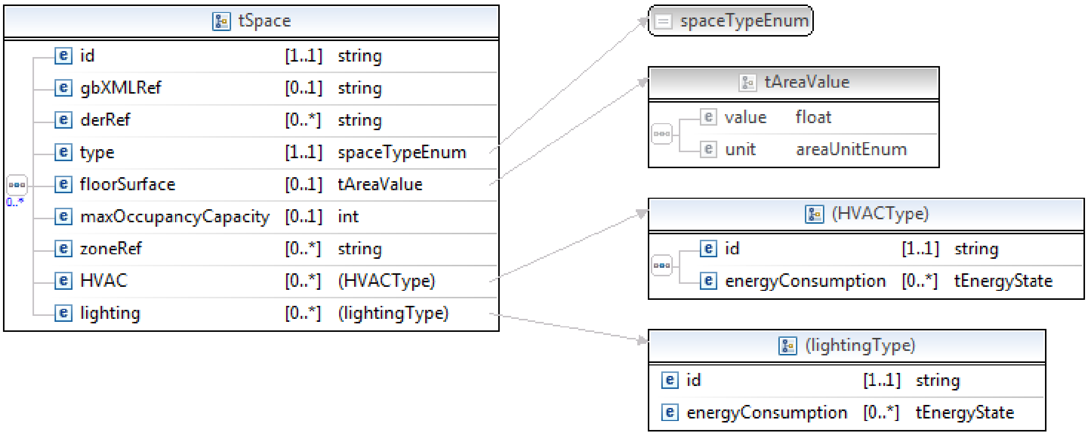



***Figure 25: Space schema containing energy consumption characteristics, occupancy characteristics and reference to gbXML model***

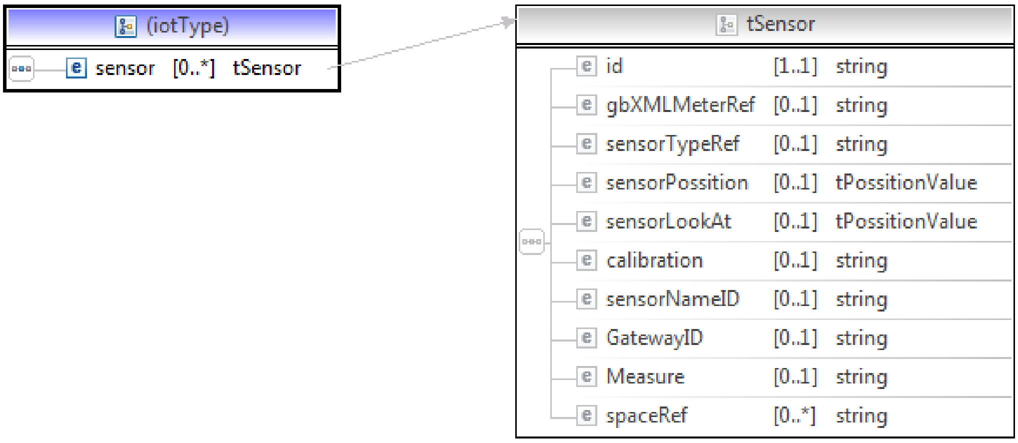



***Figure 26: IoT schema containing sensor characteristics and reference to gbXML model***

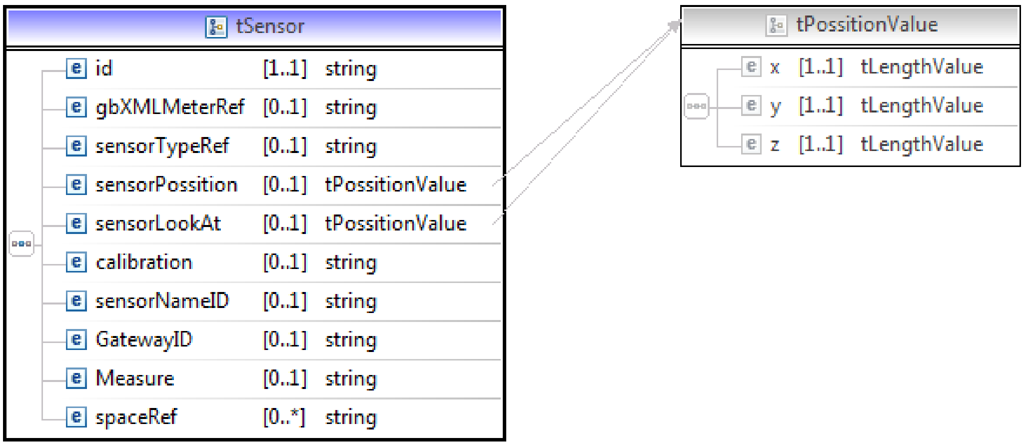



***Figure 27: Sensor schema containing its position and orientation in space***

**DB tables**

The importing IAM interfaces will be used to store data from OCC component in the DB tables. These DB tables are also used when retrieving data for OCC component and when retrieving occupancy data for the other INERTIA component.

The definition of the DB tables used by IAM to store OCC data is below.

OCC_exportRealTimeOccupancyPerSpace

DB_table

SQL specification:

```
CREATE TABLE rtocc_number

(

        entryID int NOT NULL PRIMARY KEY,

        dateTimeStamp datetime NOT NULL DEFAULT GETDATE()

        spatialID (255) NOT NULL,

        numberOfOccupants int NOT NULL

);
```

OCC_exportRealTimeOccupantLocation

DB_table

SQL specification:

```
CREATE TABLE rtocc_location

(

        entryID int NOT NULL PRIMARY KEY,

        dateTimeStamp datetime NOT NULL DEFAULT GETDATE()

        occupantID (255) NOT NULL,

        spatialID (255) NOT NULL

);
```

OCC_exportOverallInertiaOccupancyAndFlowModel

DB_table

DB table schemas for overall occupancy model:

SQL specification:

```
CREATE TABLE overall_occ_model
(
	entryID int NOT NULL PRIMARY KEY,
	dateTimeStamp datetime NOT NULL DEFAULT GETDATE()
	LCH_ID (255) NOT NULL,
	spatialID (255) NOT NULL,
);
CREATE TABLE overall_occ_temporal
(
	entryID int NOT NULL PRIMARY KEY,
	overalloccID int NOT NULL FOREIGN KEY,
	season (255),
	dayType (255),
	firstArrival time,
	lastDeparture time
);
CREATE TABLE overall_occ_density
(
	entryID int NOT NULL PRIMARY KEY,
	overalloccID int NOT NULL FOREIGN KEY,
	timeStart datetime NOT NULL,
	timeEnd datetime NOT NULL,
	min float,
	max float,
	median float,
	mean float,
	standardDeviation float
);
```

DB table schemas for overall Flow model:

SQL specifications:

```
CREATE TABLE overall_flow_model
(
        entryID int NOT NULL PRIMARY KEY,
        groupID (255) NOT NULL,
        dateTimeStamp datetime NOT NULL DEFAULT GETDATE()
        season (255),
        dayType (255),
);
CREATE TABLE overall_flow_tmatrix
(
        entryID int NOT NULL PRIMARY KEY,
        oFlowModelID int NOT NULL FREIGN KEY,
        spaceFromID (255) NOT NULL,
        spaceToID (255) NOT NULL,
        probability float NOT NULL,
);
```

OCC_exportIndividuallInertiaOccupancyAndFlowModel

DB_table

SQL specification:

```
CREATE TABLE indiv_occ_model
(
        entryID int NOT NULL PRIMARY KEY,
        occupantID (255) NOT NULL,
        season (255),
```

```
    dayType (255),
    firstArrival time,
    lastDeparture time,
    absencesPattern (255),
    absencesPatternUnit (255)
);
CREATE TABLE indivocc_preferences
(
    preferenceEntryID int NOT NULL PRIMARY KEY,
    iOccupantModelID int NOT NULL FOREIGN KEY,
);
```

The DB table schemas for visual, thermal and operational preferences are defined in the section about Ambient User Interface component.

```
CREATE TABLE indiv_occ_roles
(
    entryID int NOT NULL PRIMARY KEY,
    iOccModelID int NOT NULL FOREIGN KEY,
    occupantRole (255) NOT NULL,
);
CREATE TABLE indivocc_absences_prob
(
    entryID int NOT NULL PRIMARY KEY,
    iOccModelID int NOT NULL FOREIGN KEY,
    min float,
    max float,
    median float,
    mean float,
```

```
    standardDeviation float
);
```

Sojourn:

```
CREATE TABLE indivocc_sojourn_timespace
(
    entryID int NOT NULL PRIMARY KEY,
    iOccModelID int NOT NULL FOREIGN KEY,
    spaceRef (255),
    proportion float,
);
CREATE TABLE indivocc_sojourn_durtaionspace
(
    entryID int NOT NULL PRIMARY KEY,
    iOccModelID int NOT NULL FOREIGN KEY,
    spaceRef (255),
    duration float,
    timeUnit (255)
);
```

DB table schemas for individual Flow model:

SQL specifications:

```
CREATE TABLE individual_flow_model
(
    entryID int NOT NULL PRIMARY KEY,
    occupantID (255) NOT NULL,
    dateTimeStamp datetime NOT NULL DEFAULT GETDATE()
    season (255),
    dayType (255),
```

```
);

CREATE TABLE individual_flow_tmatrix

(

        entryID int NOT NULL PRIMARY KEY,

        oFlowModelID int NOT NULL FREIGN KEY,

        spaceFromID (255) NOT NULL,

        spaceToID (255) NOT NULL,

        probability float NOT NULL

);
```

OCC_exportSpatialOccupancyAndFlowPrediction

DB_table

SQL specification:

```
CREATE TABLE occ_prediction_model

(

        entryID int NOT NULL PRIMARY KEY,

        spatialID (255) NOT NULL

        timeStart datetime,

        timeEnd datetime,

        min float,

        max float,

        median float,

        mean float,

        standardDeviation float

);

CREATE TABLE flow_data_prediction
```

```
(
        entryID int NOT NULL PRIMARY KEY,
        dateTimeStamp datetime NOT NULL,
        spatialID (255) NOT NULL
);
```

OCC_exportOccupantLocationAndFlowPrediction

DB_table

SQL specification:

```
CREATE TABLE occ_flow_model
(
        entryID int NOT NULL PRIMARY KEY,
        predictionDate datetime NOT NULL
        timeFrom datetime NOT NULL
        timeTo datetime NOT NULL
        timeStep int NOT NULL
        spatialID (255) NOT NULL
);
CREATE TABLE occ_transition_probalities
(
        entryID int NOT NULL PRIMARY KEY,
        modelID int NOT NULL FREIGN KEY
        spatialIDFrom (255) NOT NULL
        spatialIDTo (255) NOT NULL
        trasition float NOT NULL
);
```

**Related semantic models**

The data entries stored in these tables contain semantic annotation to the ontology model. Concretely, occupantID, roleID (or grouID), spatialID, LCH_ID are semantic annotations to the instances of Occupancy ontology (specified in D3.2) and BIM ontology.

#### 6.1.2.1 BIM model in INERTIA CIM

The location/building data objects are considered as static data (no updates, or rare updates are expected). The following list of concepts needs to be modelled in the IAM ontology (based on the analysis of requirements in section 4):

- Location, LocalHub, Facility, Building, Area, Sub-Area, Zone, Space

Following semantic relations enable to bind these concepts:

- hasSubSpace (transitive relation), hasZone, hasSubSpace (transitive relation), hasSubLocation

The requests from INERTIA components will specify location elements by URIs (IDs). These URIs will refer to the instances of Location instances of the location/building part of ontology.

**BIM initial conceptualisation**

Based on the analysis of requirements from OCC component developers, the basic structure of BIM ontology (putting together identified concepts and relations) was created as following:

Aggregator Hub – semantic relation hasSubLocation:

- **Location hasSubLocation LocalHub/Facility**
  **Example: cityOfThesaloniki hasSubLocation premisesCERTH**

Local Control Hub – semantic relations hasSubSpace, hasZone:

- **Higher level Location/Building concepts:** Location hasSubLocation LocalHub/Facility hasSubSpace Building/Area
  **Example:** premisesCERTH hasSubSpace researchAndTechnologyBuilding
- **Mid-level Location/Building concepts:** Building/Area hasSubSpace Sub-Area
  **Example:** researchAndTechnologyBuilding hasSubSpace 1stFloor
- **Low-level Location/Building concepts:** Sub-Area hasSubSpace Space

  **Example:** 1stFloor hasSubSpace meetingOffice1
  **Low-level Location/Building concepts:** Space hasZone Zone
  **Example:** meetingOffice1 hasZone thermalZone1

Semantic relation partOf is the inverse relation to the relation hasSubSpace and hasSubLocation. It enables to traverse in the ontology form the bottom (e.g. to get from the meetingOffice1 to the premisesCERTH). The relation partOf is complementary relation to the relation hasZone.

**Definitions:**

- Space partOf Zone
- Space partOf Sub-Area partOf Building (or Sub-Area partOf Area); Sub-Area partOf Local Hub; Building partOf LocalHub

**Examples:**

- meetingOffice1 partOf thermalZone 1

- meetingOffice1 partOf 1stFloor partOf researchAndTechnologyBuilding partOf premisesCERTH partOf cityOfThessaloniki

Note this is preliminary proposal of conceptualisation. The full specification of the BIM ontology will be provided in D3.2. The prototype specification can be found in D3.1.

### *6.1.3* User profiling in CIM

**Local Hub IAM interfaces**

Based on the requirements from User Profile Interface component the following interface will be defined in IAM, thus they are part of CIM vocabulary:

| |
|---|
| UP_getOccupantEventsOrStreotypes |
| UP_correlateEventToOccupantOrStreotypes |
| UP_assignOccupantWeightToEventOrStreotypes |
| UP_getOccupantWeightsForEventOrStreotypes |
| UP_updateOccupantOrStereotypeMinLuminance |
| UP_updateOccupantOrStereotypeMaxLuminance |
| UP_updateOccupantOrStereotypeCurrentLuminance |
| UP_updateOccupantOrStereotypeAverageLuminance |
| UP_updateOccupantOrStereotypeMinPMV |
| UP_updateOccupantOrStereotypeMaxPMV |
| UP_updateOccupantOrStereotypeCurrentPMV |
| UP_updateOccupantOrStereotypeAveragePMV |
| UP_updateOccupantEV |
| UP_updateOccupantOrStereotypeWaterHeater |
| UP_updateOccupantOrStereotypeRefrigeratorSetpoint |
| UP_updateOccupantOrStereotypeDishwasher |
| visualPreferencesType UP_getOccupantOrStereotypeLuminance(<br><br>String occupantID |

| |
|---|
| ) |
| thermalPreferences UP_getOccupantOrStereotypePMV(<br><br>String occupantID) |
| UP_getGroupEvents |
| UP_correlateEventToGroup |
| UP_getGroupWeightsForEvent |
| UP_updateGroupMinLuminance |
| UP_updateGroupMaxLuminance |
| UP_updateGroupCurrentLuminance |
| UP_updateGroupAverageLuminance |
| UP_updateGroupMinPMV |
| UP_updateGroupMaxPMV |
| UP_updateGroupCurrentPMV |
| UP_updateGroupAveragePMV |
| thermalPreferences UP_getGroupPMV(String occupantID) |
| visualPreferencesType UP_getGroupLuminance(String occupantID) |

**XSD specification:**

The following INERTIA data types are used in the above defined interfaces:

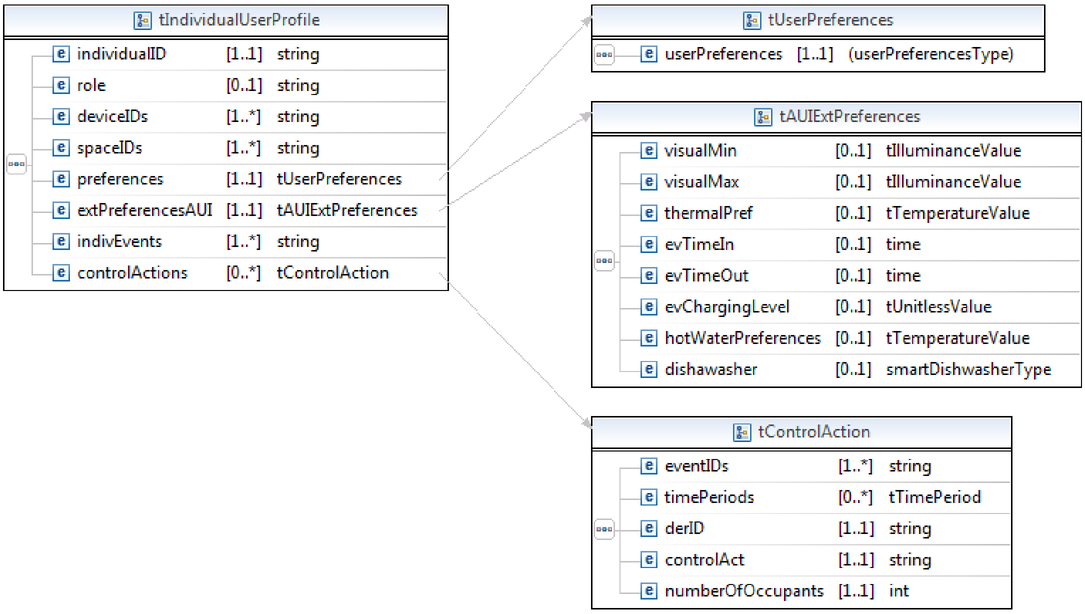



***Figure 28: INERTIA individual User profile schema***

## Inertia DER Type

The specification is done in CIM for LinkSmart

## Inertia BIM Type

The specification is done in CIM for Linksmart or OCC

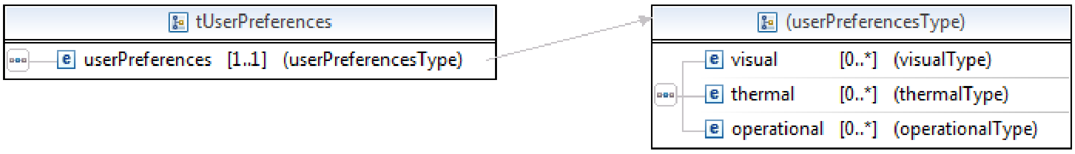



***Figure 29: User Preferences XSD specification***

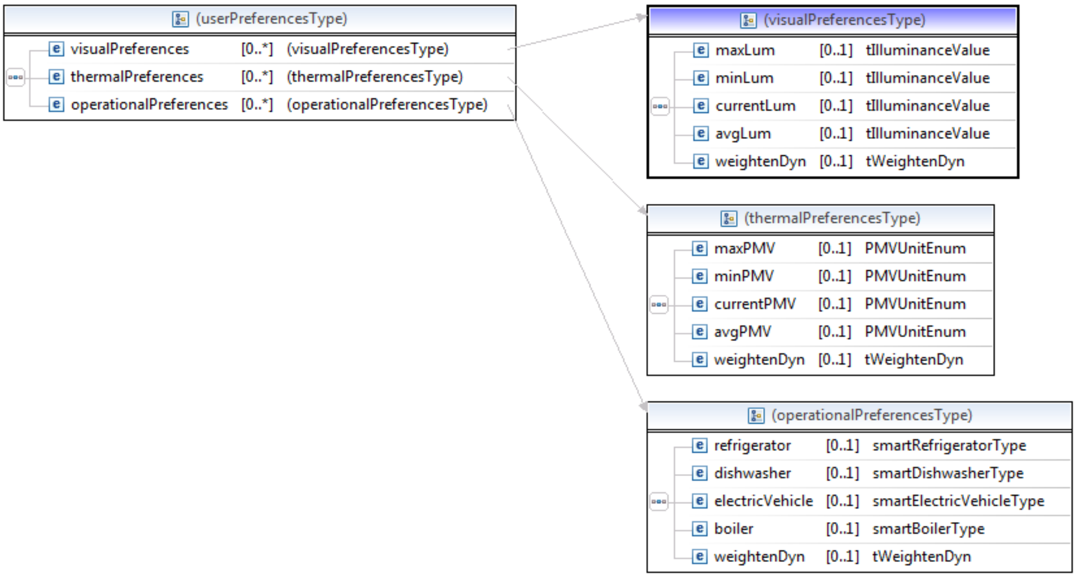



***Figure 30: User Preferences type specified as Visual Preferences type, Thermal Preferences type***

***Operational preferences type***

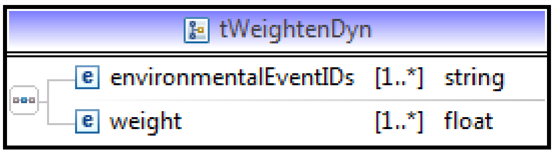



***Figure 31: Weighted Dynamic XSD specification***

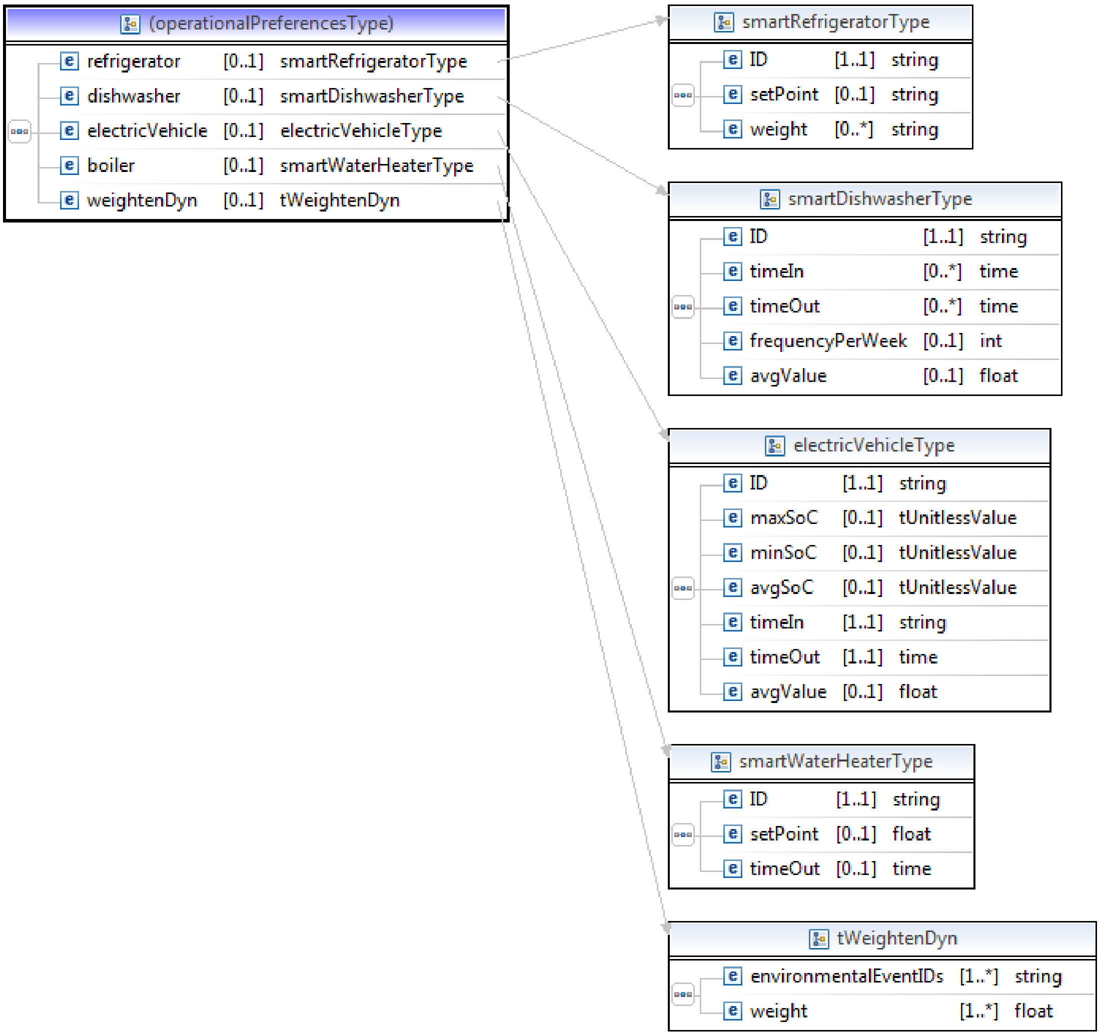



***Figure 32: Operational preference with modelled devices XSD specification***

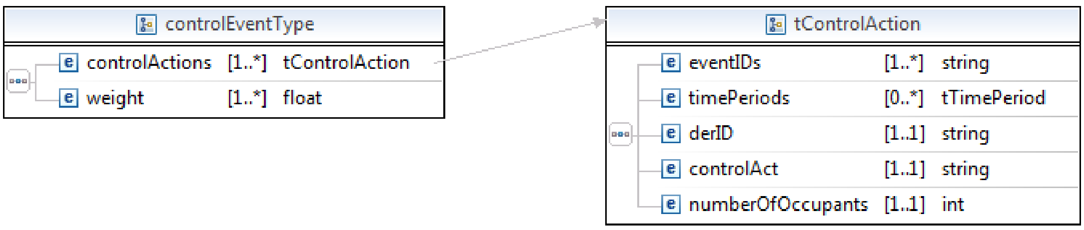



***Figure 33: Control Actions XSD specification and Control Dynamic XSD specification***

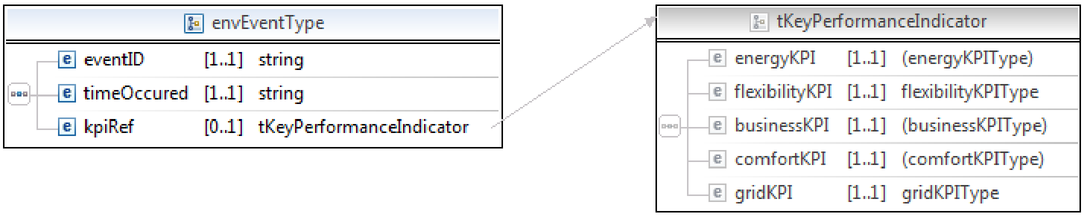



***Figure 34: Environmental Event specification***

**DB Schemas**

SQL specification:

User preferences:

```
CREATE TABLE indiv_or_group_preferences
(
	entryPreferenceID int NOT NULL PRIMARY KEY,
	userID (255)
	groupID (255)
);



CREATE TABLE user_preferences_visual
(
	entryID int NOT NULL PRIMARY KEY,
	preferencesID int NOT NULL FOREIGN KEY,
	maxLum float,
	minLum float,
	currentLum float,
	avgLum float,
	unitRef (255)
);
CREATE TABLE user_preferences_thermal
```

```
(
        entryID int NOT NULL PRIMARY KEY,
        preferencesID int NOT NULL FOREIGN KEY,
        maxPMV int,
        minPMV int,
        currentPMV int,
        avgPMV int,
        unitRef (255)
);
CREATE TABLE user_preferences_operational
(
        entryID int NOT NULL PRIMARY KEY,
        preferencesID int NOT NULL FOREIGN KEY,
        refrigeratorID (255),
        refrigeratorSetPoint (255),
        dishWasherID (255),
        dWasherTimeIn time,
        dWasherTimeOut time,
        dWasherFrekvPerWeek int,
        dWasherAvgValue float,
        evID (255),
        evMaxSoC float,
        evMinSoC float,
        evAvgSoC float,
        evTimeIn time,
        evTimeOut time,
        evAvgValue float,
        waterHeaterID (255),
```

```
    whSetPoint float,
    whTimeOut time,
);
CREATE TABLE user_weighten_dynpref
(
    entryID int NOT NULL PRIMARY KEY,
    preferencesID int NOT NULL FOREIGN KEY,
    iPreferID int NOT NULL FOREIGN KEY,
    iPreferTypeIN int NOT NULL FOREIGN KEY,
    envEventID (255) NOT NULL,
    weight float NOT NULL,
);
```

**Related semantic models**

The fields with annotation to the DER semantic models are device IDs (refrigeratorID, dishWasherID, evID, waterHeaterID) while occupantID and groupID holds the annotation to the occupancy related semantic model (both modes will be defined in D3.2).

### *6.1.4* Ambient User Interface in CIM

**Local Hub IAM interfaces**

| |
|---|
| UI_getHistoricalDERData |
| UI_getHistoricalDataForSpace |
| OCC_importHistoricalOccupancy |
| OCC_importHistoricalOccupantLocation |
| OCC_importRealTimeOccupancy |
| OCC_importRealTimeOccupantLocation |
| OCC_importSpatialSchedule |

| |
|---|
| OCC_importIndividualIInertiaOccupancyAndFlowModel |
| UI_getTariffs |
| UI_getEmissions |
| UP_getListOfDERsForOpMode |
| getLocationModel |
| getDERModel<br><br>getDeviceLocationModel |
| thermalPreferences UP_getGroupPMV(String occupantID) |
| visualPreferencesType UP_getGroupLuminance(String occupantID) |
| thermalPreferences UP_getOccupantOrStereotypePMV(<br><br>String occupantID) |
| visualPreferencesType UP_getOccupantOrStereotypeLuminance(<br><br>String occupantID<br><br>) |
| UI_getLoggingData |
| UI_insertOperationalSchedules |
| UI_insertOperationalSchedules |
| UI_insertOperationalSchedules |
| UI_insertOperationalSchedules |
| UI_updateOperationalSchedules |
| OCC_exportSpatialSchedule |
| UI_insertOperationalMode |
| UI_updateOperationalMode |
| UI_deleteOperationalMode |
| UI_setOpIntervalforDERforOpMode |

## XSD schemas:

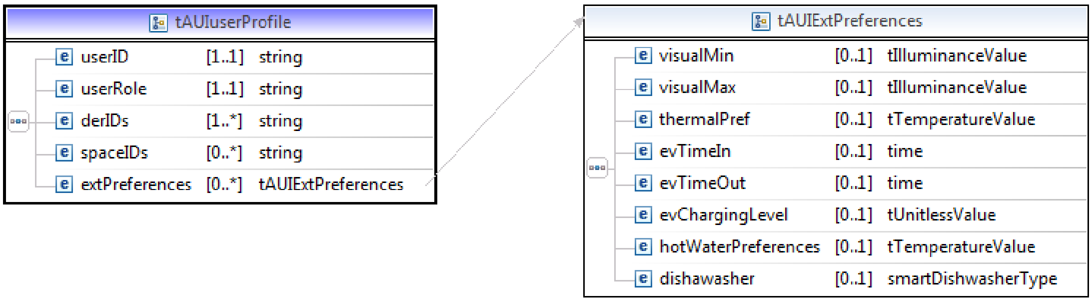



***Figure 35: Explicit Preferences of user in AUI XSD specification***

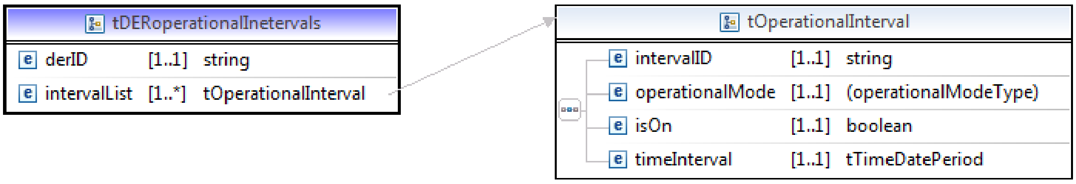



***Figure 36: DER operational intervals in AUI XSD specification***

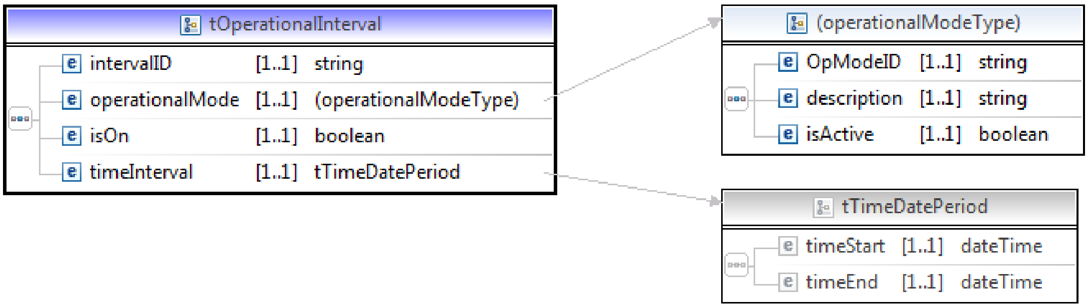



***Figure 37: Operational interval with operational mode in AUI XSD specification***

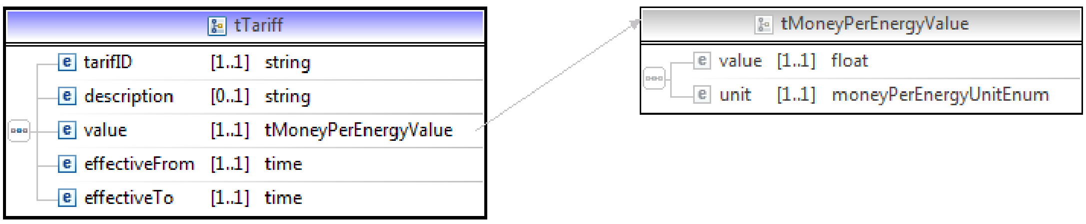



***Figure 38: Tariff in AUI XSD specification***

**DB Schemas:**

SQL specification:

```
CREATE TABLE aui_tariff
(
    entryID int NOT NULL PRIMARY KEY,
    tarifID (255) NOT NULL,
    descriptionID (255),
    value float NOT NULL,
    unit (255)
    effectiveFrom time,
    effectiveTo time,
);
```

DER operational intervals:

```
CREATE TABLE aui_der_operational
(
    entryID int NOT NULL PRIMARY KEY,
    derID (255), NOT NULL


);
CREATE TABLE aui_der_operational
(
    entryID int NOT NULL PRIMARY KEY,
    derOerID int NOT NULL FOREIGN KEY,
    opMode (255) NOT NULL,
    description (255) NOT NULL,
```

```
    isActive boolean NOT NULL,
    isOn Boolean NOT NULL,
    timeFrom dateTime NOT NULL,
    timeTo dateTime NOT NULL
);
```

User profile:

```
CREATE TABLE aui_userprof_head
(
    entryID int NOT NULL PRIMARY KEY,
    userID (255) NOT NULL
    roleID (255) NOT NULL
);
CREATE TABLE aui_userprof_ders
(
    entryID int NOT NULL PRIMARY KEY,
    upID int NOT NULL FOREIGN KEY,
    derID (255) NOT NULL
);
CREATE TABLE aui_userprof_spaces
(
    entryID int NOT NULL PRIMARY KEY,
    upID int NOT NULL FOREIGN KEY,
    spaceID (255) NOT NULL
);
CREATE TABLE aui_userprof_extpref
(
```

```
    entryID int NOT NULL PRIMARY KEY,
    upID int NOT NULL FOREIGN KEY,
    visualMin float,
    visualMax float,
    thermalPref float,
    evTimeIn time,
    evTimeOut time,
    evChargingLevel float,
    hotWaterPref float
);
CREATE TABLE aui_userprof_extpref
(
    entryID int NOT NULL PRIMARY KEY,
    upID int NOT NULL FOREIGN KEY,
    dishWasherID (255),
    dWasherTimeIn time,
    dWasherTimeOut time,
    dWasherFrekvPerWeek int,
    dWasherAvgValue float,
)
```

**Related semantic models**

The fields with annotation to the DER semantic models are flowing derID device IDs dishWasherID. SpaceID field holds the annotation to BIM semantic model. UserID and roleID holds the annotation to the occupancy related semantic model. These semantic models will be defined in D3.2.

### *6.1.5* KPIs in CIM (Energy Performance Modelling component)

**IAM interface**

To be able to store KPI with context data following interface signature was proposed:

```
IAM.importKPI(
        String spatialIDRef,
        String derIDRef,
        String occupantGroupIDRef,
        String occupantIDRef,
        dateTime timeFrom
        dateTime timeTo
        tKeyPerformanceIndicator KPI
        )
```

**XSD schemas**

The following XSD schemas are based on the KPI categories, individual KPIs and their unit which are defined in D2.3.1. The source code of the schemas can be found the Annex II.

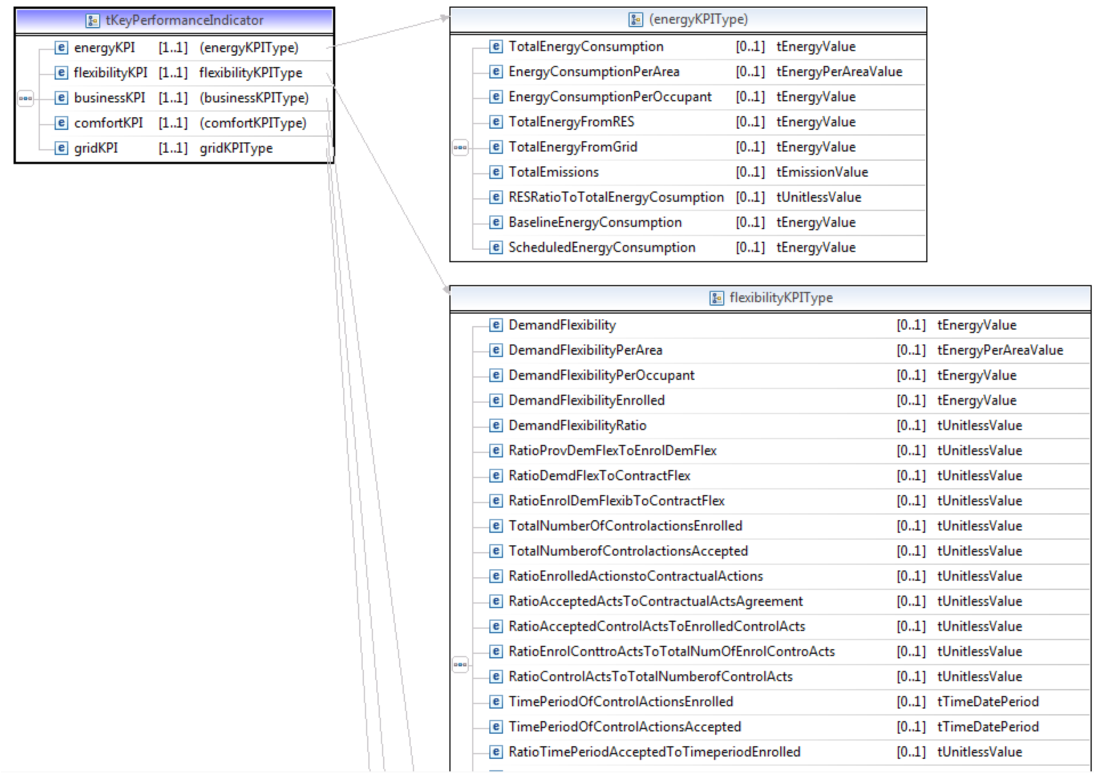



***Figure 39: KPI categorization schema***

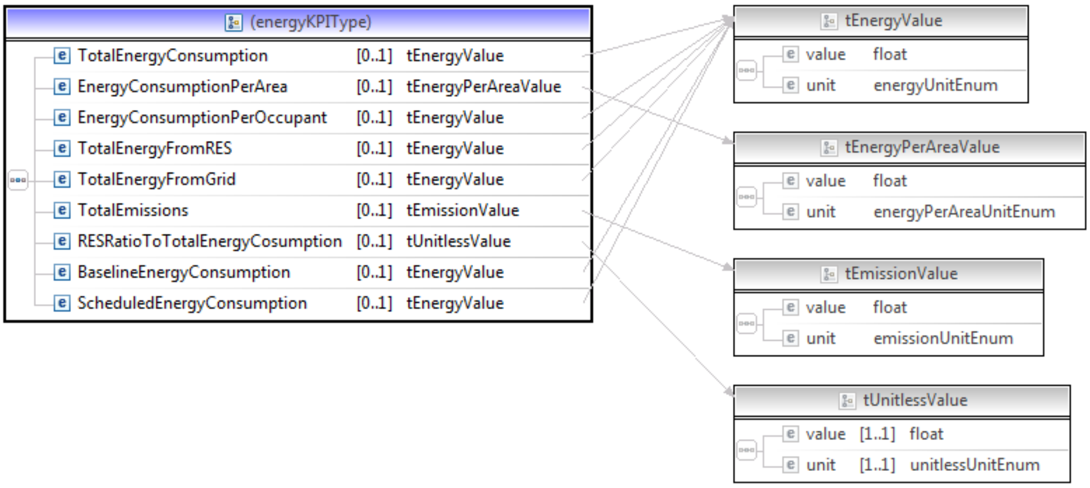



***Figure 40: Energy KPIs schema***

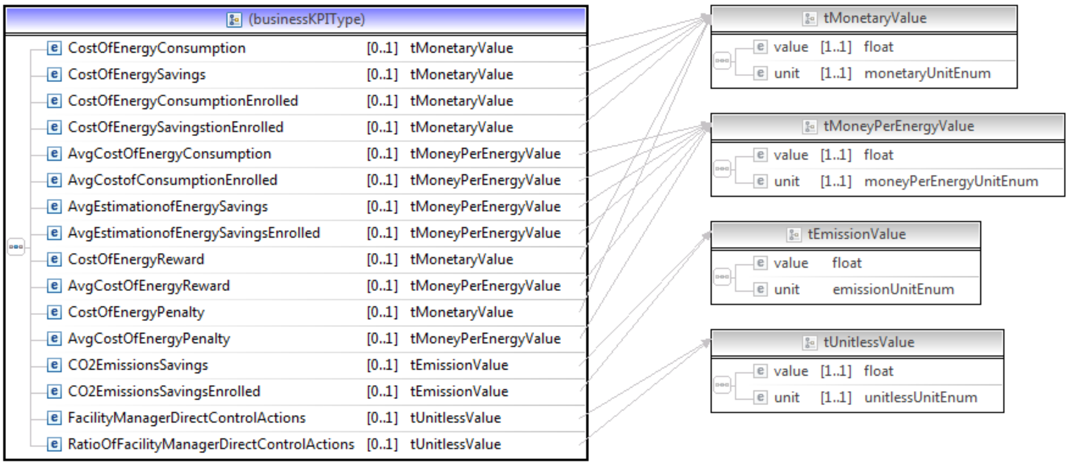



***Figure 41: Business KPIs schema***

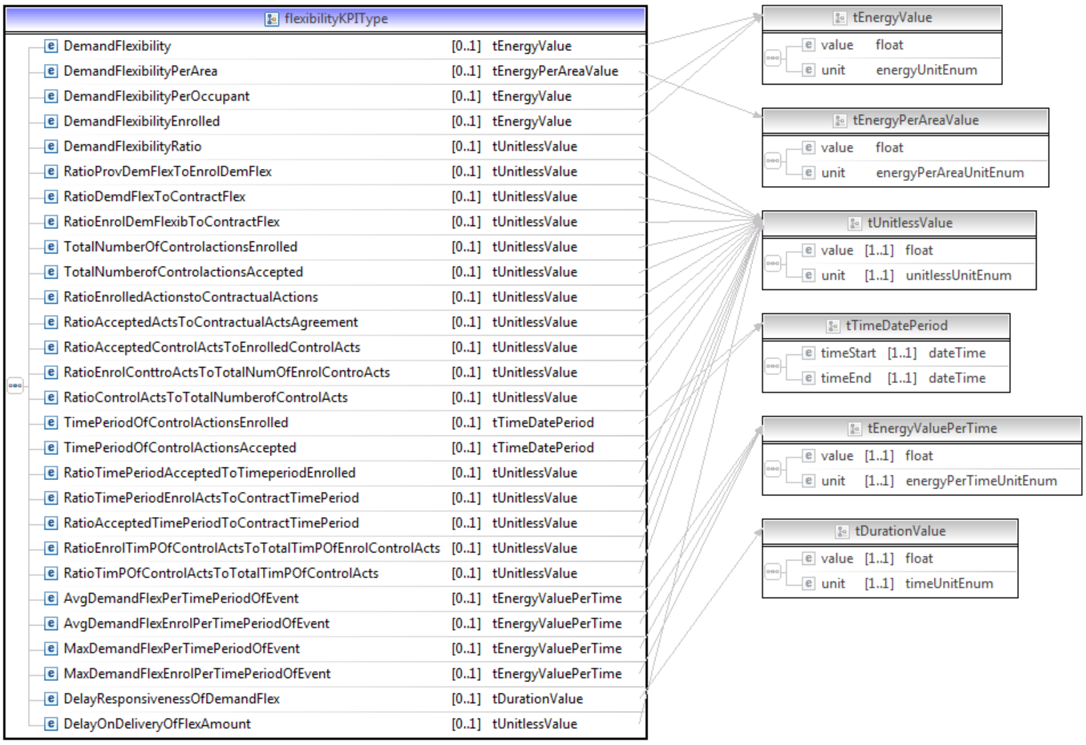



***Figure 42: Flexibility KPIs schema***

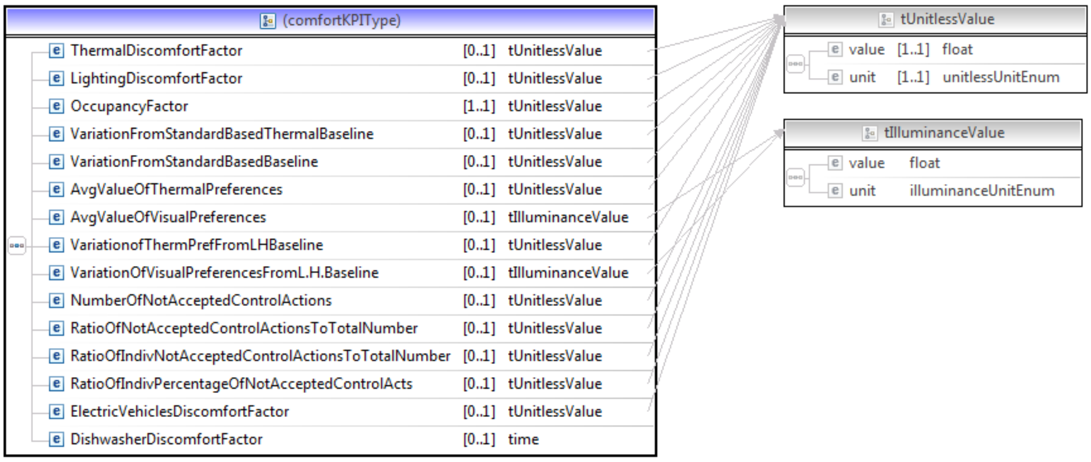



***Figure 43: Comfort KPIs schema***

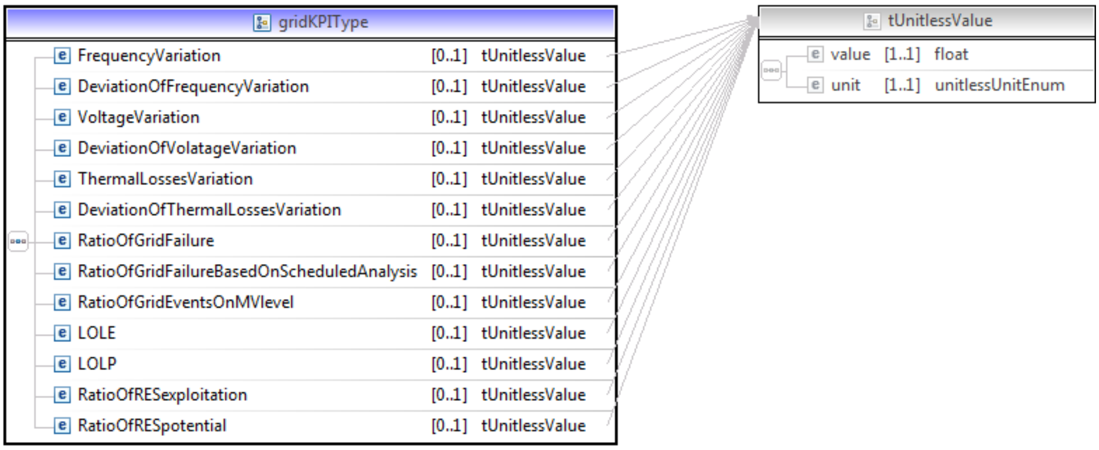



***Figure 44: Grid KPIs schema***

**DB schemas**

To store KPIs in IAM in the form presented above is not enough. The relevant aspect to KPIs calculation is to be saved as well (i.e. context information). Thus, aspects such as time period, reference to BIM model, DER model, Occupancy model and Unit model should be possible to store and then to retrieve.

SQL specification:

```
CREATE TABLE KPI_base

(

        entryID int NOT NULL PRIMARY KEY,

        KPIcategory (255) NOT NULL

        KPIname (255) NOT NULL

        KPIvalue (255) NOT NULL

        valueType (255),

        timeFrom datetime NOT NULL,

        timeTo datetime NOT NULL

);

CREATE TABLE KPI_spaces

(

        entryID int NOT NULL PRIMARY KEY,
```

```
    kpiID int NOT NULL FOREIGN KEY,
    spatialID (255) NOT NULL
);
CREATE TABLE KPI_occupancy
(
    entryID int NOT NULL PRIMARY KEY,
    kpiID int NOT NULL FOREIGN KEY,
    occupantlID (255)
    occupantGroupID (255)
);
CREATE TABLE KPI_der
(
    entryID int NOT NULL PRIMARY KEY,
    kpiID int NOT NULL FOREIGN KEY,
    derlID (255)
);
```

**Related semantic models**

The entries of the KPIs in the DB tables contain references/annotaion to BIM model (spatialID), DER model (derID), Occupancy model (occupantlID, occupantGroupID) and Unit model (valueType).

### *6.1.6* MAS Local Hub Gateway in Local Hub CIM

**Local Hub IAM interfaces**

ProsumersControl_exportForecastProfile

**XSD schemas:**

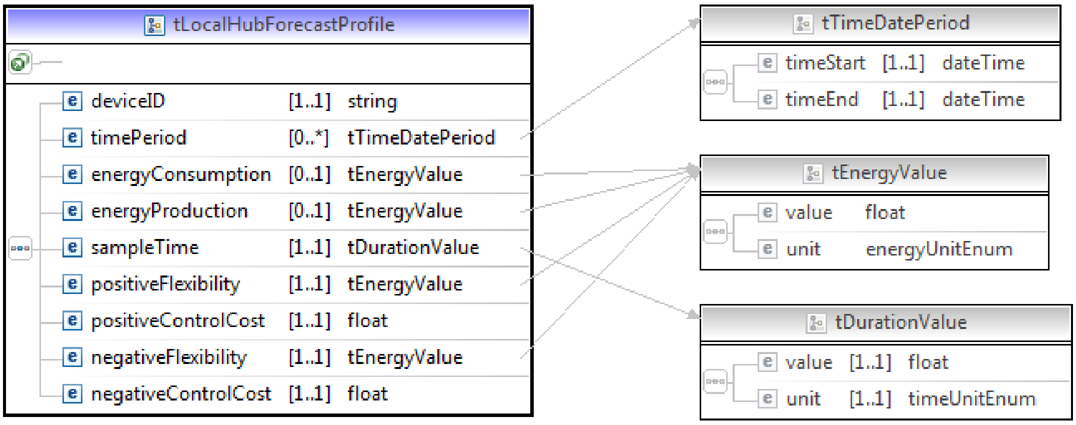



***Figure 45: Local Hub forecast profile XSD specification***

**DB schemas:**

SQL specification:

```
CREATE TABLE Prosumers_cont_optim_forecast
(
    entryID int NOT NULL PRIMARY KEY,
    deviceID (255) NOT NULL
    energyCon float
    energyProd float
    posFlex float NOT NULL
    negFlex float NOT NULL
    posContCost float NOT NULL
    negContCost float NOT NULL
    timeFrom datetime NOT NULL
    timeTo datetime NOT NULL
    sampleTime (255)

);
```

**Related semantic models**

In case of Prosumers Control & Optimization data on the local hub level the device ID is sent and stored in the IAM DB table. This ID refers to the semantic model (the ID has form of URI of semantic model instance) and enables to retrieve more context (such as static characteristics of device - energy profile, device location) if needed.

### *6.1.7* Local Hub DER Flexibility Modelling in CIM

**IAM Interface**

The interface savin flexibility of local hub is the following:

- DERflex_setLocalHubFlexibility

Since the binding of data stored in IAM and calculation of DER prediction based on DER model will be defined in detail in D2.4.2 INERTIA Multi-dimensional DER flexibility models see this deliverable for all IAM interfaces dedicated for DER calculation support.

**XSD specification**

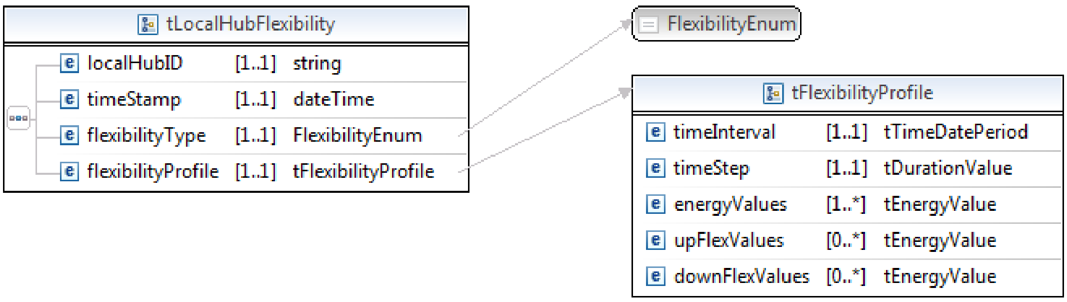



***Figure 46: Local hub flexibility XSD specification***

Parameter flexibilityType can have two values: DemandFlexibility or GenerationFlexibility.

See D2.4.2 INERTIA Multi-dimensional DER flexibility models for updated CIM XSD schemas.

**DB schemas**

Basically the DB schema for LinkSmart event is used to store data about DER states. The flexibility of Local Control Hub is stored in DB tables designed for the needs of MAS Based Holistic Flexibilityin Local Hub. The states of the devices are serving to support monitoring and controlling of devices while the flexibility of control hub calculated and sored by MAS components provide the basis for the IAM interface/service. The table entries contain also annotation to the DER ontology.

#### 6.1.7.1 DER semantic model in CIM

Using semantic annotations in data connect the RAW data stored in databases with the semantic model (see the example of DER ontology in INERTIA deliverable D3.1 and full specification in deliverable D3.2 as well as IAM support of DER calculations in D2.4.1). Semantic annotations hold the stable vocabulary of nominal parameters across all INERTIA components. The reference enables to connect device with any BIM information and the references enable to connect device Status and value Type with information about devices such as All possible device statuses and respective Energy consumption information, units etc. By using semantic annotations, it is possible to retrieve additional semantic context to be used for further decision making or for retrieving additional information.

The DER ontology model needs to represent hierarchy of devices and their profiles parts of DER and need to be related to concept of BIM ontology so the location of devices or DERs is known. This semantic model will contains the instance per each concrete DER. The DER has assigned consuming devices. The devices are organized in the class hierarchy. Each device has assigned the set of statuses (e.g. On, Off, StanBy, etc.). Devices have also attached the energy profiles tied with these statuses. The energy profile parameters may have different values (which have also assigned the units) depending on the status. The prototype semantic model of DER can be found in the INERTIA deliverable D3.1 and final DER model will be defined in the INERTIA deliverable D3.2 (Ontology Specification).

## *6.2 Aggregator Control Hub CIM*

### *6.2.1* Prosumers Control & Optimization in Aggregator CIM

**Aggregator IAM interface**

Interface for storage

ProsCO_exportForecastProfile

**XSD schemas**

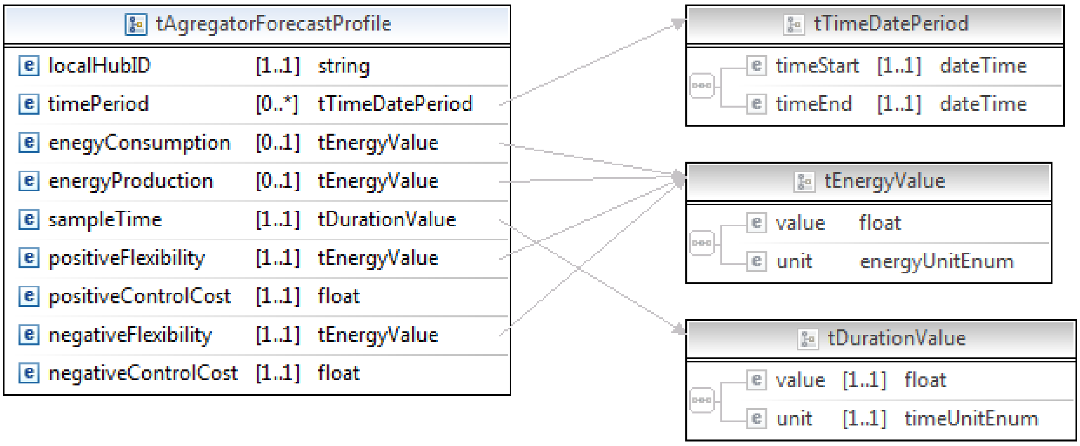



***Figure 47: Aggregator forecast profile XSD schema***

**DB schema**

SQL specification:

CREATE TABLE pros_co_forecast

(

entryID int NOT NULL PRIMARY KEY,

LH_ID (255) NOT NULL

energyCon float

energyProd float

posFlex float NOT NULL

negFlex float NOT NULL

```
    posContCost float NOT NULL
    negContCost float NOT NULL
    timeFrom datetime NOT NULL
    timeTo datetime NOT NULL
    sampleTime (255)
);

CREATE TABLE pros_co_drstatiscics_segment_values
(
    entryID int NOT NULL PRIMARY KEY,
    LH_ID (255) NOT NULL ,
    name (255) NOT NULL,
    valueName (255) NOT NULL
    value float NOT NULL,
);
CREATE TABLE prosumers_cont_optim_itemdetails
(
    entryID int NOT NULL PRIMARY KEY,
    segmentID int NOT NULL FOREIGN KEY,
    itemID (255) NOT NULL,
    detail NOT NULL
);
CREATE TABLE pros_co_drstatiscics_metric_values
(
    entryID int NOT NULL PRIMARY KEY,
    LH_ID (255) NOT NULL,
    valueName (255) NOT NULL
    value float NOT NULL,
```

unit (255) NOT NULL

timestamp datetime NOT NULL

);

**Related semantic models**

In case of Prosumers Control & Optimization data on the aggregator level the Local Hub ID is sent and stored in the IAM DB table. This ID refers to the semantic model and enables to retrieve more context data (such as information about structure of Buildings of LH or any information about DER) if needed.

### *6.2.2* Grid Coordinator& DR Activation in Aggregator CIM

**Aggregator IAM interfaces**

Exporting interface

GridCoordinator_setDSODispatch

GridCoordinator_setMV2LCHMapping

Importing interface

GridCoordinator_getMV2LCHMapping

**XSD schemas**

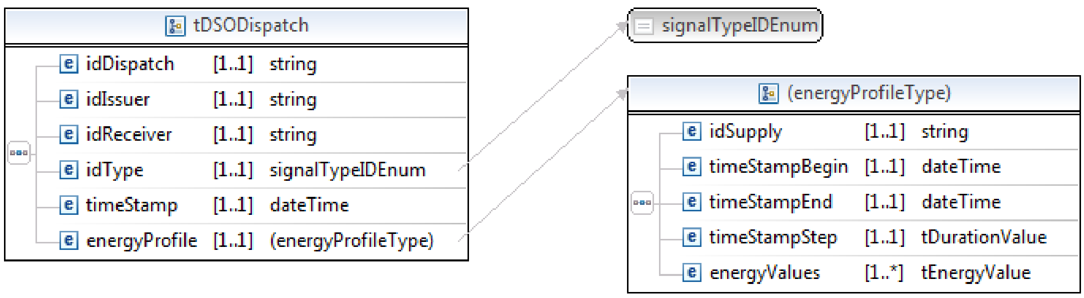



***Figure 48: DSO dispatch data type schema used in GridCoordinator_setDSODispatch interface***

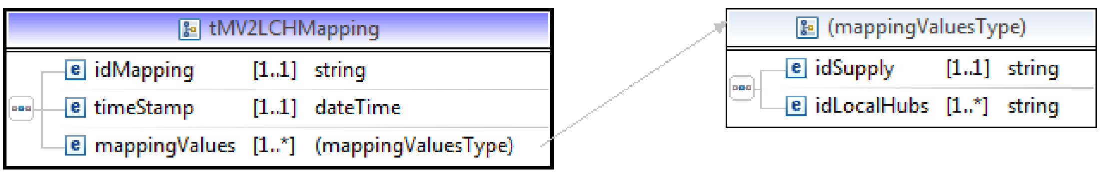



***Figure 49: MV2LCHMapping data type schema used in GridCoordinator_getMV2LCHMapping interface***

**DB schemas**

SQL specification for DSO dispatch:

```
CREATE TABLE dso_dispatch
(
	entryID int NOT NULL PRIMARY KEY,
	idDispatch (255) NOT NULL
	idIssuer (255) NOT NULL
	idReceiver (255) NOT NULL
	idType (255) NOT NULL
	timeStamp datetime NOT NULL
);
CREATE TABLE energy_profile
(
	entryID int NOT NULL PRIMARY KEY,
	dsoDispatchID int NOT NULL FOREIGN KEY,
	idSupply(255) NOT NULL
	timeStampBegin datetime NOT NULL
	timeStampEnd datetime NOT NULL
	timeStampStep float NOT NULL
	energyValue float NOT NULL
);
```

SQL specification for MV to LH Mapping:

```
CREATE TABLE mv2lh_mapping
(
	entryID int NOT NULL PRIMARY KEY,
	idMapping (255) NOT NULL
	timeStamp datetime NOT NULL
);
CREATE TABLE map2supplies
```

```
(
        entryID int NOT NULL PRIMARY KEY,
        mv2lhID int NOT NULL FOREIGN KEY,
        idSupply(255) NOT NULL
);
CREATE TABLE supplies2localhubs
(
        entryID int NOT NULL PRIMARY KEY,
        map2supID int NOT NULL FOREIGN KEY,
        idLocalHubs(255) NOT NULL
);
```

**Related semantic models**

Local Hub ID stored in the DB table refers to the semantic model and enables to retrieve more context data (such as information about structure of Buildings of LH or any information about DER) if needed.

### *6.2.3* DER Flexibility Analysis, Aggregation and Forecasting

This component is also known as Visual Analytics tool (VA)

**Aggregator IAM Interfaces**

Interfaces for retrieval

- VA_getAllDREventsForAggregator
- VA_getAllDRStatisticsForAggregator
- VA_getStatisticsForAllDRRequestsFromAggregator
- VA_getStatisticsForDRRequest
- VA_getKPIsForAggregator
- VA_getKPIsForDRSignal
- VA_getDRStatisticsPerTimestep
- VA_getCandidateLocalHubsForDRSignalResponse
- VA_getHighLevelStrategy

Interfaces for storage

- VA_storeCandidatesForDRSignalResponse
- VA_storeHighLevelStrategy
- VA_updateHighLevelStrategy

**XSD schemas**

The XSD schemas for flexibility data (see CIM for Prosumers Control & Optimization component, CIM for MAS Local Hub Gateway component and CIM for Local Hub DER Flexibility Modelling component) and DSO dispatch data (see CIM for Grid Coordinator& DR Activation component) serve IAM retrieval interfaces relevant for VA tool.

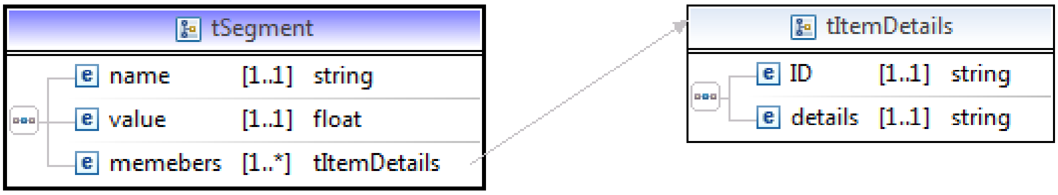



***Figure 50: XSD schema used in IAM interfaces VA_storeCandidatesForDRSignalResponse and VA_getCandidateLocalHubsForDRSignalResponse***

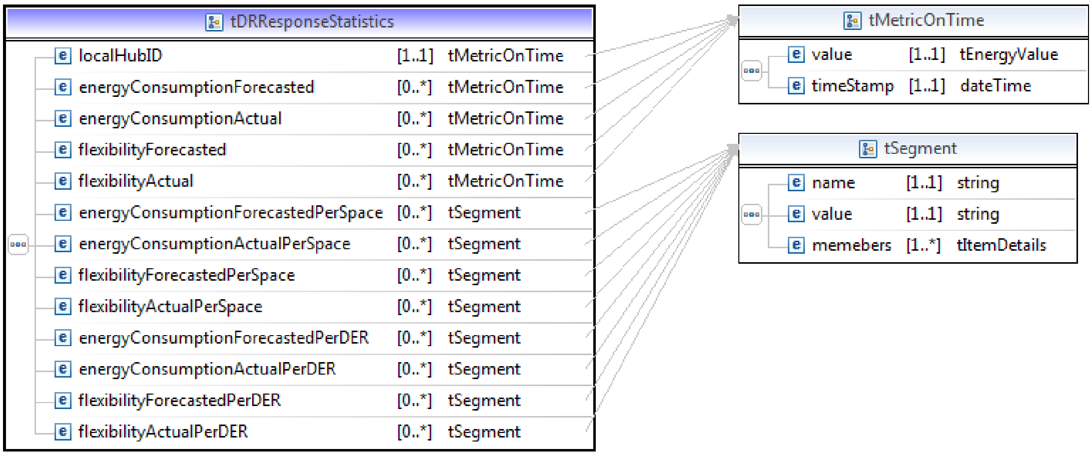



***Figure 51: DR response Statistics XSD schema***

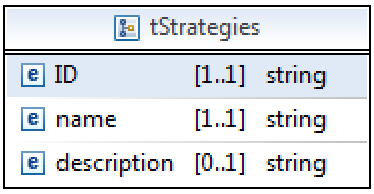



***Figure 52: XSD schema for high-level strategies***

**DB schemas**

The DB schemas for flexibility data (see Aggregator CIM for Prosumers Control & Optimization component, CIM for MAS Local Hub Gateway component and CIM for Local Hub DER Flexibility Modelling component) and DSO dispatch data (see CIM for Grid Coordinator& DR Activation component) and schemas for storing KPIs (see KPIs in CIM) serve for data operation to the IAM interfaces relevant for VA tool.

SQL specification of VA_storeCandidatesForDRSignalResponse:

```
CREATE TABLE va_drsig_segment_candidates

(

        entryID int NOT NULL PRIMARY KEY,

        name (255) NOT NULL,

        value float NOT NULL

);

CREATE TABLE va_drsig_itemdetails

(

        entryID int NOT NULL PRIMARY KEY,

        drCandidatesID int NOT NULL FOREIGN KEY,

        itemID (255) NOT NULL,

        detail NOT NULL

);


CREATE TABLE va_hl_strategies

(

        entryID int NOT NULL PRIMARY KEY,

        strategyID (255) NOT NULL,

        name (255) NOT NULL,

        description (255)

);
```

**Related semantic models**

No semantic annotations for data stored by this component are expected. Thus, there are no fields in the DB tables containing identifiers to the instances of semantic models.

# 7. Conclusion

This deliverable presented the results of Task T2.1 – Common Information Model. More specifically, this deliverable defines Common Information Model (CIM) for Local Control Hub and Aggregator Control Hub of the INERTIA demand side management Framework.

The CIM was defined as vocabulary that enables to address the information needs for the INERTIA framework components. CIM describes the heterogeneous information sources of the INERTIA components in a unified way. The vocabulary is used by all the INERTIA components and thus any INERTIA component can access information elements that are underlying the INERTIA Framework.

The requirements from INERTIA components on the IAM were the basis for the definition of CIM. However, the information models from other relevant projects and some of the industry standards were analysed as a possible basis.

The relevant projects and industry standards were analysed during first phase of T2.1, since there were not concrete/stable requirements from component's developers on the IAM. The projects that deal with similar topics were identified and their information models were analysed during this period. Some of these specifications directly influenced CIM definition as pointed out in Section 3. At the same time, the industry standards that could be possibly used as an inspiration source were identified and analysed. The results from such analyses are documented in Section 3 and 4.

Even though the Information models from other project as well as chosen industry standards were also taken into consideration as a possible basis for the definition of the INERTIA CIM their impact is rather marginal. The main basis for the definition of CIM was the requirements from INERTIA components on storage services provided by the IAM. Thus the analysis (presented in Section 5) and specifications (presented in Section 6) of Local Hub CIM and Aggregator Hub CIM are structured based on the names of these components. More explicitly, the following structure was adopted:

- Local Hub CIM
    - LinkSmart
    - Occupancy
    - User Profiling
    - Ambient User Interface
    - Multi Agent Based Local Hub Gateway
    - Local Hub DER Flexibility
    - Energy Performance Modelling and KPIs
- Aggregator Hub CIM
    - Multi Agent Based Prosumers Control & Optimization
    - Grid Coordinator& DR Activation
    - Aggregator Analytics

The requirements on the IAM from the component's developers were collected, analysed, harmonised and then used for the definition of CIM. These requirements were collected in form of unified descriptions of component's interfaces that will store data to or retrieve data from the IAM. The data types specific for INERTIA were described in a unified form (by partners responsible for component development). These descriptions are documented in Section 5.

The obtained descriptions were then analysed and harmonised with the intention to define CIM. The CIM was then defined. The definition of CIM consists of two parts that enable Inertia components to access dedicated IAM data services:

- Names and signatures of IAM interfaces that provide data service to the INERTIA components,
- The specification of data types that are used in these interfaces

These parts of CIM are presented in Section 6.

Furthermore, there are two additional parts of CIM needed to represent, combine and manipulate with data within IAM, and thus enable to provide the data service.

- Database schemas that serve for storing dynamic data,
- Semantic models that serve for storing static data

The database schemas are presented in Section 6 of this deliverable. Some of the fields defined by these database schemas are designed for holding annotation to the semantic model. Such fields are inputs into design of semantic models that is being performed within task T3.3. The complete semantic models will be presented in report D3.2, while the prototype semantic models can be found in the report D3.1.

Note, the presented INERTIA CIM reflects its current state. Although the INERTIA task T2.1 dedicated to the definition of CIM finishes, this version of CIM is most likely not a final one. The reason is that there are other tasks running and these tasks may impact the CIM and induce its modifications in next project period. Additionally, since the INERTIA modelling activities are not closed yet, it was not possible to cover all dependencies between stored information in the IAM and requested information from IAM in this deliverable.

# References

[1] INERTIA Grant Agreement Annex I – "Description of Work" (DoW)

[2] Common Information Model (computing) http://en.wikipedia.org/wiki/Common_Information_Model_%28computing%29 (30.9.2013)

[3] Adapt4EE project. http://www.adapt4ee.eu/adapt4ee/project/index.html (30.9.2013)

[4] Skokan, M., et al., Adapt4EE Common Information Model, Adapt4EE Project deliverable D2.2, 2012.

[5] Mirable D2.3. http://wwwdb.inf.tu-dresden.de/miracle/publications/D2.3.pdf

[6] Fenix VPP economic perspective. http://fenix.iwes.fraunhofer.de/docs/documents/Project_Fenix_2009-07-20_The_virtual_power_plant_concept_from_an_economic_perspective_Jaap_Jansen_v1.pdf

[7] Fenix vision. http://fenix.iwes.fraunhofer.de/docs/documents/Project_Fenix_2008-06-19_The_FENIX_vision__Christophe_Kieny_v1.pdf

[8] Fenix conceptual framework. http://fenix.iwes.fraunhofer.de/docs/documents/Project_Fenix_2009-07-06_WP3_2_6_Contractual_Framework_FullReport_Dafydd%20Elis_v1.pdf

[9] IEC 61970: Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base, Third Edition (2009)

[10] CIMTool. http://wiki.cimtool.org/index.html (30.9.2013)

[11] INTERNATIONAL STANDARD IEC 61968-1 Application integration at electric utilities – System interfaces for distribution management – Part 1: Interface architecture and general requirements, First edition (2003)

[12] INTERNATIONAL STANDARD IEC 61968-11 Application integration at electric utilities – System interfaces for distribution management – Part 11: Common information model (CIM) extensions for distribution, Edition 1.0 (2010)

[13] IEC 61968 wikipedia. http://en.wikipedia.org/wiki/IEC_61968 (30.9.2013)

[14] INTERNATIONAL STANDARD IEC 61850-7-420 Communication networks nad systems for power utility automation - Part 7-420: Basic communication structure - Distributed energy resources logical nodes, Final Draft International Standard (FDIS), available at http://osgug.ucaiug.org/

[15] Common Information Model (electricity). http://en.wikipedia.org/wiki/Common_Information_Model_%28electricity%29 (30.9.2013)

[16] The Green Building XML (gbXML) open schema. http://www.gbxml.org/

[17] Deliverable D2.2: End-to-End Platform specification

BeyWatch Data Model (Annex), (2008).

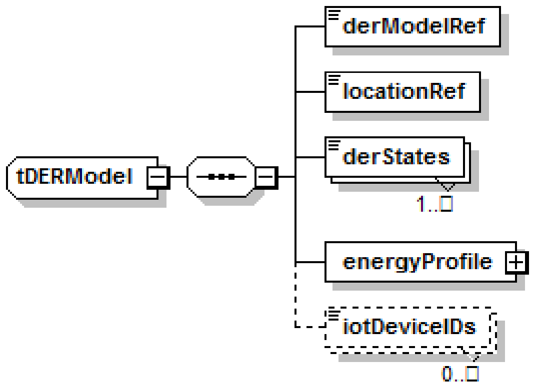
# Annex I: Technical specification of XSD schemas

## Schema inertiaLinkSmart.xsd

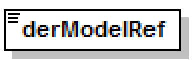
| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaLinkSmart.xsd** |
| attributeFormDefault: | **unqualified** |
| elementFormDefault: | **unqualified** |
| targetNamespace: | **http://www.inertia.eu/2013/schema/event/** |

Complex types
**tDERModel**
**tEnergyProfile**
**tLinkSmartEvent**
**tSpaceWithDER**
**tSpatialStructure**

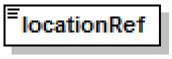
### complexType **tDERModel**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/event/ |
| children | **derModelRef locationRef derStates energyProfile iotDeviceIDs** |
| source | <complexType name="tDERModel"><br><sequence><br><element name="derModelRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="locationRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="derStates" type="string" minOccurs="1" maxOccurs="unbounded"/><br><element name="energyProfile" type="inerevent:tEnergyProfile" minOccurs="1" maxOccurs="1"/><br><element name="iotDeviceIDs" type="string" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

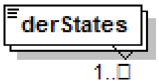
### element **tDERModel/derModelRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="derModelRef" type="string" minOccurs="1" maxOccurs="1"/> |

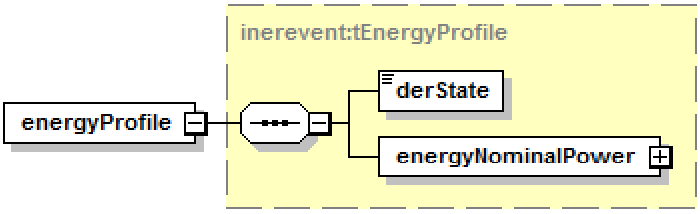
element **tDERModel/locationRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="locationRef" type="string" minOccurs="1" maxOccurs="1"/> |

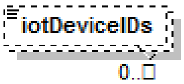
element **tDERModel/derStates**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="derStates" type="string" minOccurs="1" maxOccurs="unbounded"/> |

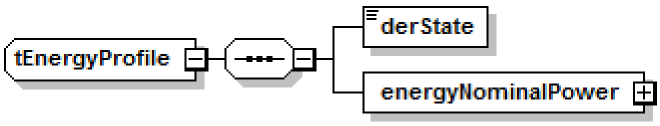
element **tDERModel/energyProfile**

| | |
|---|---|
| diagram |  |
| type | **inerevent:tEnergyProfile** |
| properties | content complex |
| children | **derState energyNominalPower** |
| source | <element name="energyProfile" type="inerevent:tEnergyProfile" minOccurs="1" maxOccurs="1"/> |

element **tDERModel/iotDeviceIDs**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="iotDeviceIDs" type="string" minOccurs="0" maxOccurs="unbounded"/> |

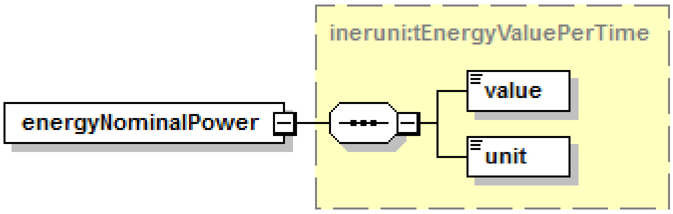
complexType **tEnergyProfile**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/event/ |
| children | **derState energyNominalPower** |
| used by | element **tDERModel/energyProfile** |
| source | <complexType name="tEnergyProfile"><br><sequence><br><element name="derState" type="string" minOccurs="1" maxOccurs="1"/><br><element name="energyNominalPower" type="ineruni:tEnergyValuePerTime" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

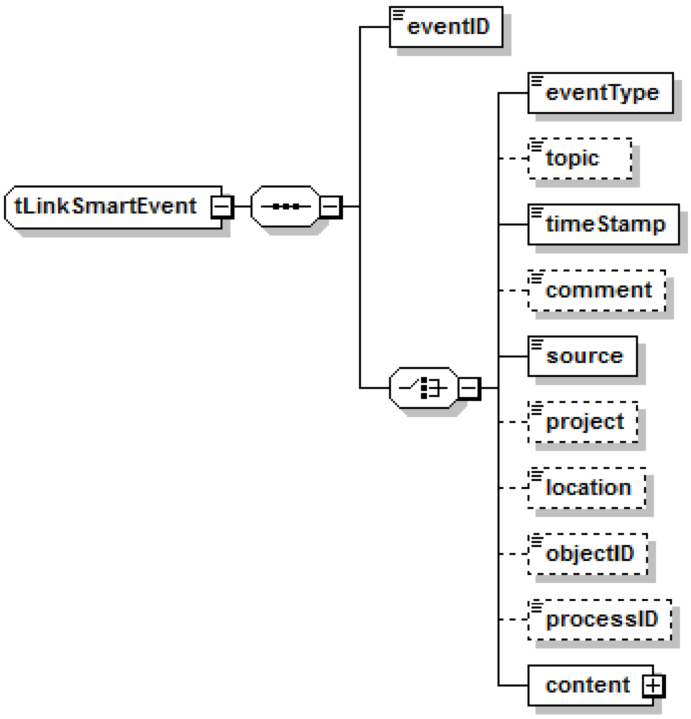
element **tEnergyProfile/derState**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="derState" type="string" minOccurs="1" maxOccurs="1"/> |

element **tEnergyProfile/energyNominalPower**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValuePerTime** |
| properties | content complex |
| children | **value unit** |
| source | <element name="energyNominalPower" type="ineruni:tEnergyValuePerTime" minOccurs="1" maxOccurs="1"/> |

complexType **tLinkSmartEvent**

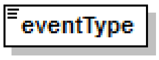
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/event/ |
| children | **eventID eventType topic timeStamp comment source project location objectID processID content** |
| source | <complexType name="tLinkSmartEvent"><br><sequence><br><element name="eventID" type="string" minOccurs="1" maxOccurs="1"/><br><choice minOccurs="1" maxOccurs="1"><br><element name="eventType" type="ineruni:eventTypeEnum" minOccurs="1" maxOccurs="1"/><br><element name="topic" type="string" minOccurs="0" maxOccurs="1"/><br><element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="comment" type="string" minOccurs="0" maxOccurs="1"/><br><element name="source" type="string" minOccurs="1" maxOccurs="1"/> |

```
      <element name="project" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="location" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="objectID" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="processID" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="content" minOccurs="1" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="observation" type="string" minOccurs="1" maxOccurs="1"/>
            <element name="observationProperty" type="string" minOccurs="0" maxOccurs="1"/>
            <element name="observablePropertyType" type="string" minOccurs="0" maxOccurs="1"/>
            <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/>
            <element name="observationTimeStamp" type="dateTime" minOccurs="0" maxOccurs="1"/>
          </sequence>
        </complexType>
      </element>
    </choice>
  </sequence>
</complexType>
```

element **tLinkSmartEvent/eventID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="eventID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tLinkSmartEvent/eventType**

| | | | |
|---|---|---|---|
| diagram |  | | |
| type | **ineruni:eventTypeEnum** | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | EURPerkWh | |
| | enumeration | SEKPerkWh | |
| | enumeration | CZKPerkWh | |
| | enumeration | DKKPerkWh | |
| | enumeration | NOKPerkWh | |
| | enumeration | CHFPerkWh | |
| | enumeration | HRKPerkWh | |
| source | <element name="eventType" type="ineruni:eventTypeEnum" minOccurs="1" maxOccurs="1"/> | | |

element **tLinkSmartEvent/topic**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="topic" type="string" minOccurs="0" maxOccurs="1"/> |

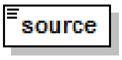
element **tLinkSmartEvent/timeStamp**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/> |

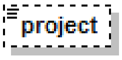
element **tLinkSmartEvent/comment**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="comment" type="string" minOccurs="0" maxOccurs="1"/> |

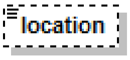
element **tLinkSmartEvent/source**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="source" type="string" minOccurs="1" maxOccurs="1"/> |

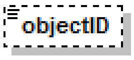
element **tLinkSmartEvent/project**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="project" type="string" minOccurs="0" maxOccurs="1"/> |

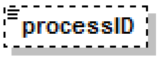
element **tLinkSmartEvent/location**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="location" type="string" minOccurs="0" maxOccurs="1"/> |

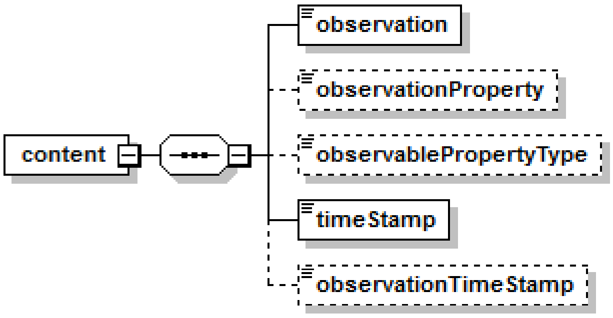
element **tLinkSmartEvent/objectID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="objectID" type="string" minOccurs="0" maxOccurs="1"/> |

element **tLinkSmartEvent/processID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="processID" type="string" minOccurs="0" maxOccurs="1"/> |

element **tLinkSmartEvent/content**

| | |
|---|---|
| diagram |  |
| properties | content complex |
| children | [observation]() [observationProperty]() [observablePropertyType]() [timeStamp]() [observationTimeStamp]() |
| source | <element name="content" minOccurs="1" maxOccurs="1"><br><complexType><br><sequence><br><element name="observation" type="string" minOccurs="1" maxOccurs="1"/><br><element name="observationProperty" type="string" minOccurs="0" maxOccurs="1"/><br><element name="observablePropertyType" type="string" minOccurs="0" maxOccurs="1"/><br><element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="observationTimeStamp" type="dateTime" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

element **tLinkSmartEvent/content/observation**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="observation" type="string" minOccurs="1" maxOccurs="1"/> |

element **tLinkSmartEvent/content/observationProperty**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="observationProperty" type="string" minOccurs="0" maxOccurs="1"/> |

element **tLinkSmartEvent/content/observablePropertyType**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="observablePropertyType" type="string" minOccurs="0" maxOccurs="1"/> |

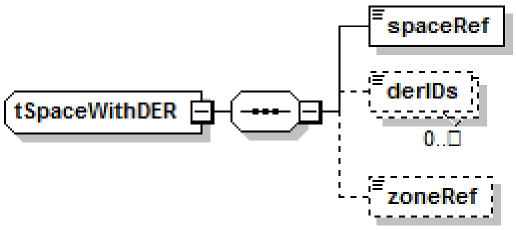
element **tLinkSmartEvent/content/timeStamp**

| | |
|---|---|
| diagram |   |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/> |

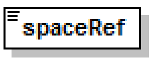
element **tLinkSmartEvent/content/observationTimeStamp**

| | |
|---|---|
| diagram |   |
| type | **dateTime** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="observationTimeStamp" type="dateTime" minOccurs="0" maxOccurs="1"/> |

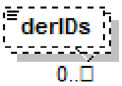
complexType **tSpaceWithDER**

| | |
|---|---|
| diagram |   |
| namespace | http://www.inertia.eu/2013/schema/event/ |
| children | **spaceRef** **derIDs** **zoneRef** |
| used by | element **tSpatialStructure/spaceWithDERs** |
| source | <complexType name="tSpaceWithDER"><br><sequence><br><element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="derIDs" type="string" minOccurs="0" maxOccurs="unbounded"/><br><element name="zoneRef" type="string" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

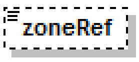
element **tSpaceWithDER/spaceRef**

| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tSpaceWithDER/derIDs**

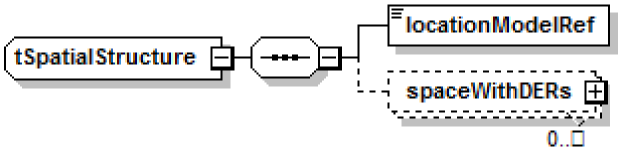
| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="derIDs" type="string" minOccurs="0" maxOccurs="unbounded"/> |

element **tSpaceWithDER/zoneRef**

| | |
|---|---|
| diagram |   |
| type | **string** |

| | |
|---|---|
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="zoneRef" type="string" minOccurs="0" maxOccurs="1"/> |

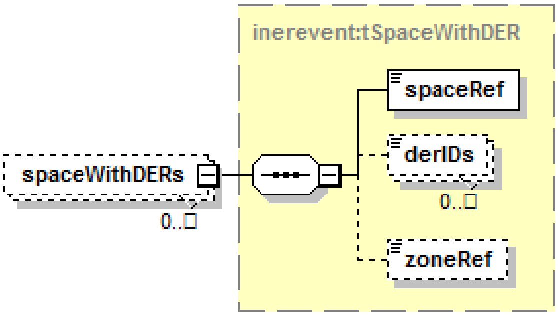
complexType **tSpatialStructure**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/event/ |
| children | **locationModelRef** **spaceWithDERs** |
| source | <complexType name="tSpatialStructure"><br><sequence><br><element name="locationModelRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="spaceWithDERs" type="inerevent:tSpaceWithDER" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tSpatialStructure/locationModelRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="locationModelRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tSpatialStructure/spaceWithDERs**

| | |
|---|---|
| diagram |  |
| type | **inerevent:tSpaceWithDER** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **spaceRef** **derIDs** **zoneRef** |
| source | <element name="spaceWithDERs" type="inerevent:tSpaceWithDER" minOccurs="0" maxOccurs="unbounded"/> |

XML Schema documentation generated by **XMLSpy** Schema Editor **http://www.altova.com/xmlspy**

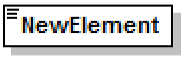
# Schema inertiaUserProfilingandAUI.xsd

| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\workspace\Inertia_CIM_Schema_01\xsd\inertiaUserProfilingandAUI.xsd** |
| attributeFormDefault: | |
| elementFormDefault: | **qualified** |
| targetNamespace: | **http://www.example.org/inertiaUserProfile** |

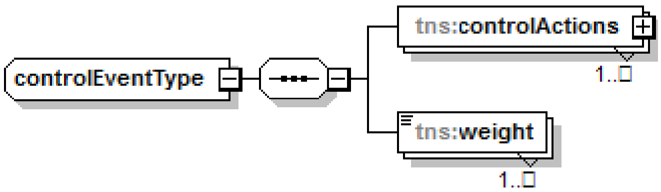
| Elements | Complex types |
|---|---|
| **NewElement** | **controlEventType** |

**electricVehicleType**
**envEventType**
**smartDishwasherType**
**smartRefrigeratorType**
**smartWaterHeaterType**
**tAUIExtPreferences**
**tAUIuserProfile**
**tControlAction**
**tDERoperationalInetervals**
**tGroupOfUserCharacteristics**
**thermalEventType**
**tIndividualUserProfile**
**tOperationalInterval**
**tTariff**
**tUPEvents**
**tUserPreferences**
**tWeightenDyn**
**visualEventType**

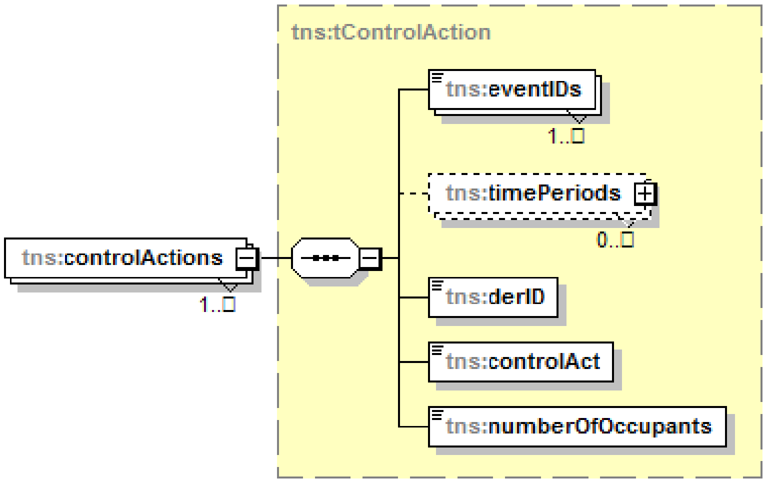
## element **NewElement**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="NewElement" type="string"/> |

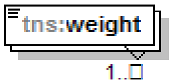
## complexType **controlEventType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:controlActions** **tns:weight** |
| used by | element **tUPEvents/controlEvents** |
| source | <complexType name="controlEventType"><br><sequence><br><element name="controlActions" type="tns:tControlAction" minOccurs="1" maxOccurs="unbounded"/><br><element name="weight" type="float" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType> |

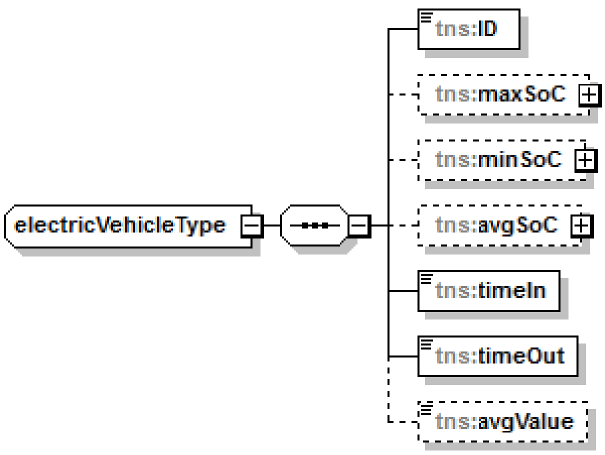
element **controlEventType/controlActions**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tControlAction** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **tns:eventIDs tns:timePeriods tns:derID tns:controlAct tns:numberOfOccupants** |
| source | <element name="controlActions" type="tns:tControlAction" minOccurs="1" maxOccurs="unbounded"/> |

element **controlEventType/weight**

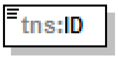
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **float** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="weight" type="float" minOccurs="1" maxOccurs="unbounded"/> |

complexType **electricVehicleType**

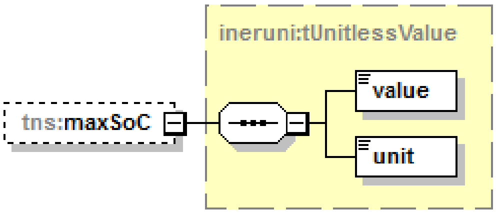
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:ID tns:maxSoC tns:minSoC tns:avgSoC tns:timeIn tns:timeOut tns:avgValue** |
| used by | element **tUserPreferences/userPreferences/operationalPreferences/electricVehicle** |
| source | <complexType name="electricVehicleType"><br><sequence><br><element name="ID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="maxSoC" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

| | |
|---|---|
| | <element name="minSoC" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="avgSoC" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="timeIn" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timeOut" type="time" minOccurs="1" maxOccurs="1"/><br><element name="avgValue" type="float" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

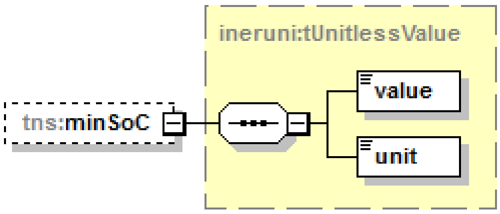
element **electricVehicleType/ID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="ID" type="string" minOccurs="1" maxOccurs="1"/> |

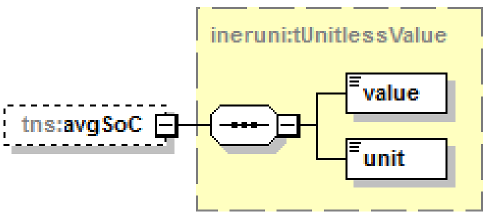
element **electricVehicleType/maxSoC**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="maxSoC" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **electricVehicleType/minSoC**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="minSoC" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **electricVehicleType/avgSoC**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |

| | |
|---|---|
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="avgSoC" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **electricVehicleType/timeIn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="timeIn" type="string" minOccurs="1" maxOccurs="1"/> |

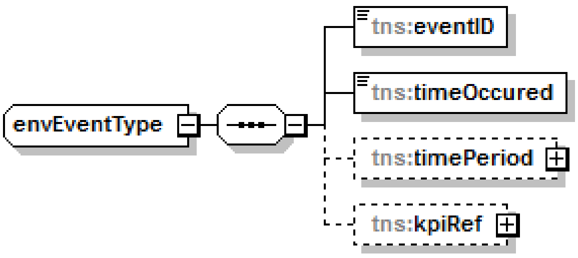
element **electricVehicleType/timeOut**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | content simple |
| source | <element name="timeOut" type="time" minOccurs="1" maxOccurs="1"/> |

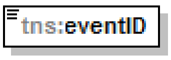
element **electricVehicleType/avgValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **float** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="avgValue" type="float" minOccurs="0" maxOccurs="1"/> |

complexType **envEventType**

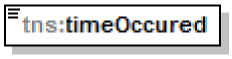
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:eventID tns:timeOccured tns:timePeriod tns:kpiRef** |
| used by | element **tUPEvents/envEvent** |
| source | <complexType name="envEventType"><br><sequence><br><element name="eventID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timeOccured" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/><br><element name="kpiRef" type="inerkpi:tKeyPerformanceIndicator" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

element **envEventType/eventID**

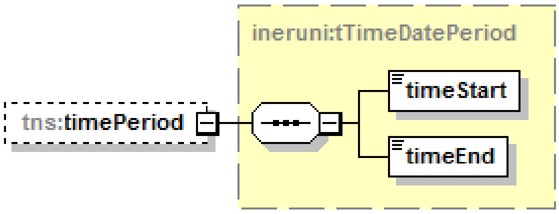
| | |
|---|---|
| diagram |  |

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="eventID" type="string" minOccurs="1" maxOccurs="1"/> |

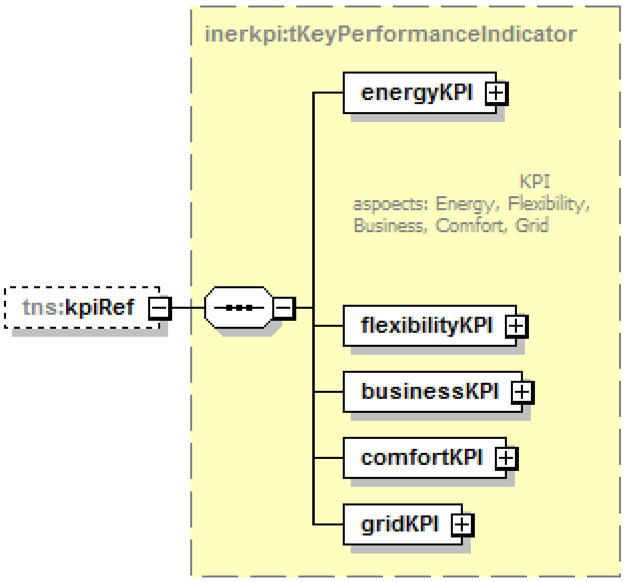
element **envEventType/timeOccured**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="timeOccured" type="string" minOccurs="1" maxOccurs="1"/> |

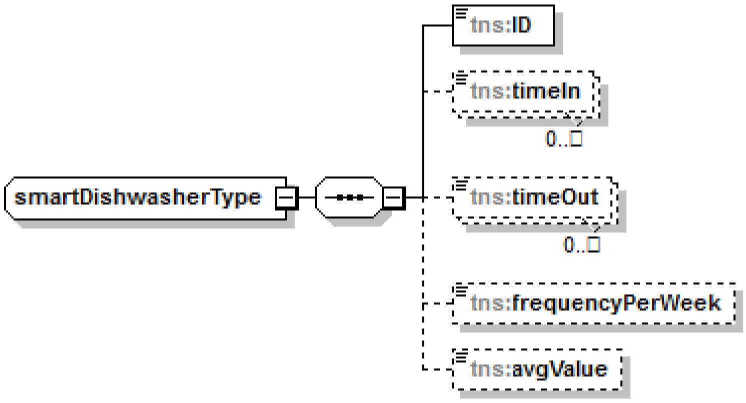
element **envEventType/timePeriod**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/> |

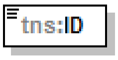
element **envEventType/kpiRef**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **inerkpi:tKeyPerformanceIndicator** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **energyKPI flexibilityKPI businessKPI comfortKPI gridKPI** |
| source | <element name="kpiRef" type="inerkpi:tKeyPerformanceIndicator" minOccurs="0" maxOccurs="1"/> |

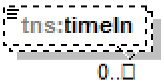
complexType **smartDishwasherType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:ID tns:timeIn tns:timeOut tns:frequencyPerWeek tns:avgValue** |
| used by | elements **tAUIExtPreferences/dishawasher** **tUserPreferences/userPreferences/operationalPreferences/dishwasher** |
| source | <complexType name="smartDishwasherType"><br><sequence><br><element name="ID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timeIn" type="time" minOccurs="0" maxOccurs="unbounded"/><br><element name="timeOut" type="time" minOccurs="0" maxOccurs="unbounded"/><br><element name="frequencyPerWeek" type="int" minOccurs="0" maxOccurs="1"/><br><element name="avgValue" type="float" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

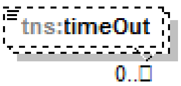
element **smartDishwasherType/ID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="ID" type="string" minOccurs="1" maxOccurs="1"/> |

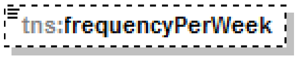
element **smartDishwasherType/timeIn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="timeIn" type="time" minOccurs="0" maxOccurs="unbounded"/> |

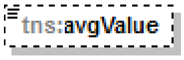
element **smartDishwasherType/timeOut**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="timeOut" type="time" minOccurs="0" maxOccurs="unbounded"/> |

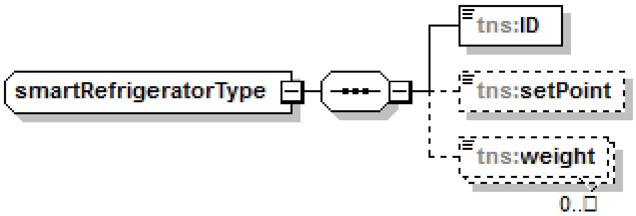
element **smartDishwasherType/frequencyPerWeek**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **int** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="frequencyPerWeek" type="int" minOccurs="0" maxOccurs="1"/> |

element **smartDishwasherType/avgValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **float** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="avgValue" type="float" minOccurs="0" maxOccurs="1"/> |

complexType **smartRefrigeratorType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:ID** **tns:setPoint** **tns:weight** |
| used by | element **tUserPreferences/userPreferences/operationalPreferences/refrigerator** |
| source | <complexType name="smartRefrigeratorType"><br><sequence><br><element name="ID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="setPoint" type="string" minOccurs="0" maxOccurs="1"/><br><element name="weight" type="string" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

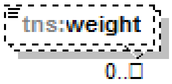
element **smartRefrigeratorType/ID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="ID" type="string" minOccurs="1" maxOccurs="1"/> |

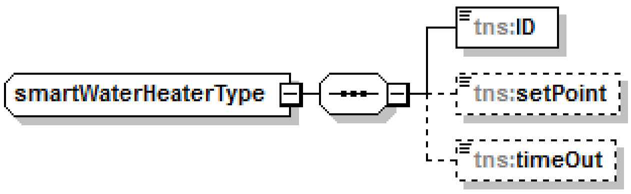
element **smartRefrigeratorType/setPoint**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="setPoint" type="string" minOccurs="0" maxOccurs="1"/> |

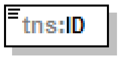
element **smartRefrigeratorType/weight**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="weight" type="string" minOccurs="0" maxOccurs="unbounded"/> |

complexType **smartWaterHeaterType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:ID tns:setPoint tns:timeOut** |
| used by | element **tUserPreferences/userPreferences/operationalPreferences/boiler** |
| source | <complexType name="smartWaterHeaterType"><br><sequence><br><element name="ID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="setPoint" type="float" minOccurs="0" maxOccurs="1"/><br><element name="timeOut" type="time" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

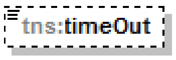
element **smartWaterHeaterType/ID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="ID" type="string" minOccurs="1" maxOccurs="1"/> |

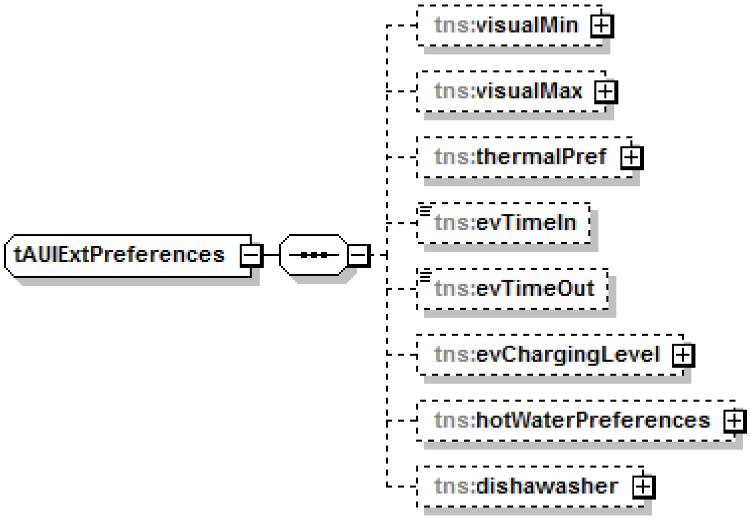
element **smartWaterHeaterType/setPoint**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **float** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="setPoint" type="float" minOccurs="0" maxOccurs="1"/> |

element **smartWaterHeaterType/timeOut**

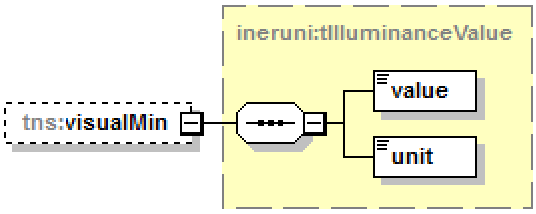
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="timeOut" type="time" minOccurs="0" maxOccurs="1"/> |

## complexType **tAUIExtPreferences**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:visualMin** **tns:visualMax** **tns:thermalPref** **tns:evTimeIn** **tns:evTimeOut** **tns:evChargingLevel** **tns:hotWaterPreferences** **tns:dishawasher** |
| used by | elements **tAUIuserProfile/extPreferences** **tIndividualUserProfile/extPreferencesAUI** |
| source | see below |

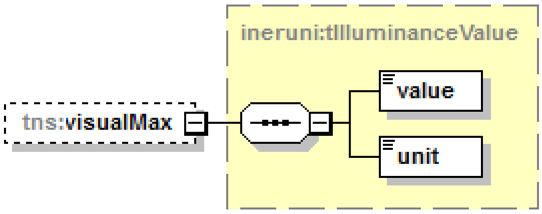
```
<complexType name="tAUIExtPreferences">
  <sequence>
    <element name="visualMin" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
    <element name="visualMax" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
    <element name="thermalPref" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/>
    <element name="evTimeIn" type="time" minOccurs="0" maxOccurs="1"/>
    <element name="evTimeOut" type="time" minOccurs="0" maxOccurs="1"/>
    <element name="evChargingLevel" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="hotWaterPreferences" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/>
    <element name="dishawasher" type="tns:smartDishwasherType" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

## element **tAUIExtPreferences/visualMin**

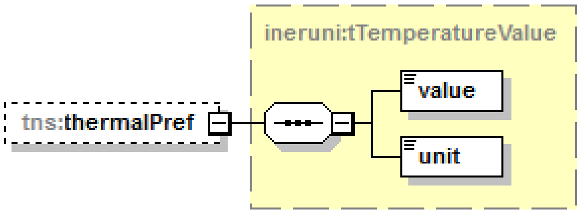
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | see below |

```
<element name="visualMin" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
```

element **tAUIExtPreferences/visualMax**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="visualMax" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

element **tAUIExtPreferences/thermalPref**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTemperatureValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="thermalPref" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/> |

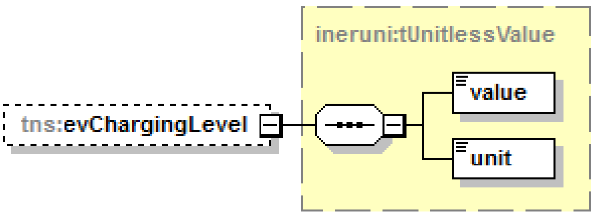
element **tAUIExtPreferences/evTimeIn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="evTimeIn" type="time" minOccurs="0" maxOccurs="1"/> |

element **tAUIExtPreferences/evTimeOut**

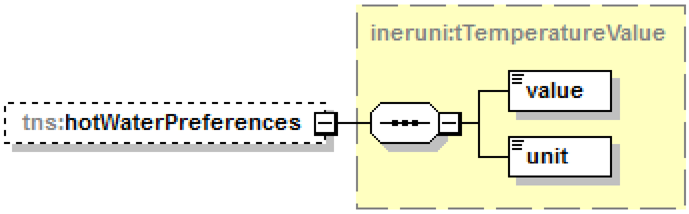
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="evTimeOut" type="time" minOccurs="0" maxOccurs="1"/> |

element **tAUIExtPreferences/evChargingLevel**

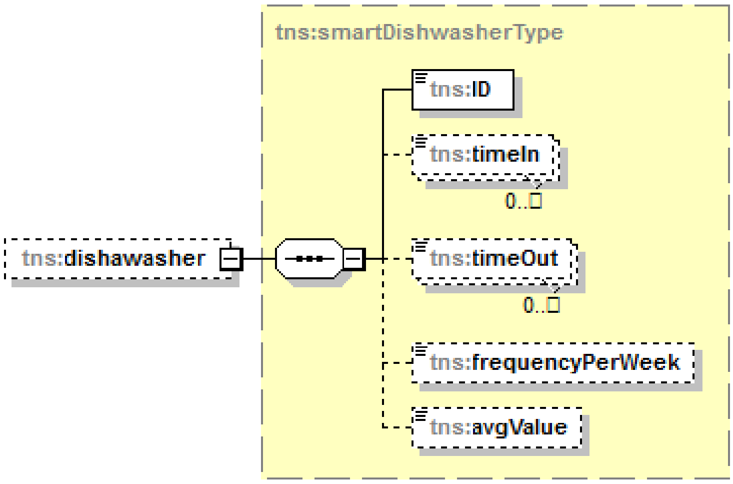
| | |
|---|---|
| diagram |  |

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="evChargingLevel" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tAUIExtPreferences/hotWaterPreferences**

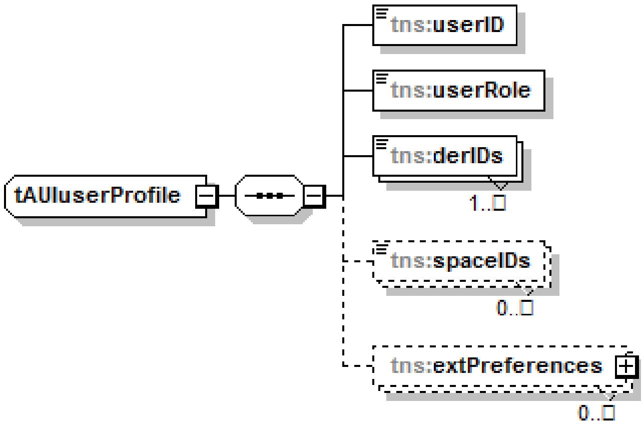
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTemperatureValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="hotWaterPreferences" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/> |

element **tAUIExtPreferences/dishawasher**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:smartDishwasherType** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **tns:ID tns:timeIn tns:timeOut tns:frequencyPerWeek tns:avgValue** |
| source | <element name="dishawasher" type="tns:smartDishwasherType" minOccurs="0" maxOccurs="1"/> |

complexType **tAUIuserProfile**

diagram



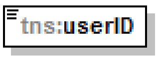

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:userID tns:userRole tns:derIDs tns:spaceIDs tns:extPreferences** |

source

```
<complexType name="tAUIuserProfile">
  <sequence>
    <element name="userID" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="userRole" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="derIDs" type="string" minOccurs="1"
maxOccurs="unbounded"/>
    <element name="spaceIDs" type="string" minOccurs="0"
maxOccurs="unbounded"/>
    <element name="extPreferences" type="tns:tAUIExtPreferences"
minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

element **tAUIuserProfile/userID**

diagram



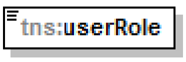

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |

source

```
<element name="userID" type="string" minOccurs="1" maxOccurs="1"/>
```

element **tAUIuserProfile/userRole**

diagram



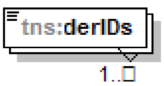

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |

source

```
<element name="userRole" type="string" minOccurs="1" maxOccurs="1"/>
```

element **tAUIuserProfile/derIDs**

diagram

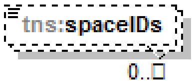



| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |

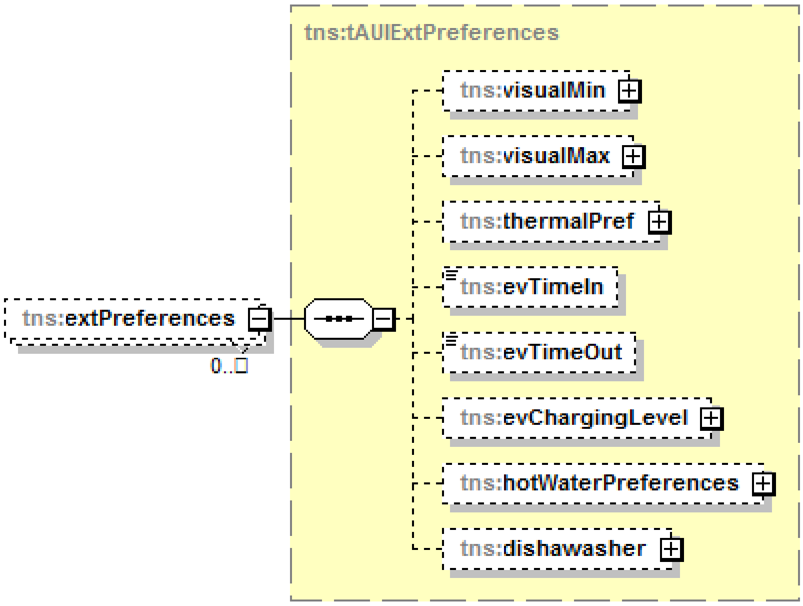
source

```
<element name="derIDs" type="string" minOccurs="1"
maxOccurs="unbounded"/>
```

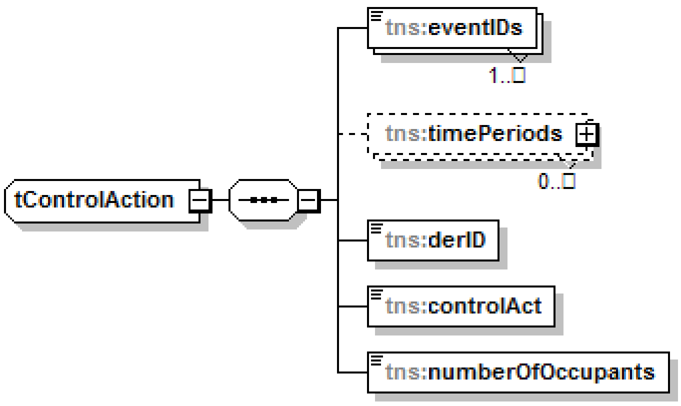
element **tAUIuserProfile/spaceIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="spaceIDs" type="string" minOccurs="0" maxOccurs="unbounded"/> |

element **tAUIuserProfile/extPreferences**

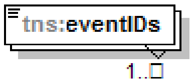
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tAUIExtPreferences** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:visualMin** **tns:visualMax** **tns:thermalPref** **tns:evTimeIn** **tns:evTimeOut** **tns:evChargingLevel** **tns:hotWaterPreferences** **tns:dishawasher** |
| source | <element name="extPreferences" type="tns:tAUIExtPreferences" minOccurs="0" maxOccurs="unbounded"/> |

complexType **tControlAction**

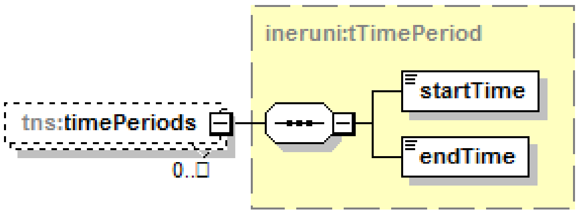
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:eventIDs** **tns:timePeriods** **tns:derID** **tns:controlAct** **tns:numberOfOccupants** |
| used by | elements **tGroupOfUserCharacteristics/controlActions** **tIndividualUserProfile/controlActions** **controlEventType/controlActions** |
| source | <complexType name="tControlAction"> |

| | |
|---|---|
| | <sequence><br><element name="eventIDs" type="string" minOccurs="1" maxOccurs="unbounded"/><br><element name="timePeriods" type="ineruni:tTimePeriod" minOccurs="0" maxOccurs="unbounded"/><br><element name="derID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="controlAct" type="string" minOccurs="1" maxOccurs="1"/><br><element name="numberOfOccupants" type="int" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

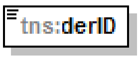
element **tControlAction/eventIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="eventIDs" type="string" minOccurs="1" maxOccurs="unbounded"/> |

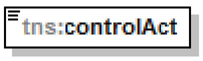
element **tControlAction/timePeriods**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTimePeriod** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **startTime endTime** |
| source | <element name="timePeriods" type="ineruni:tTimePeriod" minOccurs="0" maxOccurs="unbounded"/> |

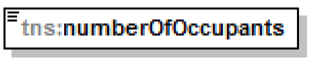
element **tControlAction/derID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="derID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tControlAction/controlAct**

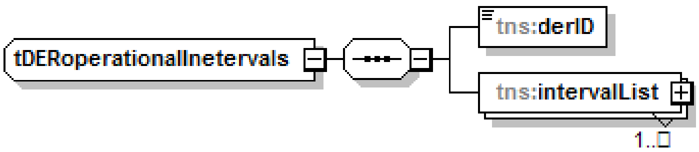
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="controlAct" type="string" minOccurs="1" maxOccurs="1"/> |

element **tControlAction/numberOfOccupants**

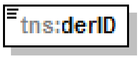
| | |
|---|---|
| diagram |  |

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **int** |
| properties | content simple |
| source | <element name="numberOfOccupants" type="int" minOccurs="1" maxOccurs="1"/> |

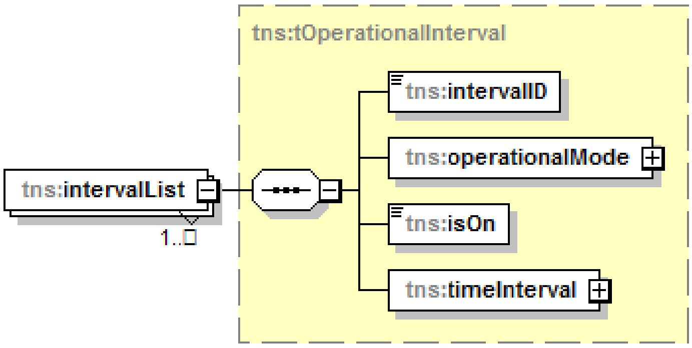
complexType **tDERoperationalInetervals**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:derID** **tns:intervalList** |
| source | <complexType name="tDERoperationalInetervals"><br><sequence><br><element name="derID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="intervalList" type="tns:tOperationalInterval" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tDERoperationalInetervals/derID**

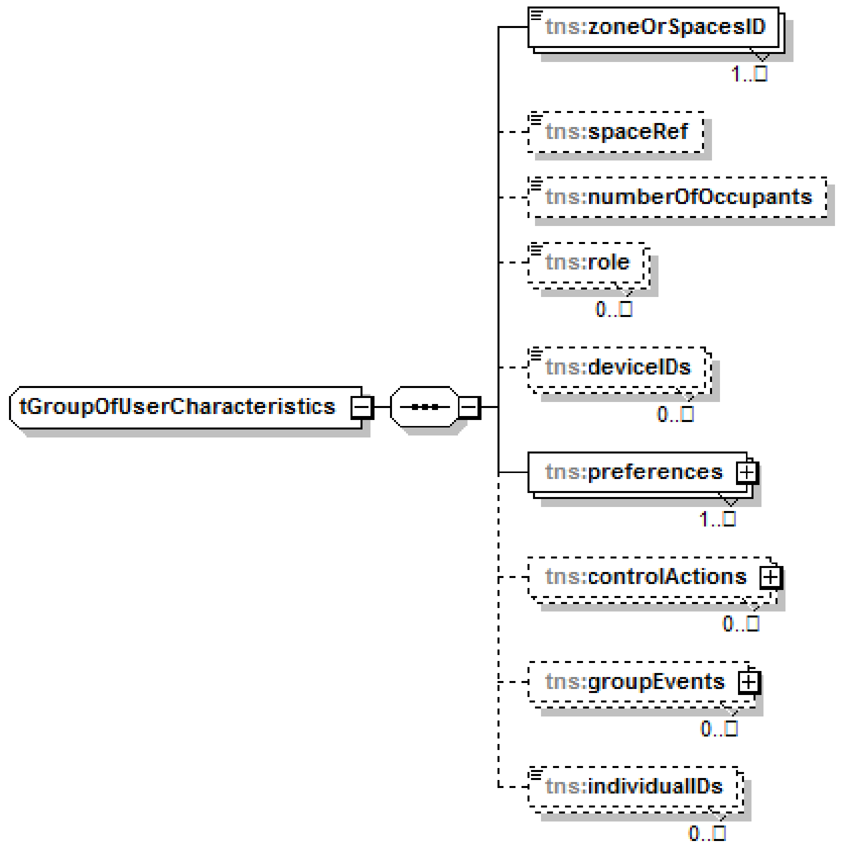
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="derID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tDERoperationalInetervals/intervalList**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tOperationalInterval** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **tns:intervalID** **tns:operationalMode** **tns:isOn** **tns:timeInterval** |
| source | <element name="intervalList" type="tns:tOperationalInterval" minOccurs="1" maxOccurs="unbounded"/> |

complexType **tGroupOfUserCharacteristics**

diagram



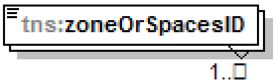

namespace http://www.example.org/inertiaUserProfile

children **tns:zoneOrSpacesID** **tns:spaceRef** **tns:numberOfOccupants** **tns:role** **tns:deviceIDs** **tns:preferences** **tns:controlActions** **tns:groupEvents** **tns:individualIDs**

source

```
<complexType name="tGroupOfUserCharacteristics">
  <sequence>
    <element name="zoneOrSpacesID" type="string" minOccurs="1" maxOccurs="unbounded"/>
    <element name="spaceRef" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="numberOfOccupants" type="int" minOccurs="0" maxOccurs="1"/>
    <element name="role" type="string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="deviceIDs" type="string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="preferences" type="tns:tUserPreferences" minOccurs="1" maxOccurs="unbounded"/>
    <element name="controlActions" type="tns:tControlAction" minOccurs="0" maxOccurs="unbounded"/>
    <element name="groupEvents" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="weight" type="float" minOccurs="0" maxOccurs="1"/>
          <element name="event" type="tns:tUPEvents" minOccurs="1" maxOccurs="1"/>
        </sequence>
      </complexType>
    </element>
    <element name="individualIDs" type="string" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

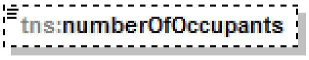
element **tGroupOfUserCharacteristics/zoneOrSpacesID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="zoneOrSpacesID" type="string" minOccurs="1" maxOccurs="unbounded"/> |

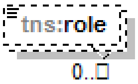
element **tGroupOfUserCharacteristics/spaceRef**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="spaceRef" type="string" minOccurs="0" maxOccurs="1"/> |

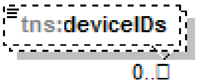
element **tGroupOfUserCharacteristics/numberOfOccupants**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **int** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="numberOfOccupants" type="int" minOccurs="0" maxOccurs="1"/> |

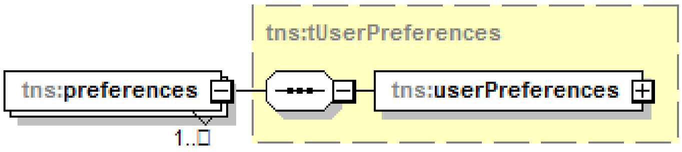
element **tGroupOfUserCharacteristics/role**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="role" type="string" minOccurs="0" maxOccurs="unbounded"/> |

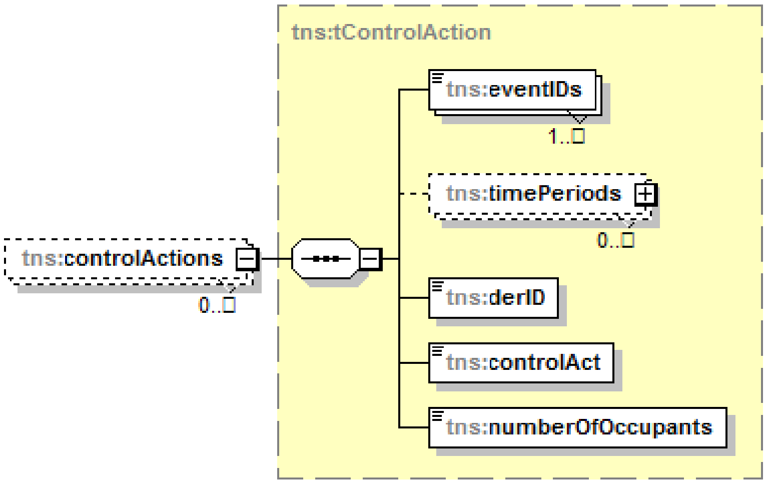
element **tGroupOfUserCharacteristics/deviceIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="deviceIDs" type="string" minOccurs="0" maxOccurs="unbounded"/> |

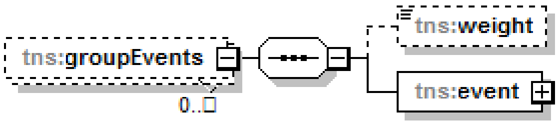
element **tGroupOfUserCharacteristics/preferences**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tUserPreferences** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **tns:userPreferences** |
| source | <element name="preferences" type="tns:tUserPreferences" minOccurs="1" maxOccurs="unbounded"/> |

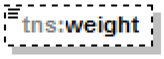
element **tGroupOfUserCharacteristics/controlActions**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tControlAction** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:eventIDs tns:timePeriods tns:derID tns:controlAct tns:numberOfOccupants** |
| source | <element name="controlActions" type="tns:tControlAction" minOccurs="0" maxOccurs="unbounded"/> |

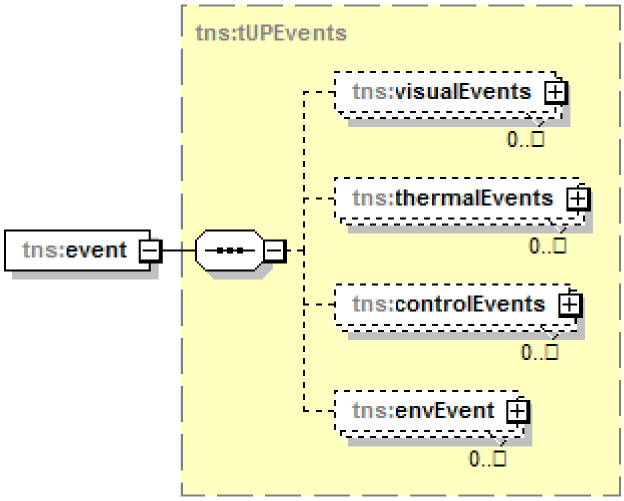
element **tGroupOfUserCharacteristics/groupEvents**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:weight tns:event** |
| source | <element name="groupEvents" minOccurs="0" maxOccurs="unbounded"><br><complexType><br><sequence><br><element name="weight" type="float" minOccurs="0" maxOccurs="1"/><br><element name="event" type="tns:tUPEvents" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

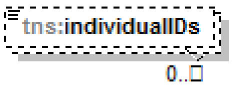
element **tGroupOfUserCharacteristics/groupEvents/weight**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **float** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="weight" type="float" minOccurs="0" maxOccurs="1"/> |

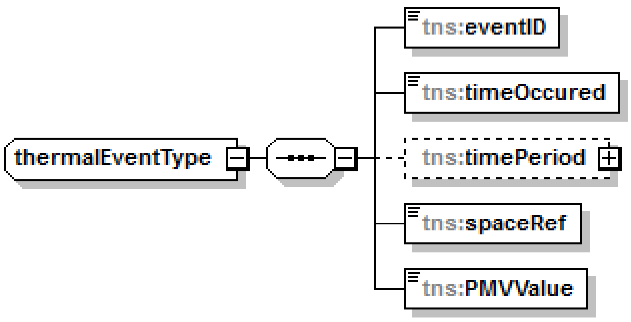
element **tGroupOfUserCharacteristics/groupEvents/event**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tUPEvents** |
| properties | content complex |
| children | **tns:visualEvents** **tns:thermalEvents** **tns:controlEvents** **tns:envEvent** |
| source | <element name="event" type="tns:tUPEvents" minOccurs="1" maxOccurs="1"/> |

element **tGroupOfUserCharacteristics/individualIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="individualIDs" type="string" minOccurs="0" maxOccurs="unbounded"/> |

complexType **thermalEventType**

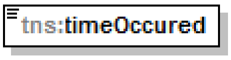
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:eventID** **tns:timeOccured** **tns:timePeriod** **tns:spaceRef** **tns:PMVValue** |
| used by | element **tUPEvents/visualEvents** |

| | |
|---|---|
| source | <complexType name="thermalEventType"><br><sequence><br><element name="eventID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timeOccured" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/><br><element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="PMVValue" type="ineruni:PMVUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

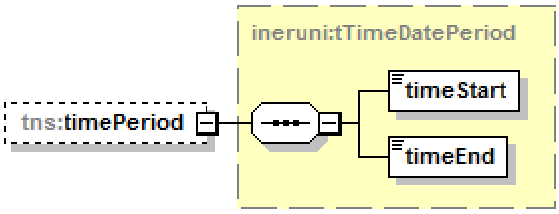
element **thermalEventType/eventID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="eventID" type="string" minOccurs="1" maxOccurs="1"/> |

element **thermalEventType/timeOccured**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeOccured" type="dateTime" minOccurs="1" maxOccurs="1"/> |

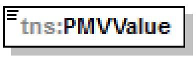
element **thermalEventType/timePeriod**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/> |

element **thermalEventType/spaceRef**

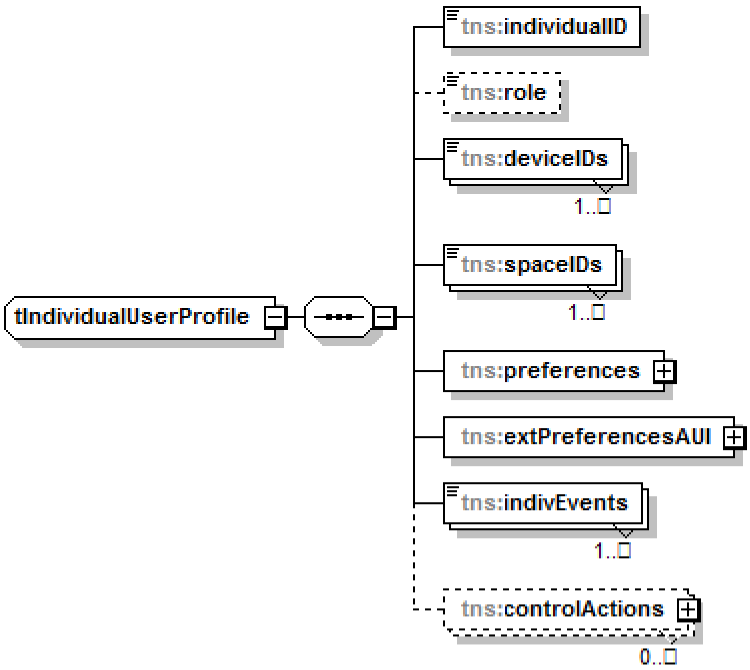
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **thermalEventType/PMVValue**

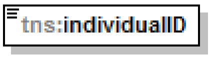
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:PMVUnitEnum** |
| properties | content simple |

| | | | |
|---|---|---|---|
| facets | Kind | Value | Annotation |
| | enumeration | -3 | |
| | enumeration | -2 | |
| | enumeration | -1 | |
| | enumeration | 0 | |
| | enumeration | 1 | |
| | enumeration | 2 | |
| | enumeration | 3 | |
| source | <element name="PMVValue" type="ineruni:PMVUnitEnum" minOccurs="1" maxOccurs="1"/> | | |

complexType **tIndividualUserProfile**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:individualID** **tns:role** **tns:deviceIDs** **tns:spaceIDs** **tns:preferences** **tns:extPreferencesAUI** **tns:indivEvents** **tns:controlActions** |
| source | <complexType name="tIndividualUserProfile"><br><sequence><br><element name="individualID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="role" type="string" minOccurs="0" maxOccurs="1"/><br><element name="deviceIDs" type="string" minOccurs="1" maxOccurs="unbounded"/><br><element name="spaceIDs" type="string" minOccurs="1" maxOccurs="unbounded"/><br><element name="preferences" type="tns:tUserPreferences" minOccurs="1" maxOccurs="1"/><br><element name="extPreferencesAUI" type="tns:tAUIExtPreferences" minOccurs="1" maxOccurs="1"/><br><element name="indivEvents" type="string" minOccurs="1" maxOccurs="unbounded"/><br><element name="controlActions" type="tns:tControlAction" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

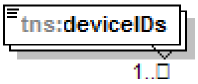
element **tIndividualUserProfile/individualID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="individualID" type="string" minOccurs="1" maxOccurs="1"/> |

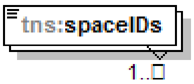
element **tIndividualUserProfile/role**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="role" type="string" minOccurs="0" maxOccurs="1"/> |

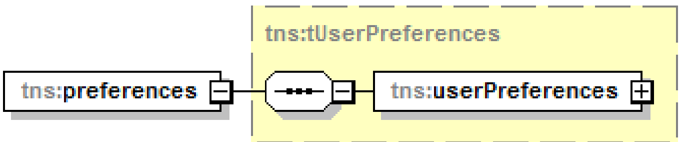
element **tIndividualUserProfile/deviceIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="deviceIDs" type="string" minOccurs="1" maxOccurs="unbounded"/> |

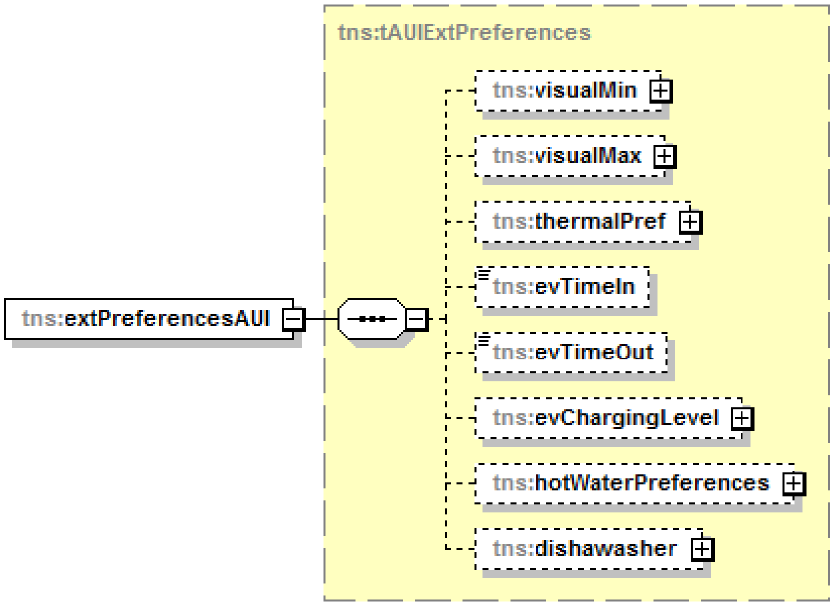
element **tIndividualUserProfile/spaceIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="spaceIDs" type="string" minOccurs="1" maxOccurs="unbounded"/> |

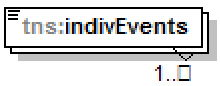
element **tIndividualUserProfile/preferences**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tUserPreferences** |
| properties | content complex |
| children | **tns:userPreferences** |
| source | <element name="preferences" type="tns:tUserPreferences" minOccurs="1" maxOccurs="1"/> |

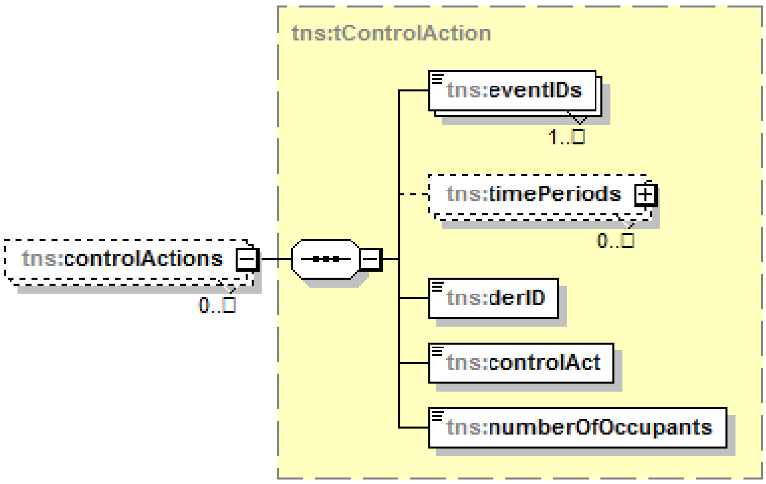
element **tIndividualUserProfile/extPreferencesAUI**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tAUIExtPreferences** |
| properties | content complex |
| children | **tns:visualMin** **tns:visualMax** **tns:thermalPref** **tns:evTimeIn** **tns:evTimeOut** **tns:evChargingLevel** **tns:hotWaterPreferences** **tns:dishawasher** |
| source | <element name="extPreferencesAUI" type="tns:tAUIExtPreferences" minOccurs="1" maxOccurs="1"/> |

element **tIndividualUserProfile/indivEvents**

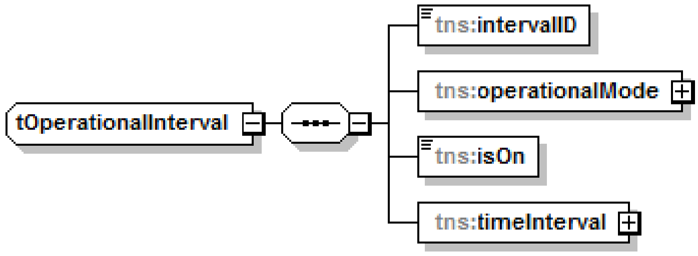
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="indivEvents" type="string" minOccurs="1" maxOccurs="unbounded"/> |

element **tIndividualUserProfile/controlActions**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tControlAction** |
| properties | minOcc 0<br>maxOcc unbounded |

| | |
|---|---|
| | content complex |
| children | **tns:eventIDs** **tns:timePeriods** **tns:derID** **tns:controlAct** **tns:numberOfOccupants** |
| source | <element name="controlActions" type="tns:tControlAction" minOccurs="0" maxOccurs="unbounded"/> |

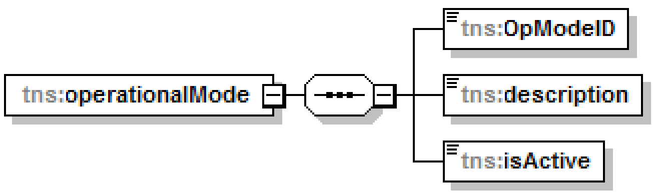
complexType **tOperationalInterval**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:intervalID** **tns:operationalMode** **tns:isOn** **tns:timeInterval** |
| used by | element **tDERoperationalInetervals/intervalList** |
| source | <complexType name="tOperationalInterval"><br><sequence><br><element name="intervalID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="operationalMode" minOccurs="1" maxOccurs="1"><br><complexType><br><sequence><br><element name="OpModeID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="description" type="string" minOccurs="1" maxOccurs="1"/><br><element name="isActive" type="boolean" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType><br></element><br><element name="isOn" type="boolean" minOccurs="1" maxOccurs="1"/><br><element name="timeInterval" type="ineruni:tTimeDatePeriod" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tOperationalInterval/intervalID**

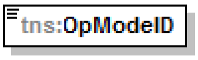
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="intervalID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tOperationalInterval/operationalMode**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| properties | content complex |
| children | **tns:OpModeID** **tns:description** **tns:isActive** |
| source | <element name="operationalMode" minOccurs="1" maxOccurs="1"><br><complexType><br><sequence> |

| | |
|---|---|
| | <element name="OpModeID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="description" type="string" minOccurs="1" maxOccurs="1"/><br><element name="isActive" type="boolean" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

element **tOperationalInterval/operationalMode/OpModeID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="OpModeID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tOperationalInterval/operationalMode/description**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="description" type="string" minOccurs="1" maxOccurs="1"/> |

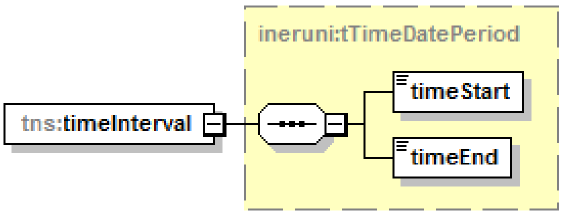
element **tOperationalInterval/operationalMode/isActive**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **boolean** |
| properties | content simple |
| source | <element name="isActive" type="boolean" minOccurs="1" maxOccurs="1"/> |

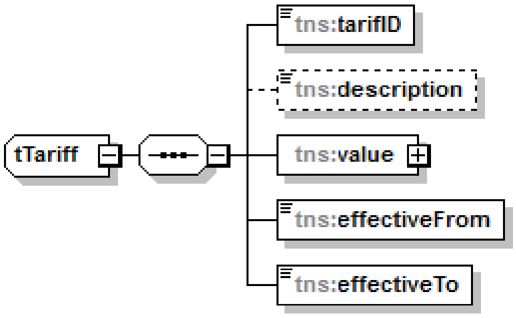
element **tOperationalInterval/isOn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **boolean** |
| properties | content simple |
| source | <element name="isOn" type="boolean" minOccurs="1" maxOccurs="1"/> |

element **tOperationalInterval/timeInterval**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTimeDatePeriod** |
| properties | content complex |
| children | **timeStart timeEnd** |
| source | <element name="timeInterval" type="ineruni:tTimeDatePeriod" minOccurs="1" maxOccurs="1"/> |

## complexType **tTariff**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:tarifID tns:description tns:value tns:effectiveFrom tns:effectiveTo** |
| source | <complexType name="tTariff"><br><sequence><br><element name="tarifID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="description" type="string" minOccurs="0" maxOccurs="1"/><br><element name="value" type="ineruni:tMoneyPerEnergyValue" minOccurs="1" maxOccurs="1"/><br><element name="effectiveFrom" type="time" minOccurs="1" maxOccurs="1"/><br><element name="effectiveTo" type="time" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

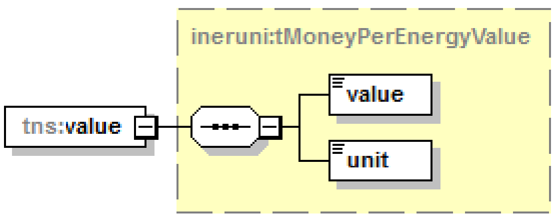
## element **tTariff/tarifID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="tarifID" type="string" minOccurs="1" maxOccurs="1"/> |

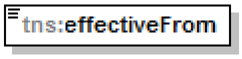
## element **tTariff/description**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="description" type="string" minOccurs="0" maxOccurs="1"/> |

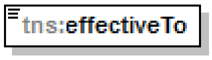
## element **tTariff/value**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="value" type="ineruni:tMoneyPerEnergyValue" minOccurs="1" maxOccurs="1"/> |

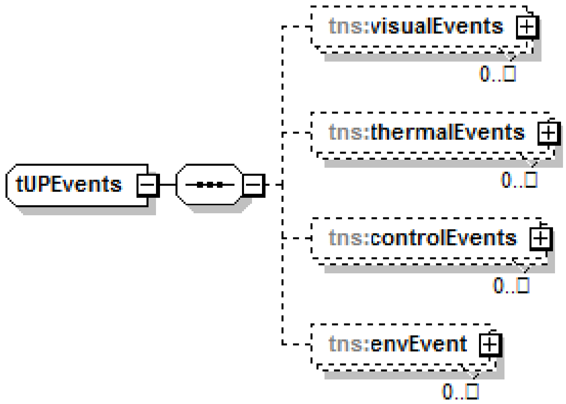
element **tTariff/effectiveFrom**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | content simple |
| source | <element name="effectiveFrom" type="time" minOccurs="1" maxOccurs="1"/> |

element **tTariff/effectiveTo**

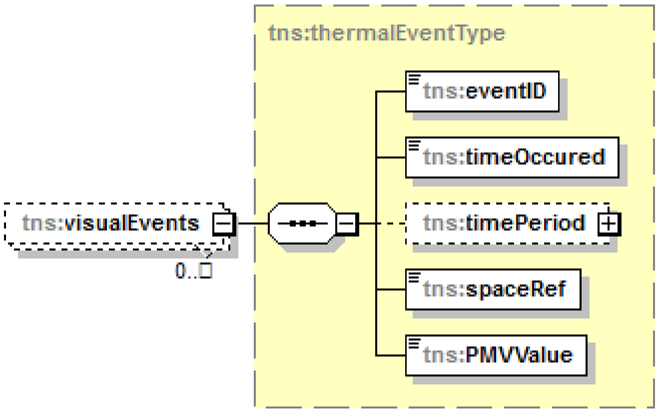
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **time** |
| properties | content simple |
| source | <element name="effectiveTo" type="time" minOccurs="1" maxOccurs="1"/> |

complexType **tUPEvents**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:visualEvents** **tns:thermalEvents** **tns:controlEvents** **tns:envEvent** |
| used by | element **tGroupOfUserCharacteristics/groupEvents/event** |
| source | <complexType name="tUPEvents"><br><sequence><br><element name="visualEvents" type="tns:thermalEventType" minOccurs="0" maxOccurs="unbounded"/><br><element name="thermalEvents" type="tns:visuallEventType" minOccurs="0" maxOccurs="unbounded"/><br><element name="controlEvents" type="tns:controlEventType" minOccurs="0" maxOccurs="unbounded"/><br><element name="envEvent" type="tns:envEventType" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tUPEvents/visualEvents**

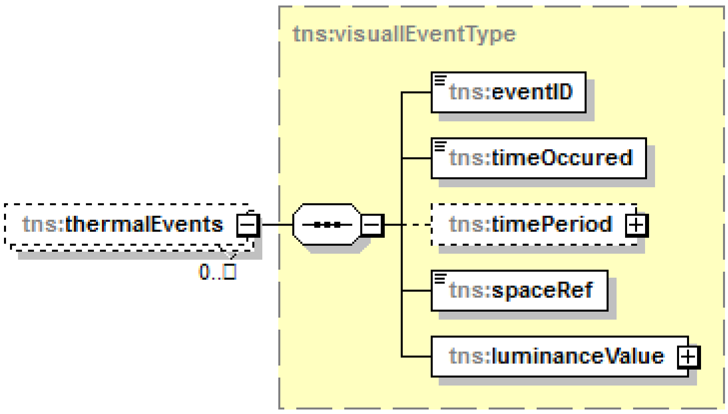
diagram




| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:thermalEventType** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:eventID** **tns:timeOccured** **tns:timePeriod** **tns:spaceRef** **tns:PMVValue** |
| source | <element name="visualEvents" type="tns:thermalEventType" minOccurs="0" maxOccurs="unbounded"/> |

element **tUPEvents/thermalEvents**

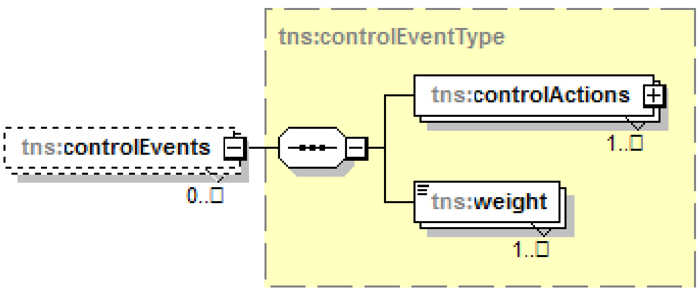
diagram




| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:visuallEventType** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:eventID** **tns:timeOccured** **tns:timePeriod** **tns:spaceRef** **tns:luminanceValue** |
| source | <element name="thermalEvents" type="tns:visuallEventType" minOccurs="0" maxOccurs="unbounded"/> |

element **tUPEvents/controlEvents**

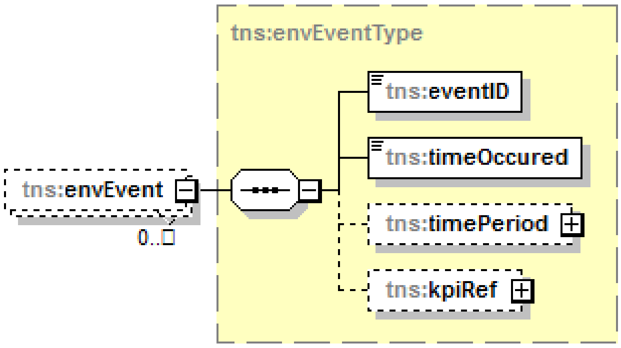
diagram



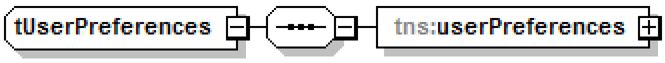

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:controlEventType** |
| properties | minOcc 0<br>maxOcc unbounded |

| | |
|---|---|
| | content complex |
| children | **tns:controlActions** **tns:weight** |
| source | `<element name="controlEvents" type="tns:controlEventType" minOccurs="0" maxOccurs="unbounded"/>` |

element **tUPEvents/envEvent**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:envEventType** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:eventID** **tns:timeOccured** **tns:timePeriod** **tns:kpiRef** |
| source | `<element name="envEvent" type="tns:envEventType" minOccurs="0" maxOccurs="unbounded"/>` |

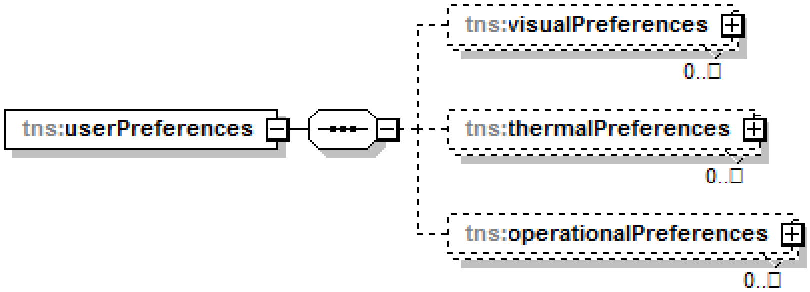
complexType **tUserPreferences**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:userPreferences** |
| used by | elements **tGroupOfUserCharacteristics/preferences** **tIndividualUserProfile/preferences** |
| source | (see below) |

```
<complexType name="tUserPreferences">
  <sequence>
    <element name="userPreferences" minOccurs="1" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="visualPreferences" minOccurs="0" maxOccurs="unbounded">
            <complexType>
              <sequence>
                <element name="maxLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
                <element name="minLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
                <element name="currentLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
                <element name="avgLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
                <element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/>
              </sequence>
            </complexType>
          </element>
          <element name="thermalPreferences" minOccurs="0" maxOccurs="unbounded">
            <complexType>
              <sequence>
                <element name="maxPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/>
                <element name="minPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/>
```

```
              <element name="currentPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/>
              <element name="avgPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/>
              <element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/>
            </sequence>
          </complexType>
        </element>
        <element name="operationalPreferences" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="refrigerator" type="tns:smartRefrigeratorType" minOccurs="0" maxOccurs="1"/>
              <element name="dishwasher" type="tns:smartDishwasherType" minOccurs="0" maxOccurs="1"/>
              <element name="electricVehicle" type="tns:electricVehicleType" minOccurs="0" maxOccurs="1"/>
              <element name="boiler" type="tns:smartWaterHeaterType" minOccurs="0" maxOccurs="1"/>
              <element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
</sequence>
</complexType>
```

element **tUserPreferences/userPreferences**

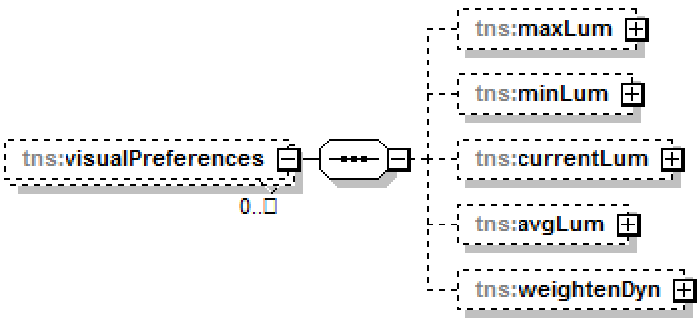
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| properties | content complex |
| children | **tns:visualPreferences tns:thermalPreferences tns:operationalPreferences** |
| source | (see below) |

```
<element name="userPreferences" minOccurs="1" maxOccurs="1">
  <complexType>
    <sequence>
      <element name="visualPreferences" minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="maxLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
            <element name="minLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
            <element name="currentLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
            <element name="avgLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
            <element name="weightenDyn" type="tns:tWeightenDyn"
```

```
minOccurs="0" maxOccurs="1"/>
          </sequence>
        </complexType>
      </element>
      <element name="thermalPreferences" minOccurs="0"
maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="maxPMV" type="ineruni:PMVUnitEnum"
minOccurs="0" maxOccurs="1"/>
            <element name="minPMV" type="ineruni:PMVUnitEnum"
minOccurs="0" maxOccurs="1"/>
            <element name="currentPMV" type="ineruni:PMVUnitEnum"
minOccurs="0" maxOccurs="1"/>
            <element name="avgPMV" type="ineruni:PMVUnitEnum"
minOccurs="0" maxOccurs="1"/>
            <element name="weightenDyn" type="tns:tWeightenDyn"
minOccurs="0" maxOccurs="1"/>
          </sequence>
        </complexType>
      </element>
      <element name="operationalPreferences" minOccurs="0"
maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="refrigerator" type="tns:smartRefrigeratorType"
minOccurs="0" maxOccurs="1"/>
            <element name="dishwasher" type="tns:smartDishwasherType"
minOccurs="0" maxOccurs="1"/>
            <element name="electricVehicle" type="tns:electricVehicleType"
minOccurs="0" maxOccurs="1"/>
            <element name="boiler" type="tns:smartWaterHeaterType"
minOccurs="0" maxOccurs="1"/>
            <element name="weightenDyn" type="tns:tWeightenDyn"
minOccurs="0" maxOccurs="1"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
```

element **tUserPreferences/userPreferences/visualPreferences**

diagram



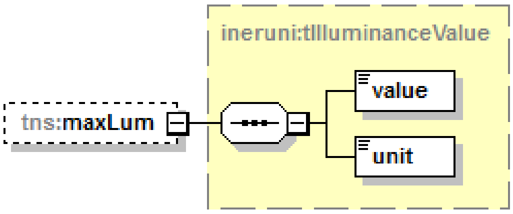

| | |
|---|---|
| namespace | http://www.example.org/inertiaUserProfile |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:maxLum tns:minLum tns:currentLum tns:avgLum tns:weightenDyn** |

source

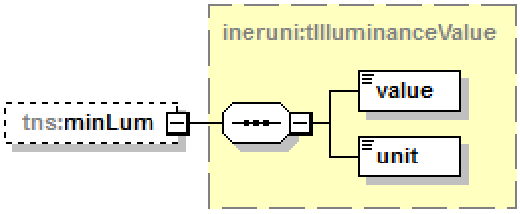
```
<element name="visualPreferences" minOccurs="0"
maxOccurs="unbounded">
  <complexType>
```

| | |
|---|---|
| | <sequence><br><element name="maxLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/><br><element name="minLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/><br><element name="currentLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/><br><element name="avgLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/><br><element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

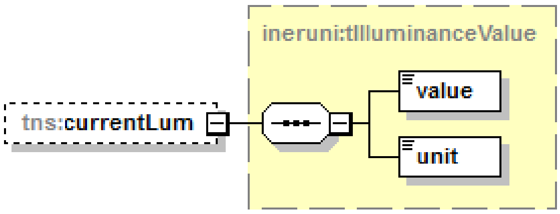
element **tUserPreferences/userPreferences/visualPreferences/maxLum**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="maxLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

element **tUserPreferences/userPreferences/visualPreferences/minLum**

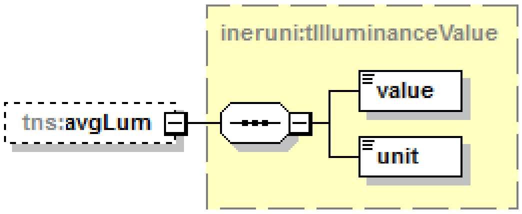
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="minLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

element **tUserPreferences/userPreferences/visualPreferences/currentLum**

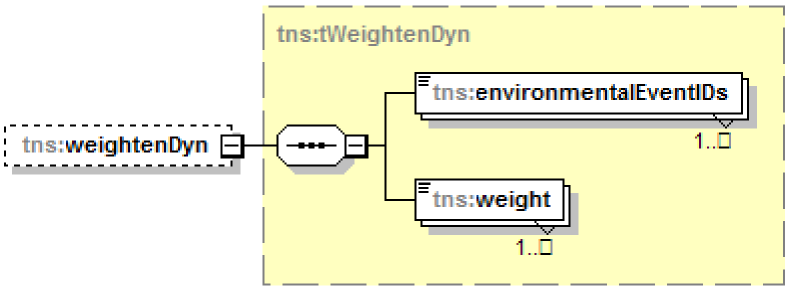
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |

| | |
|---|---|
| children | **value unit** |
| source | <element name="currentLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

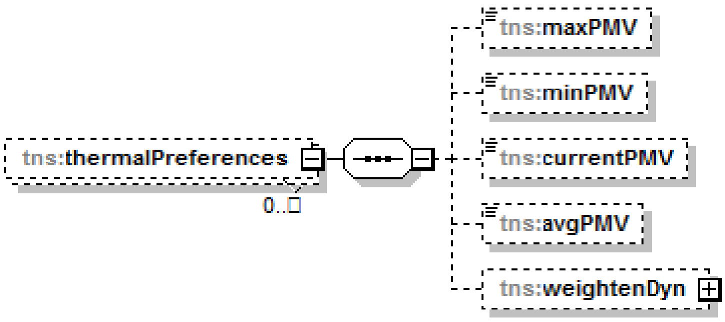
element **tUserPreferences/userPreferences/visualPreferences/avgLum**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="avgLum" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

element **tUserPreferences/userPreferences/visualPreferences/weightenDyn**

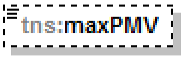
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tWeightenDyn** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **tns:environmentalEventIDs tns:weight** |
| source | <element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/> |

element **tUserPreferences/userPreferences/thermalPreferences**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:maxPMV tns:minPMV tns:currentPMV tns:avgPMV tns:weightenDyn** |
| source | <element name="thermalPreferences" minOccurs="0" maxOccurs="unbounded"><br><complexType><br><sequence><br><element name="maxPMV" type="ineruni:PMVUnitEnum" minOccurs="0" |

| | |
|---|---|
| | maxOccurs="1"/><br><element name="minPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/><br><element name="currentPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/><br><element name="avgPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/><br><element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

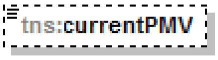
element **tUserPreferences/userPreferences/thermalPreferences/maxPMV**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:PMVUnitEnum** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| facets | Kind / Value / Annotation<br>enumeration -3<br>enumeration -2<br>enumeration -1<br>enumeration 0<br>enumeration 1<br>enumeration 2<br>enumeration 3 |
| source | <element name="maxPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/> |

element **tUserPreferences/userPreferences/thermalPreferences/minPMV**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:PMVUnitEnum** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| facets | Kind / Value / Annotation<br>enumeration -3<br>enumeration -2<br>enumeration -1<br>enumeration 0<br>enumeration 1<br>enumeration 2<br>enumeration 3 |
| source | <element name="minPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/> |

element **tUserPreferences/userPreferences/thermalPreferences/currentPMV**

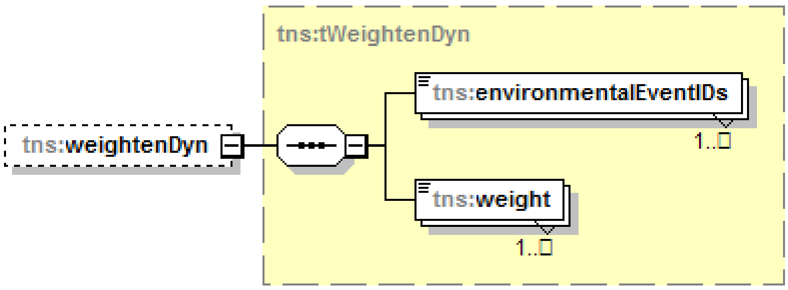
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:PMVUnitEnum** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| facets | Kind / Value / Annotation |

| | | |
|---|---|---|
| | enumeration | -3 |
| | enumeration | -2 |
| | enumeration | -1 |
| | enumeration | 0 |
| | enumeration | 1 |
| | enumeration | 2 |
| | enumeration | 3 |
| source | <element name="currentPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/> | |

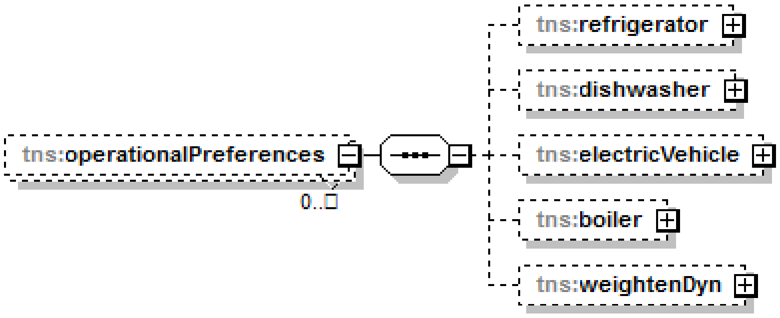
element **tUserPreferences/userPreferences/thermalPreferences/avgPMV**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:PMVUnitEnum** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| facets | Kind Value Annotation<br>enumeration -3<br>enumeration -2<br>enumeration -1<br>enumeration 0<br>enumeration 1<br>enumeration 2<br>enumeration 3 |
| source | <element name="avgPMV" type="ineruni:PMVUnitEnum" minOccurs="0" maxOccurs="1"/> |

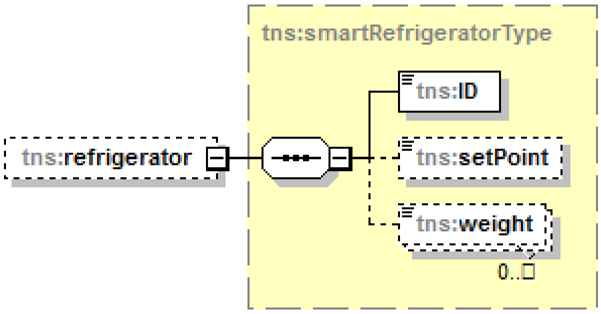
element **tUserPreferences/userPreferences/thermalPreferences/weightenDyn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tWeightenDyn** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **tns:environmentalEventIDs** **tns:weight** |
| source | <element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/> |

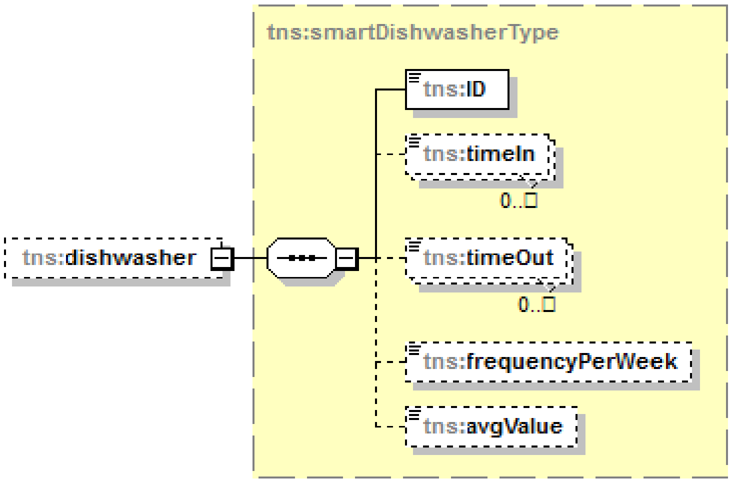
element **tUserPreferences/userPreferences/operationalPreferences**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **tns:refrigerator tns:dishwasher tns:electricVehicle tns:boiler tns:weightenDyn** |
| source | <element name="operationalPreferences" minOccurs="0" maxOccurs="unbounded"><br><complexType><br><sequence><br><element name="refrigerator" type="tns:smartRefrigeratorType" minOccurs="0" maxOccurs="1"/><br><element name="dishwasher" type="tns:smartDishwasherType" minOccurs="0" maxOccurs="1"/><br><element name="electricVehicle" type="tns:electricVehicleType" minOccurs="0" maxOccurs="1"/><br><element name="boiler" type="tns:smartWaterHeaterType" minOccurs="0" maxOccurs="1"/><br><element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

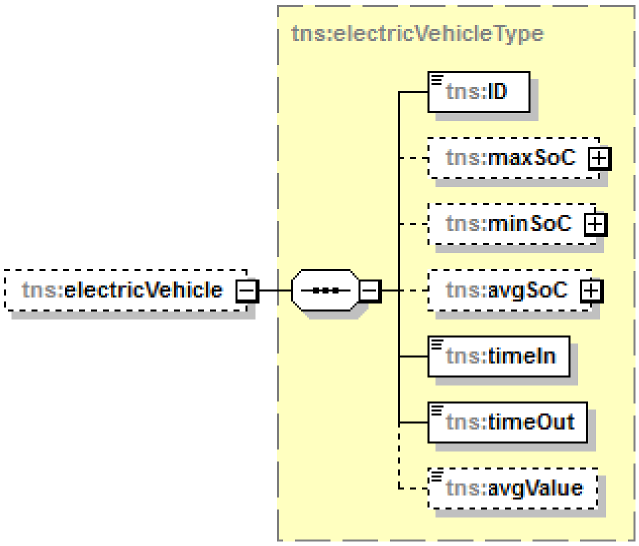
element **tUserPreferences/userPreferences/operationalPreferences/refrigerator**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:smartRefrigeratorType** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **tns:ID tns:setPoint tns:weight** |
| source | <element name="refrigerator" type="tns:smartRefrigeratorType" minOccurs="0" maxOccurs="1"/> |

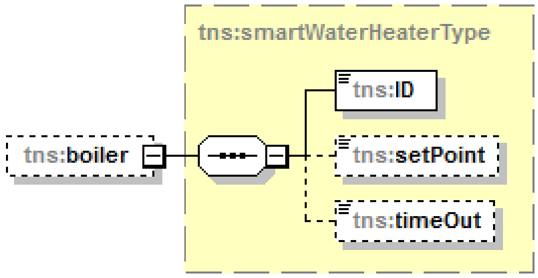
element **tUserPreferences/userPreferences/operationalPreferences/dishwasher**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.example.org/inertiaUserProfile</td></tr>
<tr><td>type</td><td>tns:smartDishwasherType</td></tr>
<tr><td>properties</td><td>minOcc 0<br>maxOcc 1<br>content complex</td></tr>
<tr><td>children</td><td>tns:ID tns:timeIn tns:timeOut tns:frequencyPerWeek tns:avgValue</td></tr>
<tr><td>source</td><td><element name="dishwasher" type="tns:smartDishwasherType" minOccurs="0" maxOccurs="1"/></td></tr>
</table>

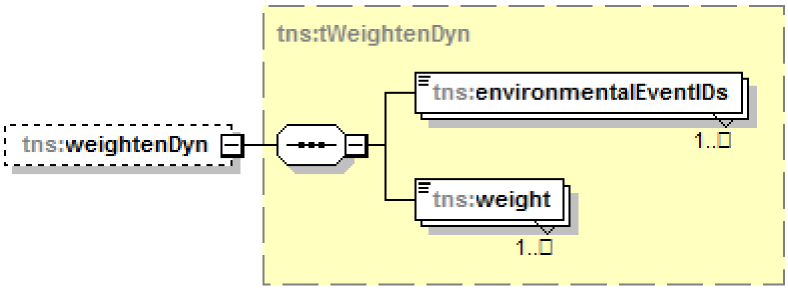
element **tUserPreferences/userPreferences/operationalPreferences/electricVehicle**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.example.org/inertiaUserProfile</td></tr>
<tr><td>type</td><td>tns:electricVehicleType</td></tr>
<tr><td>properties</td><td>minOcc 0<br>maxOcc 1<br>content complex</td></tr>
<tr><td>children</td><td>tns:ID tns:maxSoC tns:minSoC tns:avgSoC tns:timeIn tns:timeOut tns:avgValue</td></tr>
<tr><td>source</td><td><element name="electricVehicle" type="tns:electricVehicleType" minOccurs="0" maxOccurs="1"/></td></tr>
</table>

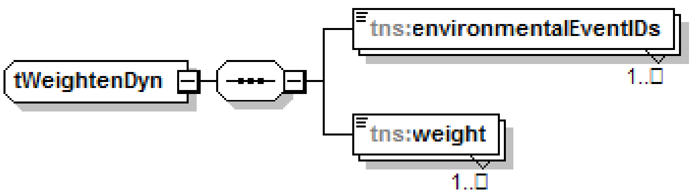
element **tUserPreferences/userPreferences/operationalPreferences/boiler**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:smartWaterHeaterType** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **tns:ID** **tns:setPoint** **tns:timeOut** |
| source | <element name="boiler" type="tns:smartWaterHeaterType" minOccurs="0" maxOccurs="1"/> |

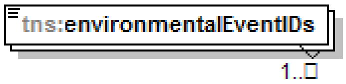
element **tUserPreferences/userPreferences/operationalPreferences/weightenDyn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **tns:tWeightenDyn** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **tns:environmentalEventIDs** **tns:weight** |
| source | <element name="weightenDyn" type="tns:tWeightenDyn" minOccurs="0" maxOccurs="1"/> |

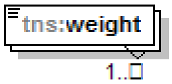
complexType **tWeightenDyn**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:environmentalEventIDs** **tns:weight** |
| used by | elements **tUserPreferences/userPreferences/visualPreferences/weightenDyn**<br>**tUserPreferences/userPreferences/thermalPreferences/weightenDyn**<br>**tUserPreferences/userPreferences/operationalPreferences/weightenDyn** |
| source | <complexType name="tWeightenDyn"><br><sequence><br><element name="environmentalEventIDs" type="string" minOccurs="1" maxOccurs="unbounded"/><br><element name="weight" type="float" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType> |

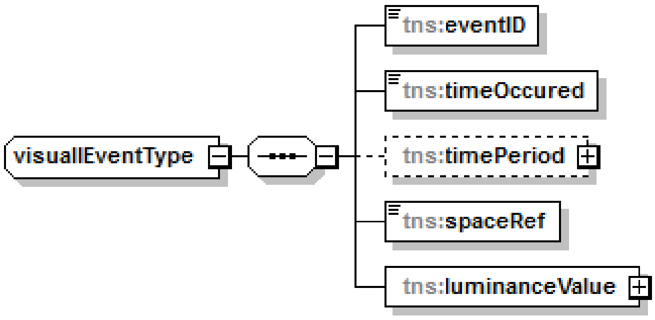
element **tWeightenDyn/environmentalEventIDs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="environmentalEventIDs" type="string" minOccurs="1" maxOccurs="unbounded"/> |

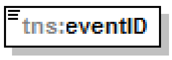
element **tWeightenDyn/weight**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **float** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="weight" type="float" minOccurs="1" maxOccurs="unbounded"/> |

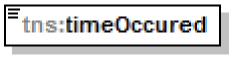
complexType **visualEventType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| children | **tns:eventID tns:timeOccured tns:timePeriod tns:spaceRef tns:luminanceValue** |
| used by | element **tUPEvents/thermalEvents** |
| source | <complexType name="visualEventType"><br><sequence><br><element name="eventID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timeOccured" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/><br><element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="luminanceValue" type="ineruni:tIlluminanceValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

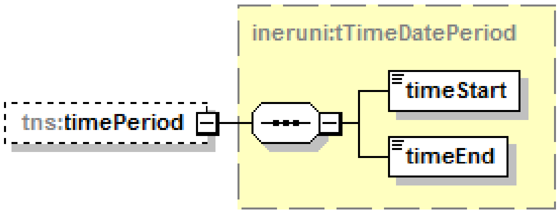
element **visualEventType/eventID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="eventID" type="string" minOccurs="1" maxOccurs="1"/> |

element **visualEventType/timeOccured**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeOccured" type="dateTime" minOccurs="1" maxOccurs="1"/> |

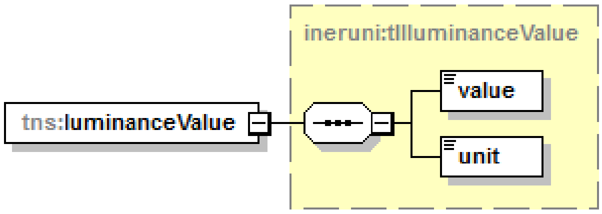
element **visualEventType/timePeriod**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/> |

element **visualEventType/spaceRef**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **string** |
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **visualEventType/luminanceValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaUserProfile |
| type | **ineruni:tIlluminanceValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="luminanceValue" type="ineruni:tIlluminanceValue" minOccurs="1" maxOccurs="1"/> |

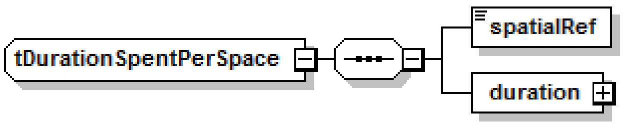
XML Schema documentation generated by **XMLSpy** Schema Editor **http://www.altova.com/xmlspy**

# Schema inertiaOccupancyAndFlow.xsd

| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaOccupancyAndFlow.xsd** |
| attributeFormDefault: | **unqualified** |
| elementFormDefault: | **unqualified** |
| targetNamespace: | **http://www.inertia.eu/2013/schema/occupancy/** |

Complex types
tDurationSpentPerSpace
tDynamiclExtraSchedulingParameters
tFlowIndividual
tFlowOverall
tLocationTransitionMatrix
tOccupancyIndividual
tOccupancyOverall
tOccupancyPrediction
tOccupancyTransitionMatrix
tOccupant
tOccupantLocationPredictions
tOccupantLocationPresence
tOccupantSchedule
tOpenReferenceOccupancyModel
tProportionOfTimePerSpace
tScheduledEvent
tSojournTime
tSpaceSchedule
tTemporalDistributionType
tTemporalDistributionWithPattern
tTemporalType
tTransitionCouple
tTypicalExtraSchedulingParameters

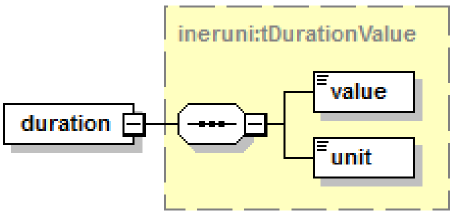
complexType **tDurationSpentPerSpace**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | spatialRef duration |
| used by | element tSojournTime/meanSojournTimePerSpace |
| source | <complexType name="tDurationSpentPerSpace"><br><sequence><br><element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="duration" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

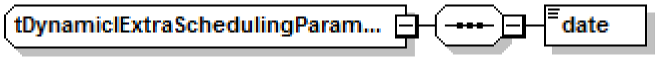
element **tDurationSpentPerSpace/spatialRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tDurationSpentPerSpace/duration**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tDurationValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="duration" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/> |

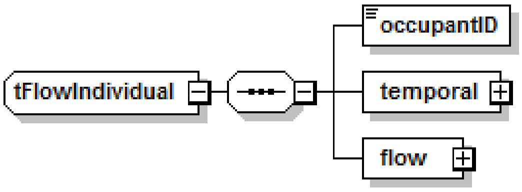
complexType **tDynamicIExtraSchedulingParameters**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **date** |
| used by | elements **tScheduledEvent/extraDynamicDescription** **tOccupantSchedule/extraDynamicDescription** |
| source | <complexType name="tDynamicIExtraSchedulingParameters"><br><sequence><br><element name="date" type="date" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

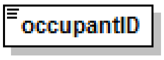
element **tDynamicIExtraSchedulingParameters/date**

| | |
|---|---|
| diagram |  |
| type | **date** |
| properties | content simple |
| source | <element name="date" type="date" minOccurs="1" maxOccurs="1"/> |

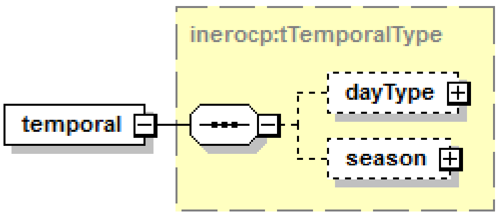
complexType **tFlowIndividual**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **occupantID** **temporal** **flow** |
| source | <complexType name="tFlowIndividual"><br><sequence><br><element name="occupantID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="temporal" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/><br><element name="flow" type="inerocp:tLocationTransitionMatrix" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

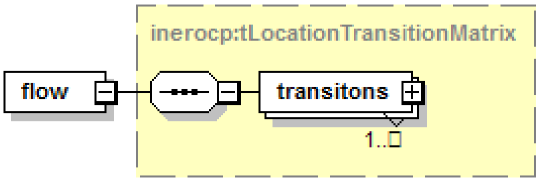
element **tFlowIndividual/occupantID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="occupantID" type="string" minOccurs="1" maxOccurs="1"/> |

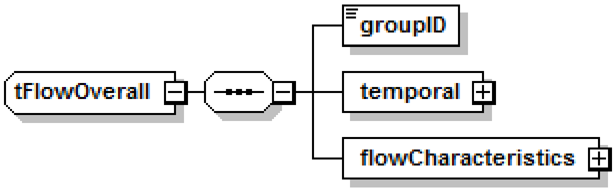
element **tFlowIndividual/temporal**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tTemporalType** |
| properties | content complex |
| children | **dayType** **season** |
| source | <element name="temporal" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/> |

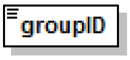
element **tFlowIndividual/flow**

| | |
|---|---|
| diagram |   |
| type | **inerocp:tLocationTransitionMatrix** |
| properties | content complex |
| children | **transitons** |
| source | <element name="flow" type="inerocp:tLocationTransitionMatrix" minOccurs="1" maxOccurs="1"/> |

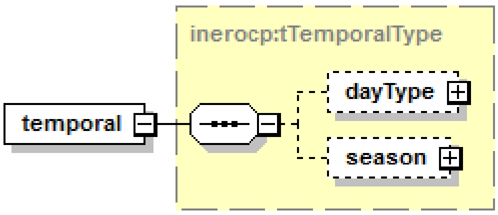
complexType **tFlowOverall**

| | |
|---|---|
| diagram |   |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **groupID temporal flowCharacteristics** |
| source | <complexType name="tFlowOverall"><br><sequence><br><element name="groupID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="temporal" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/><br><element name="flowCharacteristics" type="inerocp:tOccupancyTransitionMatrix" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

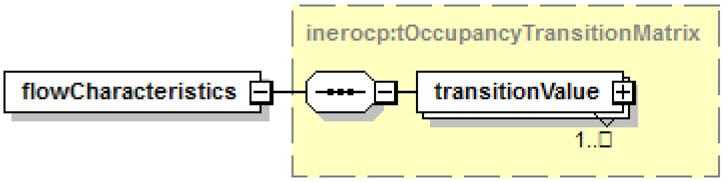
element **tFlowOverall/groupID**

| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | content simple |
| source | <element name="groupID" type="string" minOccurs="1" maxOccurs="1"/> |

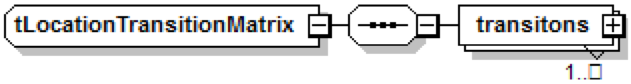
element **tFlowOverall/temporal**

| | |
|---|---|
| diagram |   |
| type | **inerocp:tTemporalType** |
| properties | content complex |
| children | **dayType season** |
| source | <element name="temporal" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/> |

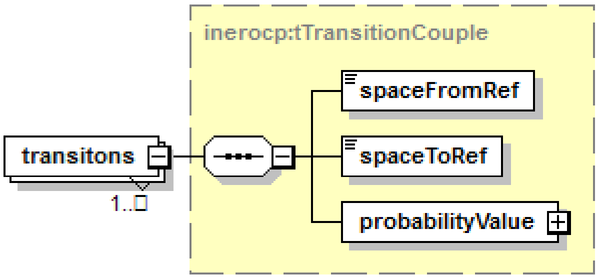
element **tFlowOverall/flowCharacteristics**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><u>**inerocp:tOccupancyTransitionMatrix**</u></td></tr>
<tr><td>properties</td><td>content complex</td></tr>
<tr><td>children</td><td><u>**transitionValue**</u></td></tr>
<tr><td>source</td><td><element name="flowCharacteristics" type="inerocp:tOccupancyTransitionMatrix" minOccurs="1" maxOccurs="1"/></td></tr>
</table>

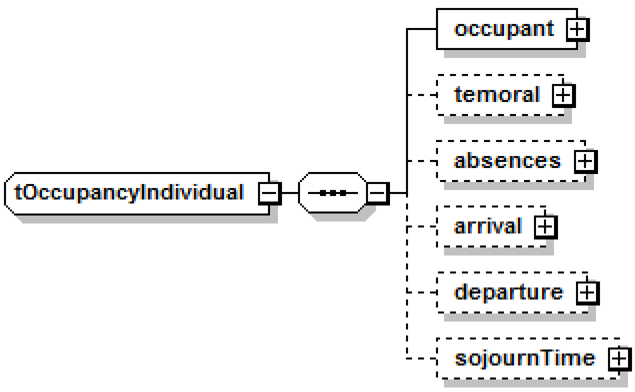
complexType **tLocationTransitionMatrix**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.inertia.eu/2013/schema/occupancy/</td></tr>
<tr><td>children</td><td><u>**transitons**</u></td></tr>
<tr><td>used by</td><td>element <u>**tFlowIndividual/flow**</u></td></tr>
<tr><td>source</td><td><complexType name="tLocationTransitionMatrix"><br><sequence><br><element name="transitons" type="inerocp:tTransitionCouple" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType></td></tr>
</table>

element **tLocationTransitionMatrix/transitons**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><u>**inerocp:tTransitionCouple**</u></td></tr>
<tr><td>properties</td><td>minOcc 1<br>maxOcc unbounded<br>content complex</td></tr>
<tr><td>children</td><td><u>**spaceFromRef**</u> <u>**spaceToRef**</u> <u>**probabilityValue**</u></td></tr>
<tr><td>source</td><td><element name="transitons" type="inerocp:tTransitionCouple" minOccurs="1" maxOccurs="unbounded"/></td></tr>
</table>

complexType **tOccupancyIndividual**

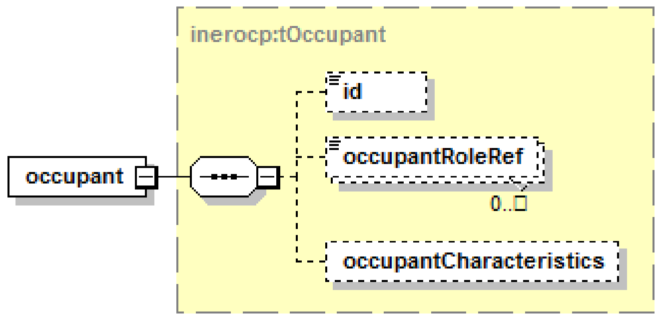
<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.inertia.eu/2013/schema/occupancy/</td></tr>
<tr><td>children</td><td><u>**occupant**</u> <u>**temoral**</u> <u>**absences**</u> <u>**arrival**</u> <u>**departure**</u> <u>**sojournTime**</u></td></tr>
</table>

source

```xml
<complexType name="tOccupancyIndividual">
  <sequence>
    <element name="occupant" type="inerocp:tOccupant" minOccurs="1" maxOccurs="1"/>
    <element name="temoral" type="inerocp:tTemporalType" minOccurs="0" maxOccurs="1"/>
    <element name="absences" type="inerocp:tTemporalDistributionWithPattern" minOccurs="0" maxOccurs="1"/>
    <element name="arrival" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/>
          <element name="statisticalChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </complexType>
    </element>
    <element name="departure" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/>
          <element name="statisticalChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </complexType>
    </element>
    <element name="sojournTime" type="inerocp:tSojournTime" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

element **tOccupancyIndividual/occupant**

diagram



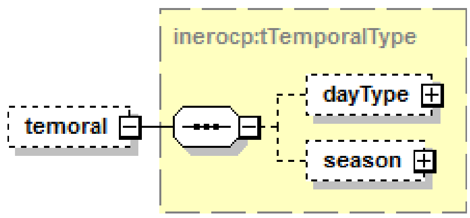

| | |
|---|---|
| type | **inerocp:tOccupant** |
| properties | content complex |
| children | **id occupantRoleRef occupantCharacteristics** |

source

```xml
<element name="occupant" type="inerocp:tOccupant" minOccurs="1" maxOccurs="1"/>
```

element **tOccupancyIndividual/temoral**

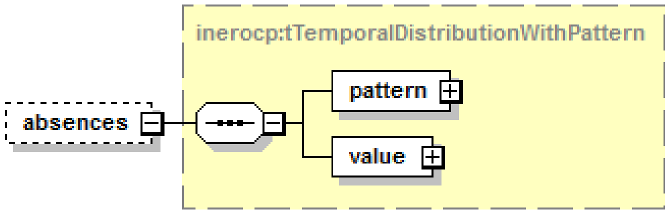
diagram



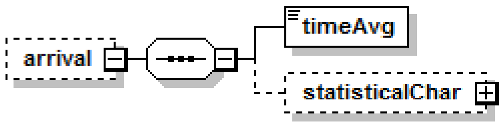

| | |
|---|---|
| type | **inerocp:tTemporalType** |

| | |
|---|---|
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **dayType** **season** |
| source | <element name="temoral" type="inerocp:tTemporalType" minOccurs="0" maxOccurs="1"/> |

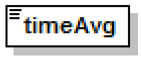
element **tOccupancyIndividual/absences**

| | |
|---|---|
| diagram |   |
| type | **inerocp:tTemporalDistributionWithPattern** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **pattern** **value** |
| source | <element name="absences" type="inerocp:tTemporalDistributionWithPattern" minOccurs="0" maxOccurs="1"/> |

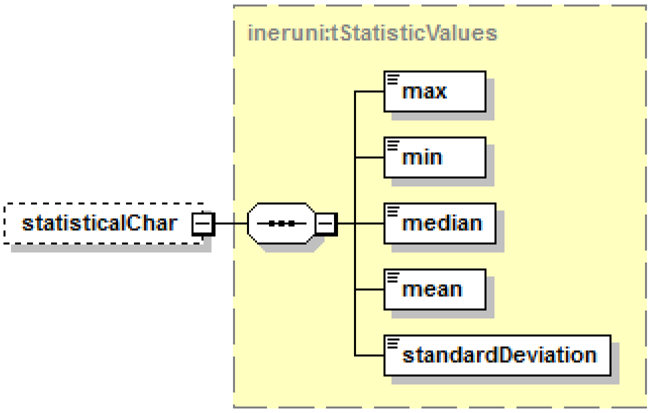
element **tOccupancyIndividual/arrival**

| | |
|---|---|
| diagram |   |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeAvg** **statisticalChar** |
| source | <element name="arrival" minOccurs="0" maxOccurs="1"><br><complexType><br><sequence><br><element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/><br><element name="statisticalChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

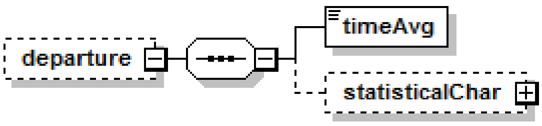
element **tOccupancyIndividual/arrival/timeAvg**

| | |
|---|---|
| diagram |   |
| type | **time** |
| properties | content simple |
| source | <element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/> |

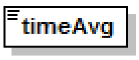
element **tOccupancyIndividual/arrival/statisticalChar**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tStatisticValues** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **max min median mean standardDeviation** |
| source | <element name="statisticalChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/> |

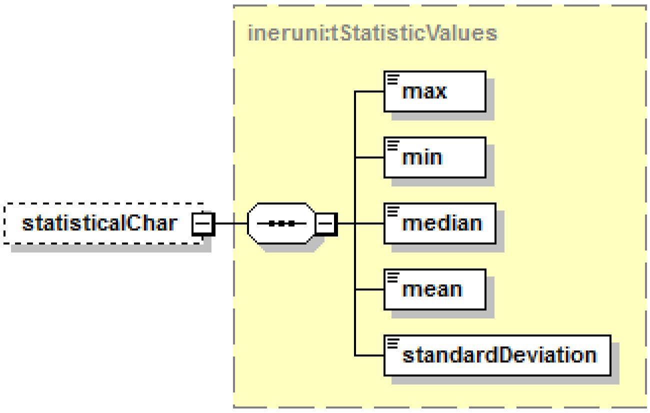
element **tOccupancyIndividual/departure**

| | |
|---|---|
| diagram |  |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeAvg statisticalChar** |
| source | <element name="departure" minOccurs="0" maxOccurs="1"><br><complexType><br><sequence><br><element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/><br><element name="statisticalChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

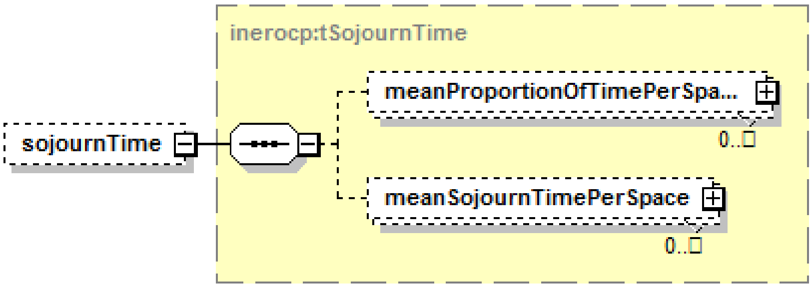
element **tOccupancyIndividual/departure/timeAvg**

| | |
|---|---|
| diagram |  |
| type | **time** |
| properties | content simple |
| source | <element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/> |

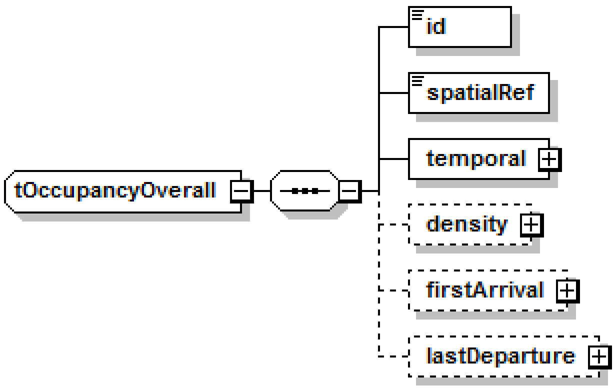
element **tOccupancyIndividual/departure/statisticalChar**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tStatisticValues** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **max min median mean standardDeviation** |
| source | <element name="statisticalChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/> |

element **tOccupancyIndividual/sojournTime**

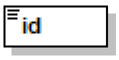
| | |
|---|---|
| diagram |  |
| type | **inerocp:tSojournTime** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **meanProportionOfTimePerSpace meanSojournTimePerSpace** |
| source | <element name="sojournTime" type="inerocp:tSojournTime" minOccurs="0" maxOccurs="1"/> |

complexType **tOccupancyOverall**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **id spatialRef temporal density firstArrival lastDeparture** |
| source | <complexType name="tOccupancyOverall"><br><sequence><br><element name="id" type="string" minOccurs="1" maxOccurs="1"/><br><element name="spatialRef" type="string" minOccurs="1" |

```
maxOccurs="1"/>
    <element name="temporal" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/>
    <element name="density" type="inerocp:tTemporalDistributionType" minOccurs="0" maxOccurs="1"/>
    <element name="firstArrival" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/>
          <element name="statisticChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </complexType>
    </element>
    <element name="lastDeparture" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="timeAvg" type="string" minOccurs="1" maxOccurs="1"/>
          <element name="statisticChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

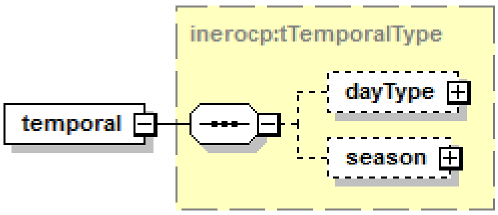
element **tOccupancyOverall/id**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="id" type="string" minOccurs="1" maxOccurs="1"/> |

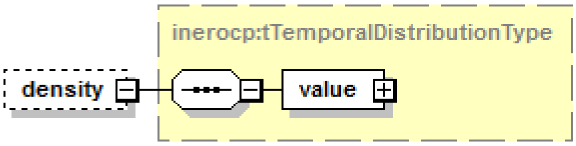
element **tOccupancyOverall/spatialRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/> |

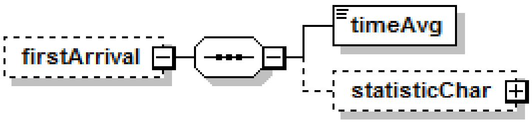
element **tOccupancyOverall/temporal**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tTemporalType** |
| properties | content complex |
| children | **dayType season** |
| source | <element name="temporal" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/> |

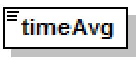
element **tOccupancyOverall/density**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tTemporalDistributionType** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value** |
| source | <element name="density" type="inerocp:tTemporalDistributionType" minOccurs="0" maxOccurs="1"/> |

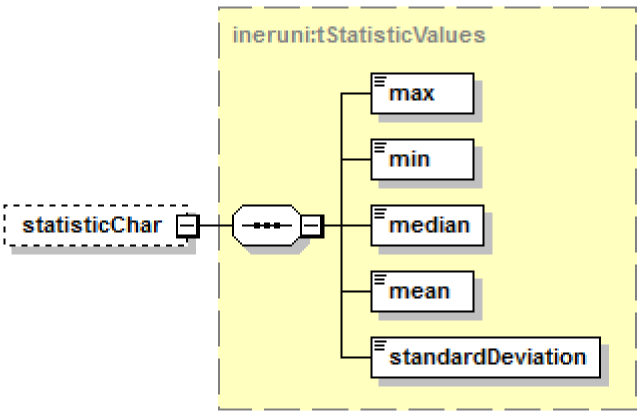
element **tOccupancyOverall/firstArrival**

| | |
|---|---|
| diagram |  |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeAvg statisticChar** |
| source | <element name="firstArrival" minOccurs="0" maxOccurs="1"><br><complexType><br><sequence><br><element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/><br><element name="statisticChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

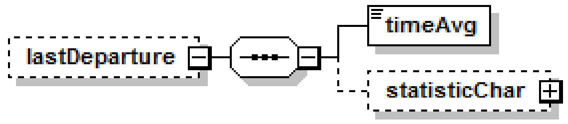
element **tOccupancyOverall/firstArrival/timeAvg**

| | |
|---|---|
| diagram |  |
| type | **time** |
| properties | content simple |
| source | <element name="timeAvg" type="time" minOccurs="1" maxOccurs="1"/> |

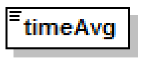
element **tOccupancyOverall/firstArrival/statisticChar**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tStatisticValues** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **max min median mean standardDeviation** |
| source | <element name="statisticChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/> |

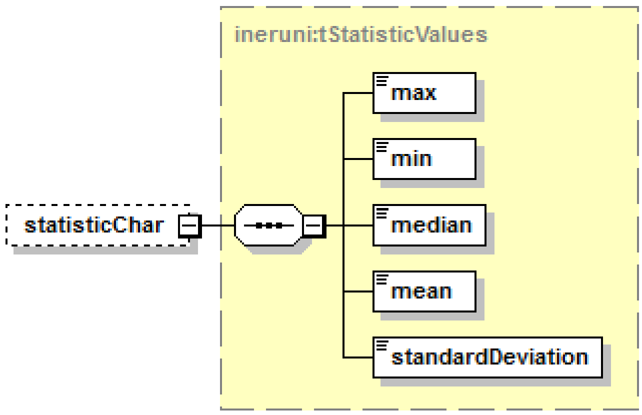
element **tOccupancyOverall/lastDeparture**

| | |
|---|---|
| diagram |  |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | [timeAvg](#) [statisticChar](#) |
| source | <element name="lastDeparture" minOccurs="0" maxOccurs="1"><br><complexType><br><sequence><br><element name="timeAvg" type="string" minOccurs="1" maxOccurs="1"/><br><element name="statisticChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

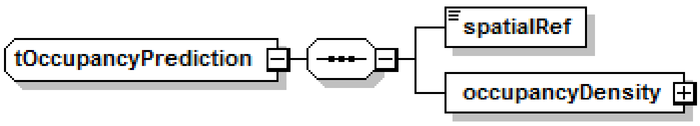
element **tOccupancyOverall/lastDeparture/timeAvg**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="timeAvg" type="string" minOccurs="1" maxOccurs="1"/> |

element **tOccupancyOverall/lastDeparture/statisticChar**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tStatisticValues** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **max min median mean standardDeviation** |
| source | <element name="statisticChar" type="ineruni:tStatisticValues" minOccurs="0" maxOccurs="1"/> |

complexType **tOccupancyPrediction**

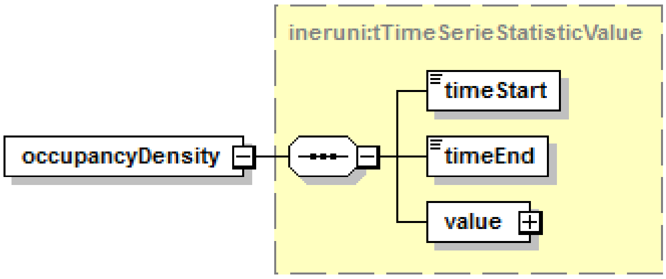
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | [spatialRef](#) [occupancyDensity](#) |
| source | <complexType name="tOccupancyPrediction"><br><sequence><br><element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/> |

| | |
|---|---|
| | <element name="occupancyDensity" type="ineruni:tTimeSerieStatisticValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

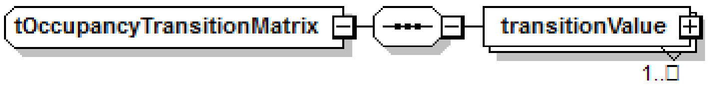
element **tOccupancyPrediction/spatialRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/> |

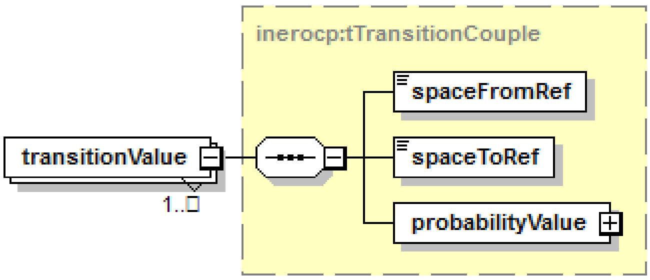
element **tOccupancyPrediction/occupancyDensity**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeSerieStatisticValue** |
| properties | content complex |
| children | **timeStart timeEnd value** |
| source | <element name="occupancyDensity" type="ineruni:tTimeSerieStatisticValue" minOccurs="1" maxOccurs="1"/> |

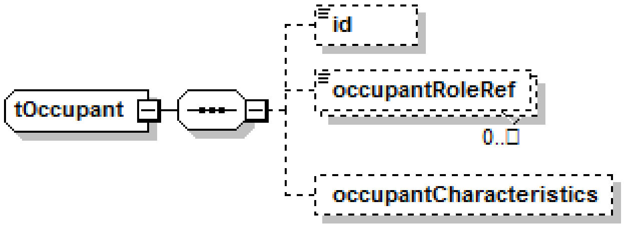
complexType **tOccupancyTransitionMatrix**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **transitionValue** |
| used by | element **tFlowOverall/flowCharacteristics** |
| source | <complexType name="tOccupancyTransitionMatrix"><br><sequence><br><element name="transitionValue" type="inerocp:tTransitionCouple" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tOccupancyTransitionMatrix/transitionValue**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tTransitionCouple** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **spaceFromRef spaceToRef probabilityValue** |
| source | <element name="transitionValue" type="inerocp:tTransitionCouple" minOccurs="1" maxOccurs="unbounded"/> |

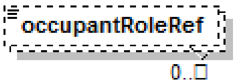
complexType **tOccupant**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **id** **occupantRoleRef** **occupantCharacteristics** |
| used by | element **tOccupancyIndividual/occupant** |
| source | <complexType name="tOccupant"><br><sequence><br><element name="id" type="string" minOccurs="0" maxOccurs="1"/><br><element name="occupantRoleRef" type="string" minOccurs="0" maxOccurs="unbounded"/><br><element name="occupantCharacteristics" type="tns:tUserPreferences" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

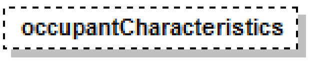
element **tOccupant/id**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="id" type="string" minOccurs="0" maxOccurs="1"/> |

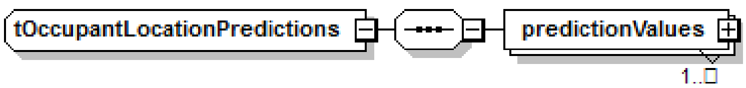
element **tOccupant/occupantRoleRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="occupantRoleRef" type="string" minOccurs="0" maxOccurs="unbounded"/> |

element **tOccupant/occupantCharacteristics**

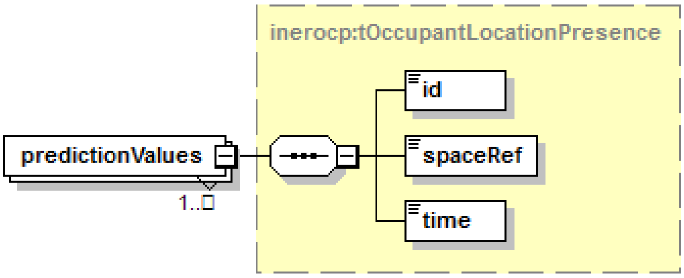
| | |
|---|---|
| diagram |  |
| type | **tns:tUserPreferences** |
| properties | minOcc 0<br>maxOcc 1 |
| source | <element name="occupantCharacteristics" type="tns:tUserPreferences" minOccurs="0" maxOccurs="1"/> |

complexType **tOccupantLocationPredictions**

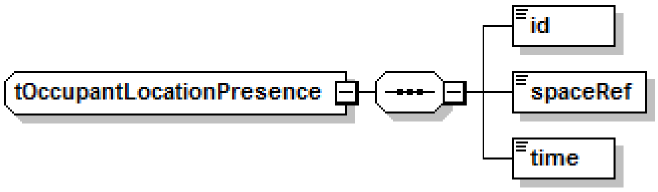
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **predictionValues** |
| source | <complexType name="tOccupantLocationPredictions"><br><sequence><br><element name="predictionValues" type="inerocp:tOccupantLocationPresence" minOccurs="1" maxOccurs="unbounded"/> |

| | |
|---|---|
| | </sequence><br></complexType> |

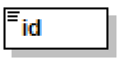
element **tOccupantLocationPredictions/predictionValues**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tOccupantLocationPresence** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **id spaceRef time** |
| source | <element name="predictionValues" type="inerocp:tOccupantLocationPresence" minOccurs="1" maxOccurs="unbounded"/> |

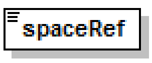
complexType **tOccupantLocationPresence**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **id spaceRef time** |
| used by | element **tOccupantLocationPredictions/predictionValues** |
| source | <complexType name="tOccupantLocationPresence"><br><sequence><br><element name="id" type="string" minOccurs="1" maxOccurs="1"/><br><element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="time" type="dateTime" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

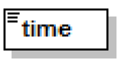
element **tOccupantLocationPresence/id**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="id" type="string" minOccurs="1" maxOccurs="1"/> |

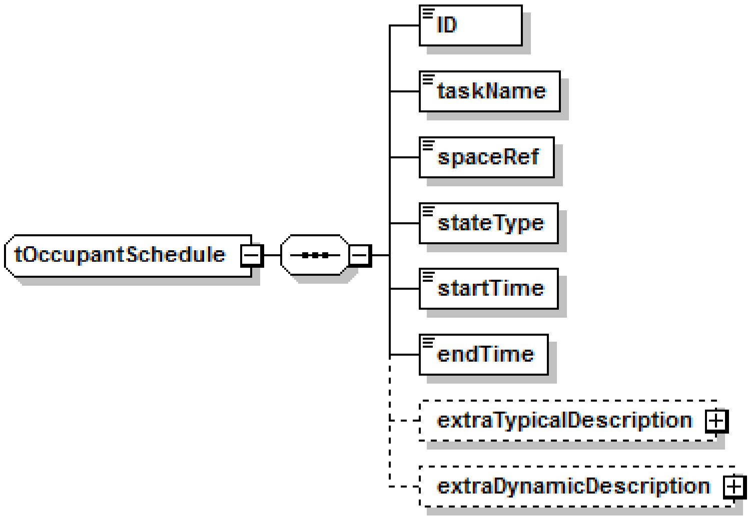
element **tOccupantLocationPresence/spaceRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

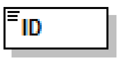
element **tOccupantLocationPresence/time**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="time" type="dateTime" minOccurs="1" maxOccurs="1"/> |

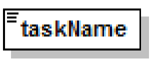
complexType **tOccupantSchedule**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **ID** **taskName** **spaceRef** **stateType** **startTime** **endTime** **extraTypicalDescription** **extraDynamicDescription** |
| source | <complexType name="tOccupantSchedule"><br><sequence><br><element name="ID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="taskName" type="string" minOccurs="1" maxOccurs="1"/><br><element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="stateType" type="ineruni:stateTypeEnum" minOccurs="1" maxOccurs="1"/><br><element name="startTime" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="endTime" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="extraTypicalDescription" type="inerocp:tTypicalExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/><br><element name="extraDynamicDescription" type="inerocp:tDynamiclExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

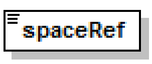
element **tOccupantSchedule/ID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="ID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tOccupantSchedule/taskName**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="taskName" type="string" minOccurs="1" maxOccurs="1"/> |

element **tOccupantSchedule/spaceRef**

| | |
|---|---|
| diagram |  |
| type | **string** |

| | |
|---|---|
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tOccupantSchedule/stateType**

| | |
|---|---|
| diagram |  |
| type | **ineruni:stateTypeEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration presence<br>enumeration absence |
| source | <element name="stateType" type="ineruni:stateTypeEnum" minOccurs="1" maxOccurs="1"/> |

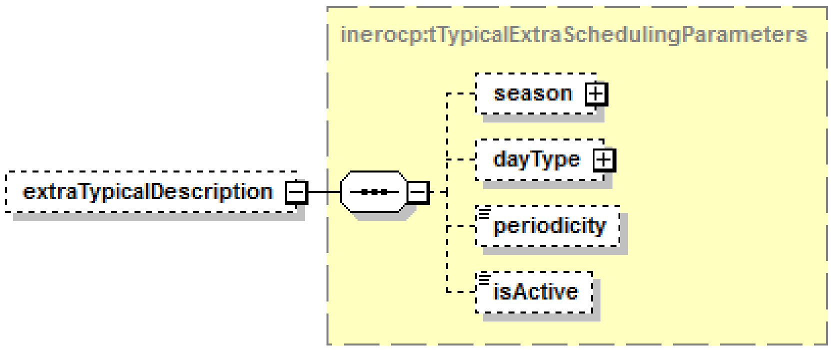
element **tOccupantSchedule/startTime**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="startTime" type="dateTime" minOccurs="1" maxOccurs="1"/> |

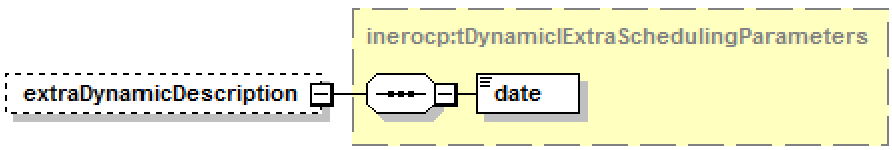
element **tOccupantSchedule/endTime**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="endTime" type="dateTime" minOccurs="1" maxOccurs="1"/> |

element **tOccupantSchedule/extraTypicalDescription**

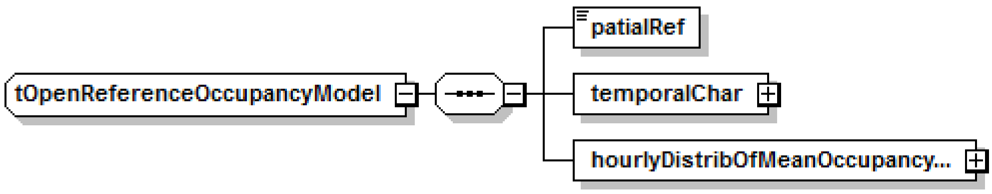
| | |
|---|---|
| diagram |  |
| type | **inerocp:tTypicalExtraSchedulingParameters** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **season dayType periodicity isActive** |
| source | <element name="extraTypicalDescription" type="inerocp:tTypicalExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/> |

element **tOccupantSchedule/extraDynamicDescription**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tDynamicExtraSchedulingParameters** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |

| | |
|---|---|
| children | **date** |
| source | <element name="extraDynamicDescription" type="inerocp:tDynamicExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/> |

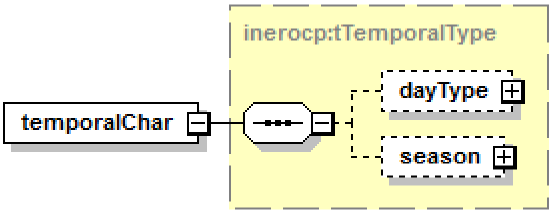
complexType **tOpenReferenceOccupancyModel**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **patialRef** **temporalChar** **hourlyDistribOfMeanOccupancyDensity** |
| source | <complexType name="tOpenReferenceOccupancyModel"><br><sequence><br><element name="patialRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="temporalChar" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/><br><element name="hourlyDistribOfMeanOccupancyDensity" type="inerocp:tTemporalDistributionType" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

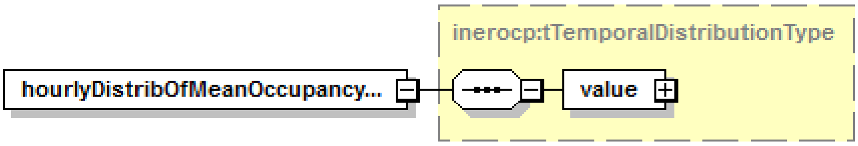
element **tOpenReferenceOccupancyModel/patialRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="patialRef" type="string" minOccurs="1" maxOccurs="1"/> |

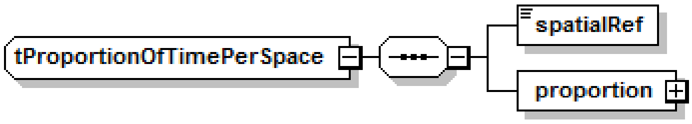
element **tOpenReferenceOccupancyModel/temporalChar**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tTemporalType** |
| properties | content complex |
| children | **dayType** **season** |
| source | <element name="temporalChar" type="inerocp:tTemporalType" minOccurs="1" maxOccurs="1"/> |

element **tOpenReferenceOccupancyModel/hourlyDistribOfMeanOccupancyDensity**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tTemporalDistributionType** |
| properties | content complex |
| children | **value** |
| source | <element name="hourlyDistribOfMeanOccupancyDensity" type="inerocp:tTemporalDistributionType" minOccurs="1" maxOccurs="1"/> |

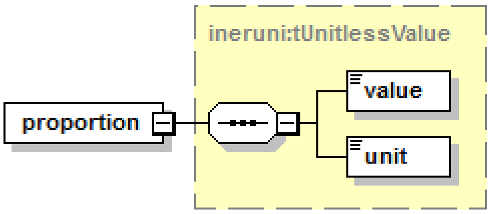
complexType **tProportionOfTimePerSpace**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **spatialRef** **proportion** |
| used by | element **tSojournTime/meanProportionOfTimePerSpace** |
| source | <complexType name="tProportionOfTimePerSpace"><br><sequence><br><element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="proportion" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

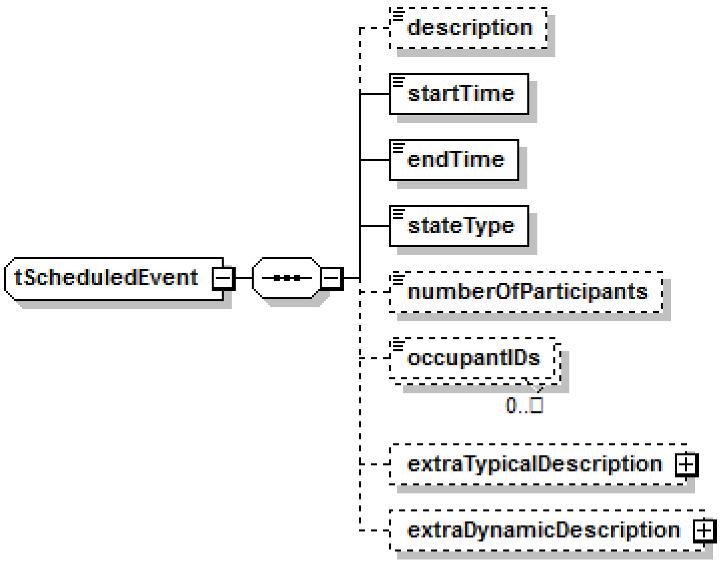
element **tProportionOfTimePerSpace/spatialRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spatialRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tProportionOfTimePerSpace/proportion**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="proportion" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/> |

complexType **tScheduledEvent**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **description** **startTime** **endTime** **stateType** **numberOfParticipants** **occupantIDs** **extraTypicalDescription** **extraDynamicDescription** |

| | |
|---|---|
| used by | element **tSpaceSchedule/scheduledEvent** |
| source | <complexType name="tScheduledEvent"><br><sequence><br><element name="description" type="string" minOccurs="0" maxOccurs="1"/><br><element name="startTime" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="endTime" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="stateType" type="ineruni:stateTypeEnum" minOccurs="1" maxOccurs="1"/><br><element name="numberOfParticipants" type="integer" minOccurs="0" maxOccurs="1"/><br><element name="occupantIDs" type="string" minOccurs="0" maxOccurs="unbounded"/><br><element name="extraTypicalDescription" type="inerocp:tTypicalExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/><br><element name="extraDynamicDescription" type="inerocp:tDynamiclExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

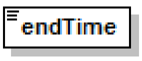
element **tScheduledEvent/description**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="description" type="string" minOccurs="0" maxOccurs="1"/> |

element **tScheduledEvent/startTime**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="startTime" type="dateTime" minOccurs="1" maxOccurs="1"/> |

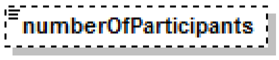
element **tScheduledEvent/endTime**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="endTime" type="dateTime" minOccurs="1" maxOccurs="1"/> |

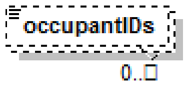
element **tScheduledEvent/stateType**

| | |
|---|---|
| diagram |  |
| type | **ineruni:stateTypeEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration presence<br>enumeration absence |
| source | <element name="stateType" type="ineruni:stateTypeEnum" minOccurs="1" maxOccurs="1"/> |

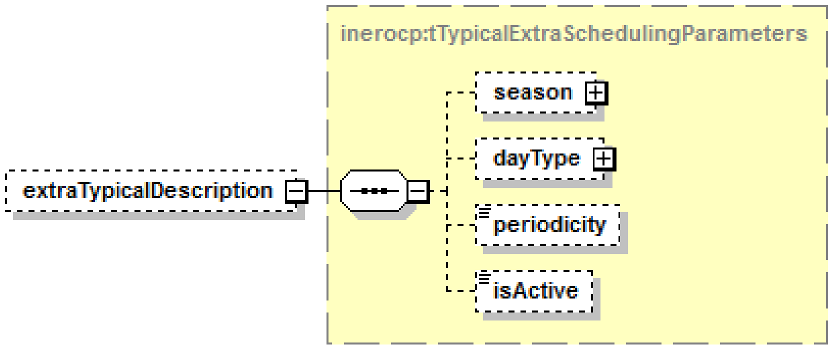
element **tScheduledEvent/numberOfParticipants**

| | |
|---|---|
| diagram |   |
| type | **integer** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="numberOfParticipants" type="integer" minOccurs="0" maxOccurs="1"/> |

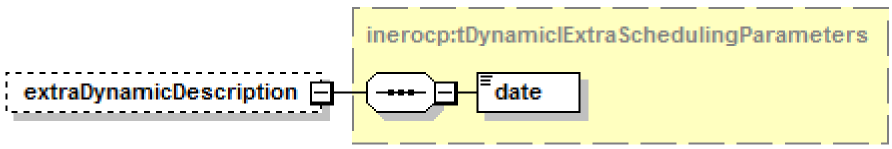
element **tScheduledEvent/occupantIDs**

| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="occupantIDs" type="string" minOccurs="0" maxOccurs="unbounded"/> |

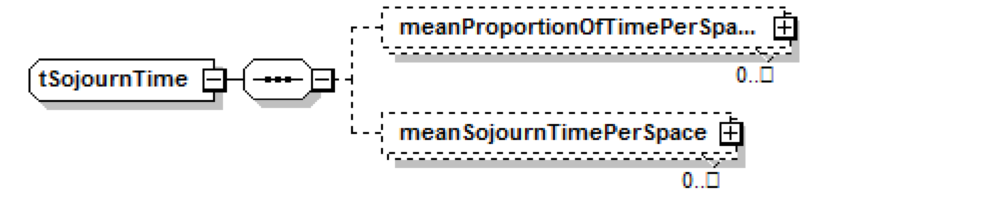
element **tScheduledEvent/extraTypicalDescription**

| | |
|---|---|
| diagram |   |
| type | **inerocp:tTypicalExtraSchedulingParameters** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **season dayType periodicity isActive** |
| source | <element name="extraTypicalDescription" type="inerocp:tTypicalExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/> |

element **tScheduledEvent/extraDynamicDescription**

| | |
|---|---|
| diagram |   |
| type | **inerocp:tDynamiclExtraSchedulingParameters** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **date** |
| source | <element name="extraDynamicDescription" type="inerocp:tDynamiclExtraSchedulingParameters" minOccurs="0" maxOccurs="1"/> |

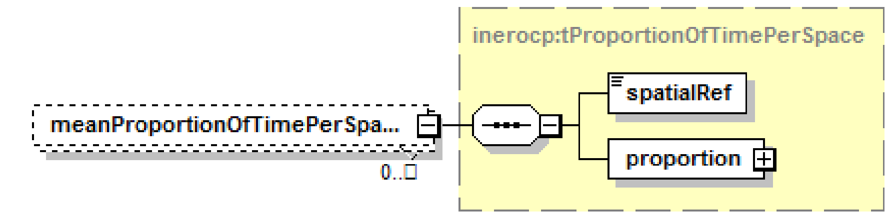
## complexType **tSojournTime**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **meanProportionOfTimePerSpace** **meanSojournTimePerSpace** |
| used by | element **tOccupancyIndividual/sojournTime** |
| source | see below |

source:

```
<complexType name="tSojournTime">
  <sequence>
    <element name="meanProportionOfTimePerSpace" type="inerocp:tProportionOfTimePerSpace" minOccurs="0" maxOccurs="unbounded"/>
    <element name="meanSojournTimePerSpace" type="inerocp:tDurationSpentPerSpace" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

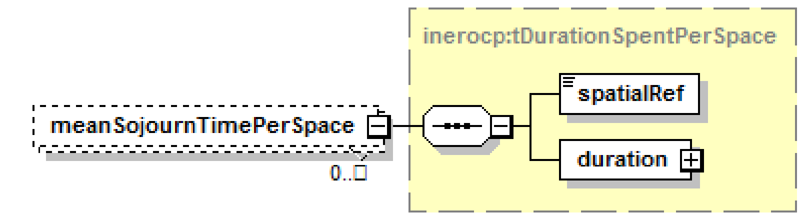
## element **tSojournTime/meanProportionOfTimePerSpace**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tProportionOfTimePerSpace** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **spatialRef** **proportion** |
| source | see below |

source:

```
<element name="meanProportionOfTimePerSpace" type="inerocp:tProportionOfTimePerSpace" minOccurs="0" maxOccurs="unbounded"/>
```

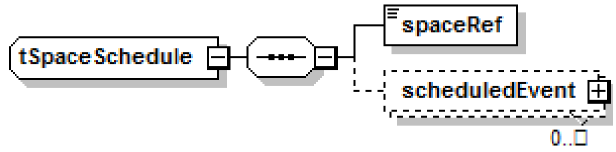
## element **tSojournTime/meanSojournTimePerSpace**

| | |
|---|---|
| diagram |  |
| type | **inerocp:tDurationSpentPerSpace** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **spatialRef** **duration** |
| source | see below |

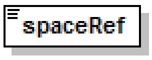
source:

```
<element name="meanSojournTimePerSpace" type="inerocp:tDurationSpentPerSpace" minOccurs="0" maxOccurs="unbounded"/>
```

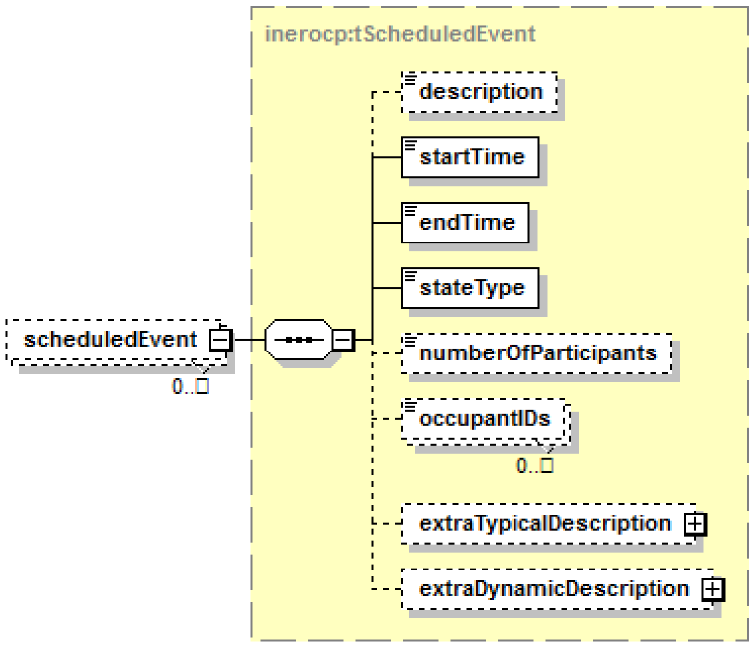
complexType **tSpaceSchedule**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **spaceRef** **scheduledEvent** |
| source | <complexType name="tSpaceSchedule"><br><sequence><br><element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="scheduledEvent" type="inerocp:tScheduledEvent" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

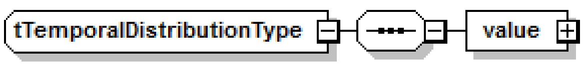
element **tSpaceSchedule/spaceRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tSpaceSchedule/scheduledEvent**

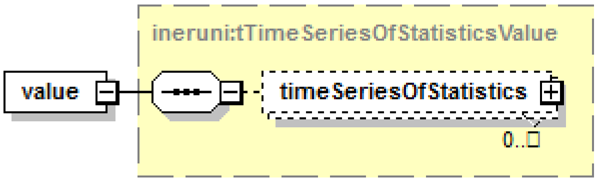
| | |
|---|---|
| diagram |  |
| type | **inerocp:tScheduledEvent** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **description** **startTime** **endTime** **stateType** **numberOfParticipants** **occupantIDs** **extraTypicalDescription** **extraDynamicDescription** |
| source | <element name="scheduledEvent" type="inerocp:tScheduledEvent" minOccurs="0" maxOccurs="unbounded"/> |

complexType **tTemporalDistributionType**

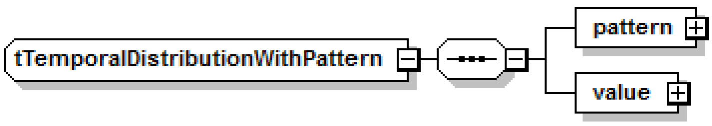
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **value** |
| used by | elements **tOccupancyOverall/density** **tOpenReferenceOccupancyModel/hourlyDistribOfMeanOccupancyDensity** |

| | tTemporalDistributionWithPattern/value |
|---|---|
| source | <complexType name="tTemporalDistributionType"><br><sequence><br><element name="value" type="ineruni:tTimeSeriesOfStatisticsValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

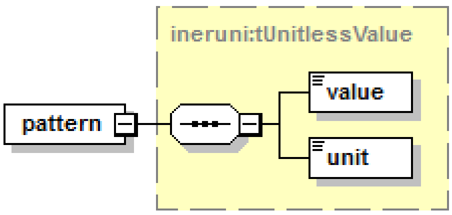
element **tTemporalDistributionType/value**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeSeriesOfStatisticsValue** |
| properties | content complex |
| children | **timeSeriesOfStatistics** |
| source | <element name="value" type="ineruni:tTimeSeriesOfStatisticsValue" minOccurs="1" maxOccurs="1"/> |

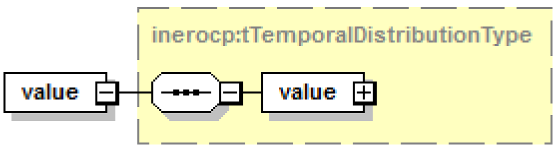
complexType **tTemporalDistributionWithPattern**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **pattern value** |
| used by | element **tOccupancyIndividual/absences** |
| source | <complexType name="tTemporalDistributionWithPattern"><br><sequence><br><element name="pattern" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/><br><element name="value" type="inerocp:tTemporalDistributionType" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tTemporalDistributionWithPattern/pattern**

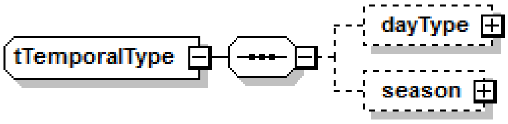
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="pattern" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/> |

element **tTemporalDistributionWithPattern/value**

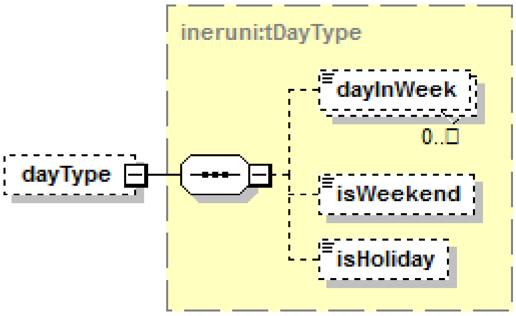
| | |
|---|---|
| diagram |  |
| type | **inerocp:tTemporalDistributionType** |
| properties | content complex |
| children | **value** |

| | |
|---|---|
| source | <element name="value" type="inerocp:tTemporalDistributionType" minOccurs="1" maxOccurs="1"/> |

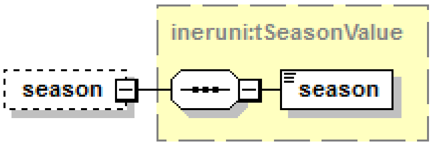
complexType **tTemporalType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **dayType** **season** |
| used by | elements **tOccupancyIndividual/temoral** **tOccupancyOverall/temporal** **tFlowOverall/temporal** **tFlowIndividual/temporal** **tOpenReferenceOccupancyModel/temporalChar** |
| source | <complexType name="tTemporalType"><br><sequence><br><element name="dayType" type="ineruni:tDayType" minOccurs="0" maxOccurs="1"/><br><element name="season" type="ineruni:tSeasonValue" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

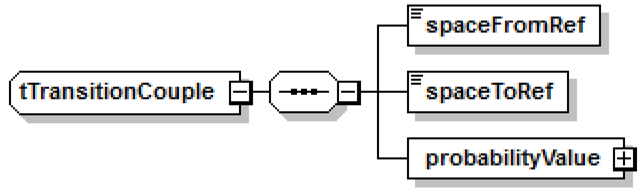
element **tTemporalType/dayType**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tDayType** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **dayInWeek isWeekend isHoliday** |
| source | <element name="dayType" type="ineruni:tDayType" minOccurs="0" maxOccurs="1"/> |

element **tTemporalType/season**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tSeasonValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **season** |
| source | <element name="season" type="ineruni:tSeasonValue" minOccurs="0" maxOccurs="1"/> |

complexType **tTransitionCouple**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | spaceFromRef spaceToRef probabilityValue |
| used by | elements tOccupancyTransitionMatrix/transitionValue<br>tLocationTransitionMatrix/transitons |
| source | <complexType name="tTransitionCouple"><br><sequence><br><element name="spaceFromRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="spaceToRef" type="string" minOccurs="1" maxOccurs="1"/><br><element name="probabilityValue" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

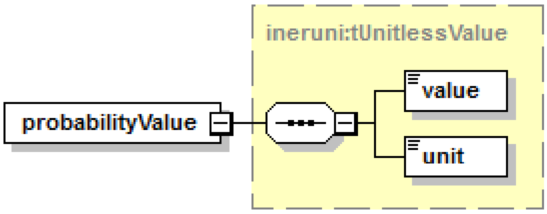
element **tTransitionCouple/spaceFromRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spaceFromRef" type="string" minOccurs="1" maxOccurs="1"/> |

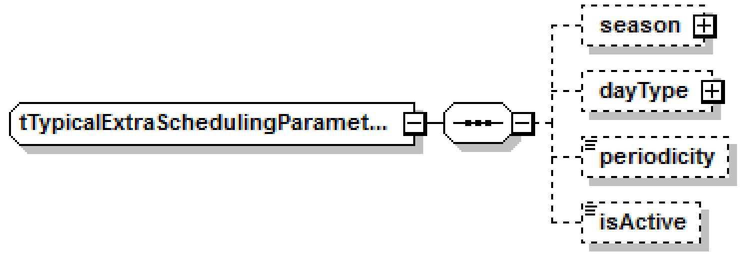
element **tTransitionCouple/spaceToRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spaceToRef" type="string" minOccurs="1" maxOccurs="1"/> |

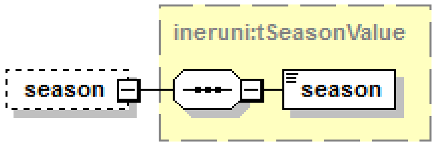
element **tTransitionCouple/probabilityValue**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="probabilityValue" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/> |

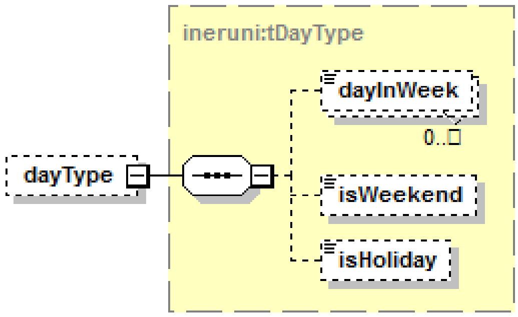
complexType **tTypicalExtraSchedulingParameters**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/occupancy/ |
| children | **season dayType periodicity isActive** |
| used by | elements **tScheduledEvent/extraTypicalDescription** **tOccupantSchedule/extraTypicalDescription** |
| source | <complexType name="tTypicalExtraSchedulingParameters"><br><sequence><br><element name="season" type="ineruni:tSeasonValue" minOccurs="0" maxOccurs="1"/><br><element name="dayType" type="ineruni:tDayType" minOccurs="0" maxOccurs="1"/><br><element name="periodicity" type="string" minOccurs="0" maxOccurs="1"/><br><element name="isActive" type="boolean" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

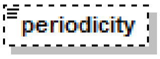
element **tTypicalExtraSchedulingParameters/season**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tSeasonValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **season** |
| source | <element name="season" type="ineruni:tSeasonValue" minOccurs="0" maxOccurs="1"/> |

element **tTypicalExtraSchedulingParameters/dayType**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tDayType** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **dayInWeek isWeekend isHoliday** |
| source | <element name="dayType" type="ineruni:tDayType" minOccurs="0" maxOccurs="1"/> |

element **tTypicalExtraSchedulingParameters/periodicity**

| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="periodicity" type="string" minOccurs="0" maxOccurs="1"/> |

element **tTypicalExtraSchedulingParameters/isActive**

| | |
|---|---|
| diagram |   |
| type | **boolean** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="isActive" type="boolean" minOccurs="0" maxOccurs="1"/> |

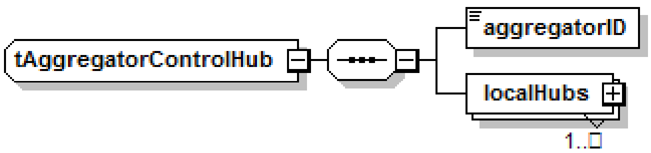
XML Schema documentation generated by **XMLSpy** Schema Editor **http://www.altova.com/xmlspy**

Schema **inertiaHubsandBIM.xsd**

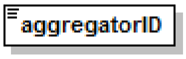
| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaHubsandBIM.xsd** |
| attributeFormDefault: | **unqualified** |
| elementFormDefault: | **unqualified** |
| targetNamespace: | **http://www.inertia.eu/2013/schema/bim/** |

Complex types
**tAggregatorControlHub**
**tBuilding**
**tEquipment**
**tEquipmentTypeCharacteristics**
**tGroupOfSpaces**
**tLocalControlHub**
**tSpace**
**tStateEnergyValue**

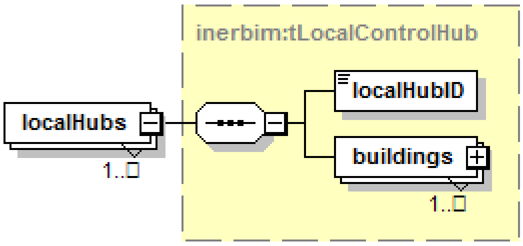
complexType **tAggregatorControlHub**

| | |
|---|---|
| diagram |   |
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | **aggregatorID localHubs** |
| source | <complexType name="tAggregatorControlHub"><br><sequence><br><element name="aggregatorID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="localHubs" type="inerbim:tLocalControlHub" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tAggregatorControlHub/aggregatorID**

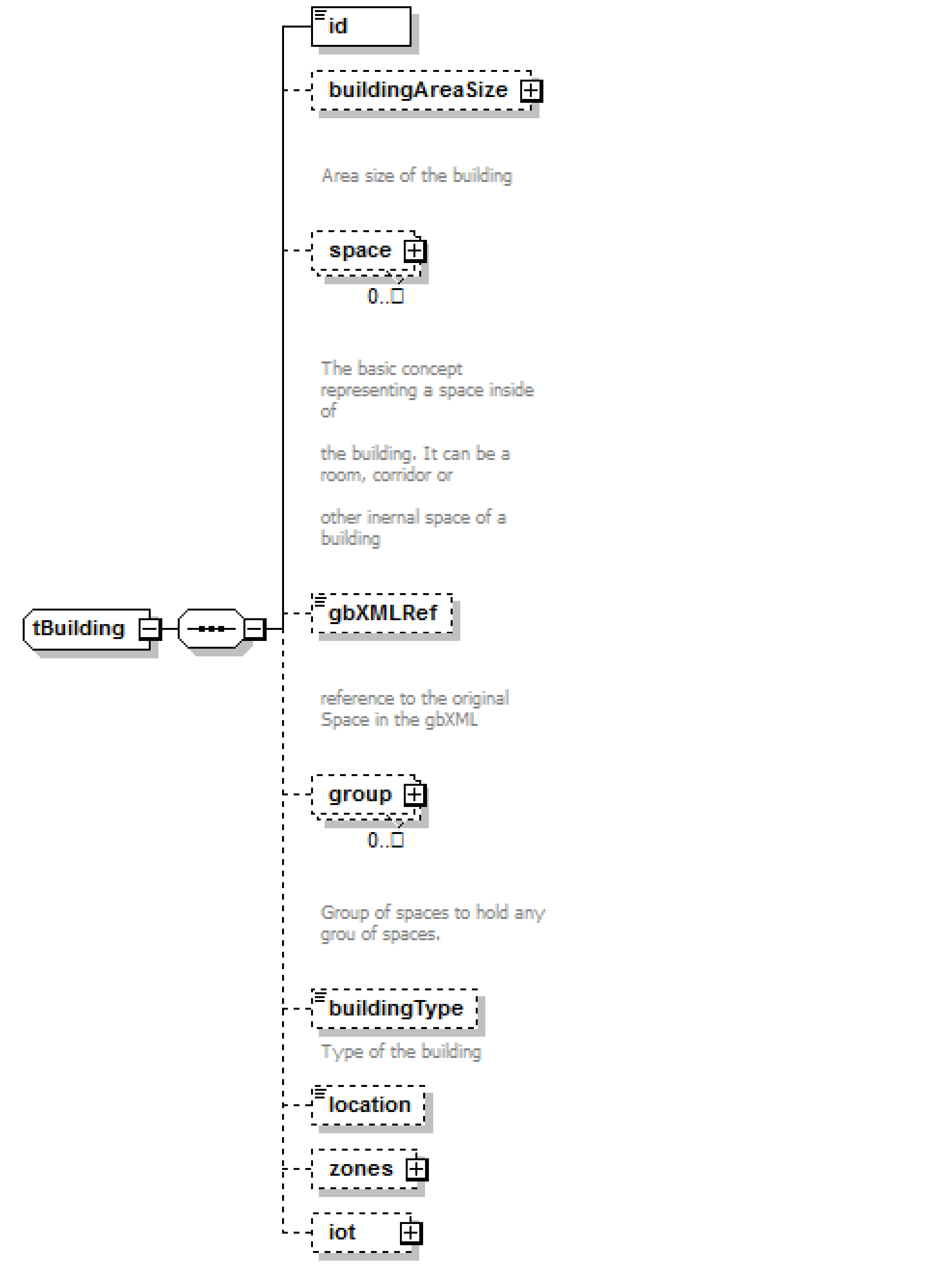
| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="aggregatorID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tAggregatorControlHub/localHubs**

| | |
|---|---|
| diagram |  |
| type | **inerbim:tLocalControlHub** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **localHubID** **buildings** |
| source | <element name="localHubs" type="inerbim:tLocalControlHub" minOccurs="1" maxOccurs="unbounded"/> |

complexType **tBuilding**

diagram



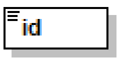

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | id buildingAreaSize space gbXMLRef group buildingType location zones iot |
| used by | element tLocalControlHub/buildings |

source

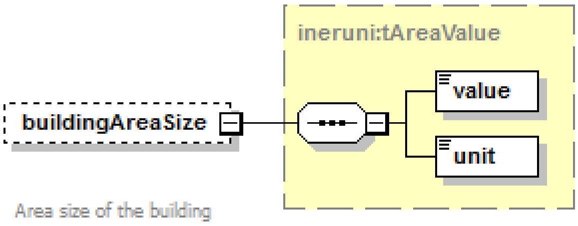
```
<complexType name="tBuilding">
  <annotation>
    <documentation/>
  </annotation>
  <sequence>
    <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="buildingAreaSize" type="ineruni:tAreaValue"
minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>
                              Area size of the building
                          </documentation>
      </annotation>
```

```
    </element>
    <element name="space" type="inerbim:tSpace" minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>
                        The basic concept representing a space inside of the building. It can be a room, corridor or other inernal space of a building
                </documentation>
      </annotation>
    </element>
    <element name="gbXMLRef" type="string" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>
                        reference to the original Space in the gbXML
                </documentation>
      </annotation>
    </element>
    <element name="group" type="inerbim:tGroupOfSpaces" minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>
                        Group of spaces to hold any grou of spaces.
                </documentation>
      </annotation>
    </element>
    <element name="buildingType" type="ineruni:buildingTypeEnum" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>Type of the building</documentation>
      </annotation>
    </element>
    <element name="location" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="zones" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="zone" type="string" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
    <element name="iot" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="sensor" type="inerDevice:tSensor" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

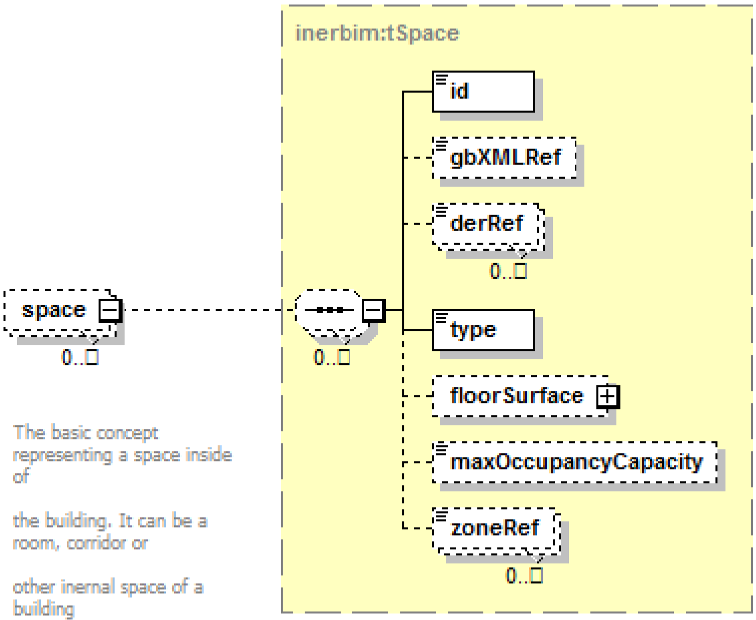
element **tBuilding/id**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="id" type="string" minOccurs="1" maxOccurs="1"/> |

element **tBuilding/buildingAreaSize**

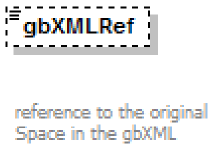
| | |
|---|---|
| diagram |   |
| type | **ineruni:tAreaValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| annotation | documentation<br>Area size of the building |
| source | <element name="buildingAreaSize" type="ineruni:tAreaValue" minOccurs="0" maxOccurs="1"><br><annotation><br><documentation><br>Area size of the building<br></documentation><br></annotation><br></element> |

element **tBuilding/space**

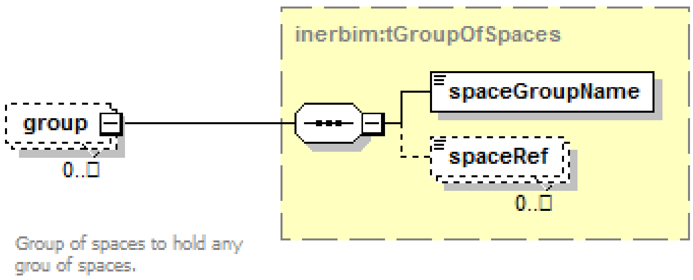
| | |
|---|---|
| diagram |   |
| type | **inerbim:tSpace** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **id gbXMLRef derRef type floorSurface maxOccupancyCapacity zoneRef** |
| annotation | documentation<br>The basic concept representing a space inside of the building. It can be a room, corridor or other inernal space of a building |
| source | <element name="space" type="inerbim:tSpace" minOccurs="0" maxOccurs="unbounded"> |

| | |
|---|---|
| | <annotation><br><documentation><br>The basic concept representing a space inside of the building. It can be a room, corridor or other inernal space of a building<br></documentation><br></annotation><br></element> |

element **tBuilding/gbXMLRef**

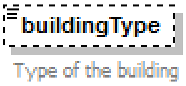
| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| annotation | documentation<br>reference to the original Space in the gbXML |
| source | <element name="gbXMLRef" type="string" minOccurs="0" maxOccurs="1"><br><annotation><br><documentation><br>reference to the original Space in the gbXML<br></documentation><br></annotation><br></element> |

element **tBuilding/group**

| | |
|---|---|
| diagram |  |
| type | **inerbim:tGroupOfSpaces** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **spaceGroupName** **spaceRef** |
| annotation | documentation<br>Group of spaces to hold any grou of spaces. |
| source | <element name="group" type="inerbim:tGroupOfSpaces" minOccurs="0" maxOccurs="unbounded"><br><annotation><br><documentation><br>Group of spaces to hold any grou of spaces.<br></documentation><br></annotation><br></element> |

element **tBuilding/buildingType**

diagram




| | |
|---|---|
| type | **ineruni:buildingTypeEnum** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | AutomotiveFacility | |
| enumeration | ConventionCenter | |
| enumeration | Courthouse | |
| enumeration | DiningBarLoungeOrLeisure | |
| enumeration | DiningCafeteriaFastFood | |
| enumeration | DiningFamily | |
| enumeration | Dormitory | |
| enumeration | ExerciseCenter | |
| enumeration | FireStation | |
| enumeration | Gymnasium | |
| enumeration | HospitalOrHealthcare | |
| enumeration | Hotel | |
| enumeration | Library | |
| enumeration | Manufacturing | |
| enumeration | Motel | |
| enumeration | MotionPictureTheatre | |
| enumeration | MultiFamily | |
| enumeration | Museum | |
| enumeration | Office | |
| enumeration | ParkingGarage | |
| enumeration | Penitentiary | |
| enumeration | PerformingArtsTheater | |
| enumeration | PoliceStation | |
| enumeration | PostOffice | |
| enumeration | ReligiousBuilding | |
| enumeration | Retail | |
| enumeration | SchoolOrUniversity | |
| enumeration | SingleFamily | |
| enumeration | SportsArena | |
| enumeration | TownHall | |
| enumeration | Transportation | |
| enumeration | Unknown | documentation<br>Enumeration value to be used when there is no building type information available. |
| enumeration | Warehouse | |
| enumeration | Workshop | |

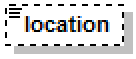
annotation

documentation
Type of the building

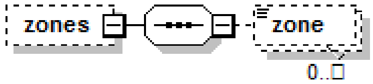
source

```
<element name="buildingType" type="ineruni:buildingTypeEnum"
minOccurs="0" maxOccurs="1">
  <annotation>
    <documentation>Type of the building</documentation>
  </annotation>
</element>
```

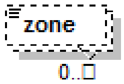
element **tBuilding/location**

| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="location" type="string" minOccurs="0" maxOccurs="1"/> |

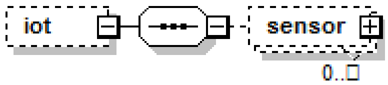
element **tBuilding/zones**

| | |
|---|---|
| diagram |   |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **zone** |
| source | <element name="zones" minOccurs="0" maxOccurs="1"><br><complexType><br><sequence><br><element name="zone" type="string" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType><br></element> |

element **tBuilding/zones/zone**

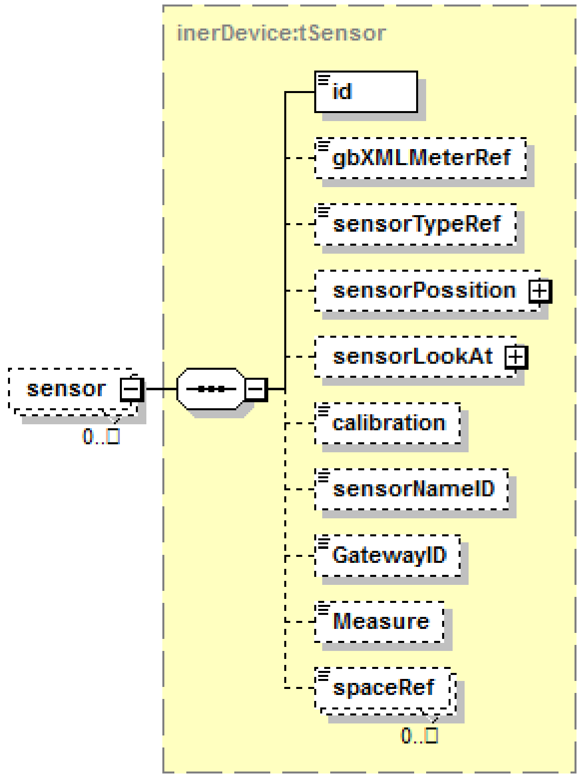
| | |
|---|---|
| diagram |   |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="zone" type="string" minOccurs="0" maxOccurs="unbounded"/> |

element **tBuilding/iot**

| | |
|---|---|
| diagram |   |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **sensor** |
| source | <element name="iot" minOccurs="0" maxOccurs="1"><br><complexType><br><sequence><br><element name="sensor" type="inerDevice:tSensor" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType><br></element> |

element **tBuilding/iot/sensor**

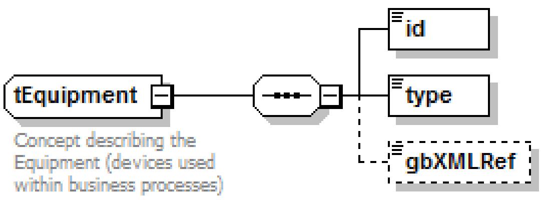
diagram




| | |
|---|---|
| type | **inerdev:tSensor** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **id gbXMLMeterRef sensorTypeRef sensorPossition sensorLookAt calibration sensorNameID GatewayID Measure spaceRef** |
| source | <element name="sensor" type="inerDevice:tSensor" minOccurs="0" maxOccurs="unbounded"/> |

complexType **tEquipment**

diagram




| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | **id type gbXMLRef** |
| annotation | documentation<br>Concept describing the Equipment (devices used within business processes) |

source

```
<complexType name="tEquipment">
  <annotation>
    <documentation>Concept describing the Equipment (devices used within business processes)</documentation>
  </annotation>
  <sequence>
    <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="type" type="ineruni:intEquipTypeEnum" minOccurs="1" maxOccurs="1"/>
    <element name="gbXMLRef" type="string" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

## element **tEquipment/id**

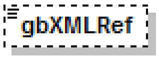
| | |
|---|---|
| diagram | id |
| type | **string** |
| properties | content simple |
| source | <element name="id" type="string" minOccurs="1" maxOccurs="1"/> |

## element **tEquipment/type**

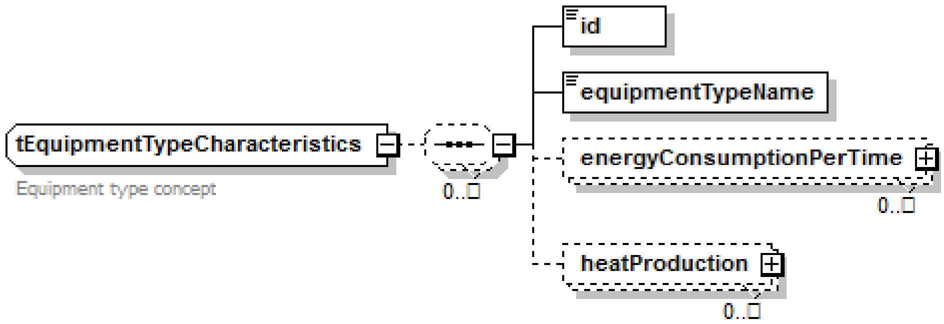
| | | | |
|---|---|---|---|
| diagram | type | | |
| type | **ineruni:intEquipTypeEnum** | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | GeneralPlugload | |
| | enumeration | Computer | |
| | enumeration | Copier | |
| | enumeration | Refrigerator | |
| | enumeration | Stove | |
| | enumeration | Cooktop | |
| | enumeration | Fryer | |
| | enumeration | Freezer | |
| | enumeration | ClothesDryer | |
| | enumeration | ClothesWasher | |
| | enumeration | Dishwasher | |
| | enumeration | TV | |
| | enumeration | VCR | |
| | enumeration | Microwave | |
| | enumeration | Fan | |
| | enumeration | AudioEquip | |
| | enumeration | MiscEquip | |
| | enumeration | Toilet | |
| | enumeration | Urinal | |
| | enumeration | Shower | |
| | enumeration | Sink | |
| | enumeration | Pool | |
| | enumeration | HotTub | |
| | enumeration | Sauna | |
| | enumeration | BatteryCharger | |
| | enumeration | Furnace | |
| | enumeration | IndustrialEquip | |
| | enumeration | Printer | |
| | enumeration | Vending | |
| | enumeration | Elevator | |
| | enumeration | Escalator | |
| | enumeration | GeneralLaundryEquip | |
| | enumeration | GeneralKitchenEquip | |
| | enumeration | GeneralMedicalEquip | |
| | enumeration | GeneralOfficeEquip | |
| | enumeration | GeneralEntertainmentEquip | |
| | enumeration | GeneralRefrigerationEquip | |
| | enumeration | GeneralFitnessEquip | |
| | enumeration | GeneralVideoEquip | |
| | enumeration | GeneralToilet | |

| | |
|---|---|
| source | <element name="type" type="ineruni:intEquipTypeEnum" minOccurs="1" maxOccurs="1"/> |

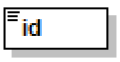
element **tEquipment/gbXMLRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="gbXMLRef" type="string" minOccurs="0" maxOccurs="1"/> |

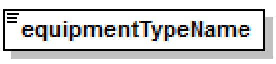
complexType **tEquipmentTypeCharacteristics**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | id equipmentTypeName energyConsumptionPerTime heatProduction |
| annotation | documentation<br>Equipment type concept |
| source | <complexType name="tEquipmentTypeCharacteristics"><br><annotation><br><documentation>Equipment type concept</documentation><br></annotation><br><sequence minOccurs="0" maxOccurs="unbounded"><br><element name="id" type="string" minOccurs="1" maxOccurs="1"/><br><element name="equipmentTypeName" type="ineruni:intEquipTypeEnum" minOccurs="1" maxOccurs="1"/><br><element name="energyConsumptionPerTime" type="inerbim:tStateEnergyValue" minOccurs="0" maxOccurs="unbounded"/><br><element name="heatProduction" type="inerbim:tStateEnergyValue" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tEquipmentTypeCharacteristics/id**

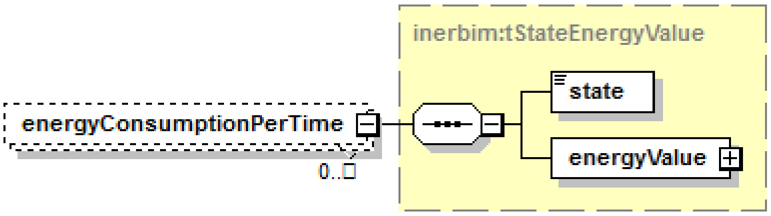
| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="id" type="string" minOccurs="1" maxOccurs="1"/> |

element **tEquipmentTypeCharacteristics/equipmentTypeName**

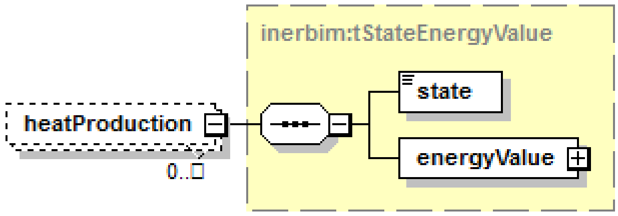
| | |
|---|---|
| diagram |  |
| type | **ineruni:intEquipTypeEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration GeneralPlugload<br>enumeration Computer<br>enumeration Copier<br>enumeration Refrigerator |

| | | |
|---|---|---|
| | enumeration | Stove |
| | enumeration | Cooktop |
| | enumeration | Fryer |
| | enumeration | Freezer |
| | enumeration | ClothesDryer |
| | enumeration | ClothesWasher |
| | enumeration | Dishwasher |
| | enumeration | TV |
| | enumeration | VCR |
| | enumeration | Microwave |
| | enumeration | Fan |
| | enumeration | AudioEquip |
| | enumeration | MiscEquip |
| | enumeration | Toilet |
| | enumeration | Urinal |
| | enumeration | Shower |
| | enumeration | Sink |
| | enumeration | Pool |
| | enumeration | HotTub |
| | enumeration | Sauna |
| | enumeration | BatteryCharger |
| | enumeration | Furnace |
| | enumeration | IndustrialEquip |
| | enumeration | Printer |
| | enumeration | Vending |
| | enumeration | Elevator |
| | enumeration | Escalator |
| | enumeration | GeneralLaundryEquip |
| | enumeration | GeneralKitchenEquip |
| | enumeration | GeneralMedicalEquip |
| | enumeration | GeneralOfficeEquip |
| | enumeration | GeneralEntertainmentEquip |
| | enumeration | GeneralRefrigerationEquip |
| | enumeration | GeneralFitnessEquip |
| | enumeration | GeneralVideoEquip |
| | enumeration | GeneralToilet |
| source | <element name="equipmentTypeName" type="ineruni:intEquipTypeEnum" minOccurs="1" maxOccurs="1"/> | |

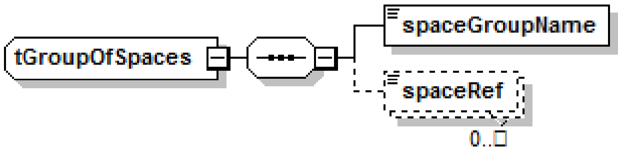
element **tEquipmentTypeCharacteristics/energyConsumptionPerTime**

| | |
|---|---|
| diagram |  |
| type | **inerbim:tStateEnergyValue** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **state energyValue** |
| source | <element name="energyConsumptionPerTime" type="inerbim:tStateEnergyValue" minOccurs="0" maxOccurs="unbounded"/> |

element **tEquipmentTypeCharacteristics/heatProduction**

| | |
|---|---|
| diagram |  |
| type | **inerbim:tStateEnergyValue** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **state** **energyValue** |
| source | <element name="heatProduction" type="inerbim:tStateEnergyValue" minOccurs="0" maxOccurs="unbounded"/> |

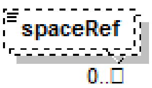
complexType **tGroupOfSpaces**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | **spaceGroupName** **spaceRef** |
| used by | element **tBuilding/group** |
| source | <complexType name="tGroupOfSpaces"><br><sequence><br><element name="spaceGroupName" type="string" minOccurs="1" maxOccurs="1"/><br><element name="spaceRef" type="string" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

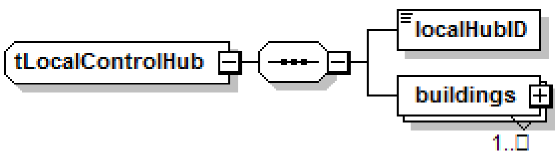
element **tGroupOfSpaces/spaceGroupName**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="spaceGroupName" type="string" minOccurs="1" maxOccurs="1"/> |

element **tGroupOfSpaces/spaceRef**

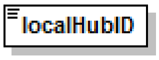
| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="spaceRef" type="string" minOccurs="0" maxOccurs="unbounded"/> |

complexType **tLocalControlHub**

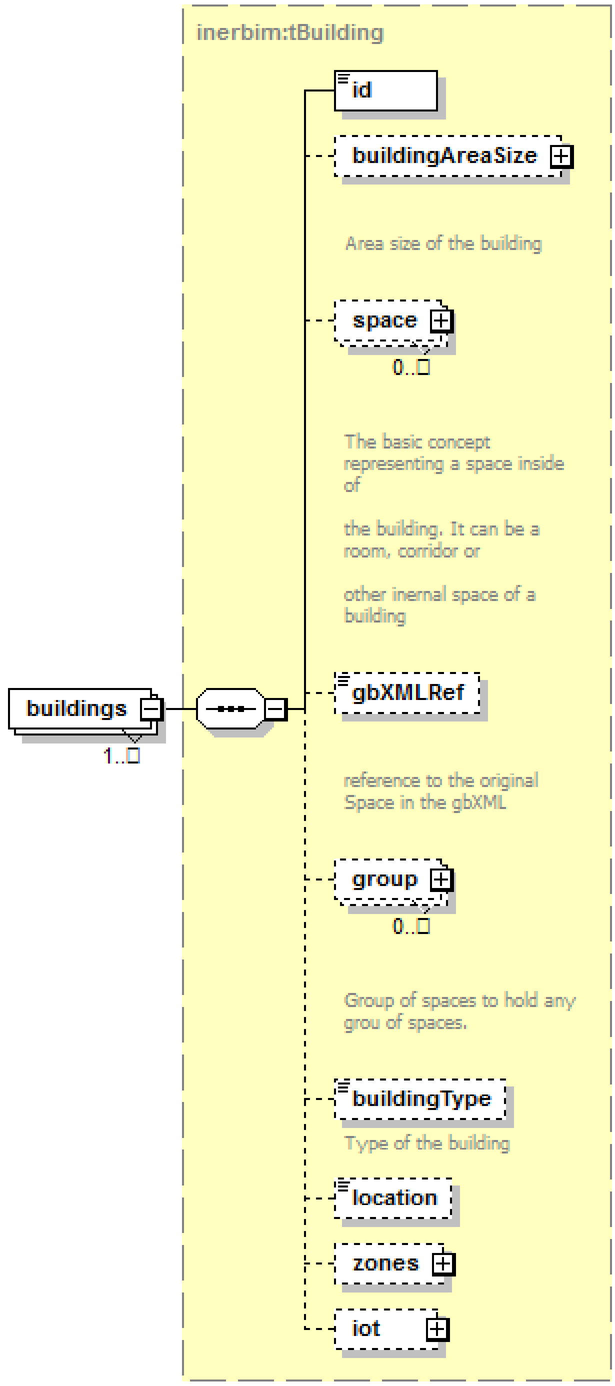
| | |
|---|---|
| diagram |  |

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | **localHubID** **buildings** |
| used by | element **tAggregatorControlHub/localHubs** |
| source | <complexType name="tLocalControlHub"><br><sequence><br><element name="localHubID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="buildings" type="inerbim:tBuilding" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType> |

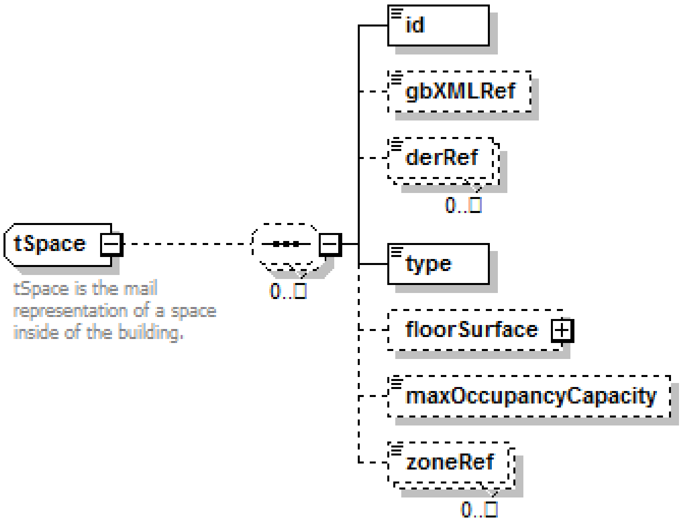
element **tLocalControlHub/localHubID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="localHubID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tLocalControlHub/buildings**

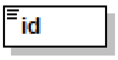
| | |
|---|---|
| diagram |  |
| type | **inerbim:tBuilding** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **id buildingAreaSize space gbXMLRef group buildingType location zones iot** |
| source | <element name="buildings" type="inerbim:tBuilding" minOccurs="1" maxOccurs="unbounded"/> |

complexType **tSpace**

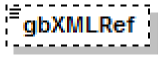
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | **id gbXMLRef derRef type floorSurface maxOccupancyCapacity zoneRef** |
| used by | element **tBuilding/space** |
| annotation | documentation<br>tSpace is the mail representation of a space inside of the building. |
| source | see below |

```
<complexType name="tSpace">
  <annotation>
    <documentation>tSpace is the mail representation of a space inside of the building.</documentation>
  </annotation>
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="gbXMLRef" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="derRef" type="string" minOccurs="0" maxOccurs="unbounded"/>
    <!--<element name="sensorDeviceRef" type="string" maxOccurs="unbounded" minOccurs="0"></element>-->
    <!--<element name="sensorDevice" type="inerDevice:tSensor" maxOccurs="unbounded" minOccurs="0"></element>-->
    <element name="type" type="ineruni:spaceTypeEnum" minOccurs="1" maxOccurs="1"/>
    <element name="floorSurface" type="ineruni:tAreaValue" minOccurs="0" maxOccurs="1"/>
    <element name="maxOccupancyCapacity" type="int" minOccurs="0" maxOccurs="1"/>
    <element name="zoneRef" type="string" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

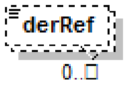
element **tSpace/id**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | `<element name="id" type="string" minOccurs="1" maxOccurs="1"/>` |

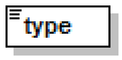
element **tSpace/gbXMLRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="gbXMLRef" type="string" minOccurs="0" maxOccurs="1"/> |

element **tSpace/derRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="derRef" type="string" minOccurs="0" maxOccurs="unbounded"/> |

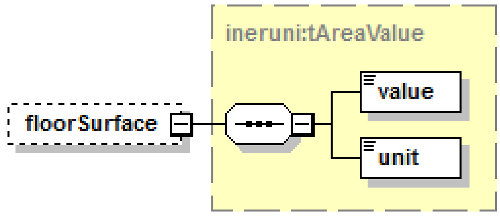
element **tSpace/type**

| | | | |
|---|---|---|---|
| diagram |  | | |
| type | **ineruni:spaceTypeEnum** | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | ActiveStorage | |
| | enumeration | ActiveStorageHospitalOrHealthcare | |
| | enumeration | AirOrTrainOrBusBaggageArea | |
| | enumeration | AirportConcourse | |
| | enumeration | AtriumEachAdditionalFloor | |
| | enumeration | AtriumFirstThreeFloors | |
| | enumeration | AudienceOrSeatingAreaPenitentiary | |
| | enumeration | AudienceOrSeatingAreaExerciseCenter | |
| | enumeration | AudienceOrSeatingAreaGymnasium | |
| | enumeration | AudienceOrSeatingAreaSportsArena | |
| | enumeration | AudienceOrSeatingAreaConventionCenter | |
| | enumeration | AudienceOrSeatingAreaMotionPictureTheatre | |
| | enumeration | AudienceOrSeatingAreaPerformingArtsTheatre | |
| | enumeration | AudienceOrSeatingAreaReligious | |
| | enumeration | AudienceOrSeatingAreaPoliceOrFireStations | |
| | enumeration | AudienceOrSeatingAreaCourtHouse | |
| | enumeration | AudienceOrSeatingAreaAuditorium | |
| | enumeration | BankCustomerArea | |
| | enumeration | BankingActivityAreaOffice | |
| | enumeration | BarberAndBeautyParlor | |
| | enumeration | CardFileAndCataloguingLibrary | |
| | enumeration | ClassroomOrLectureOrTrainingPenitentiary | |
| | enumeration | ClassroomOrLectureOrTraining | |
| | enumeration | ConfinementCellsPenitentiary | |
| | enumeration | ConfinementCellsCourtHouse | |
| | enumeration | ConferenceMeetingOrMultipurpose | |
| | enumeration | CorridorOrTransition | |
| | enumeration | CorridorOrTransitionManufacturingFacility | |
| | enumeration | CorridorsWithPatientWaitingExamHospitalOrHealthcare | |
| | enumeration | CourtSportsAreaSportsArena | |

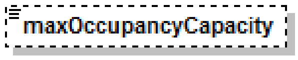
| | | |
|---|---|---|
| | enumeration | CourtroomCourtHouse |
| | enumeration | DepartmentStoreSalesAreaRetail |
| | enumeration | DetailedManufacturingFacility |
| | enumeration | DiningArea |
| | enumeration | DiningAreaHotel |
| | enumeration | DiningAreaFamilyDining |
| | enumeration | DiningAreaLoungeOrLeisureDining |
| | enumeration | DiningAreaMotel |
| | enumeration | DiningAreaTransportation |
| | enumeration | DiningAreaPenitentiary |
| | enumeration | DiningAreaCivilServices |
| | enumeration | DormitoryBedroom |
| | enumeration | DormitoryStudyHall |
| | enumeration | DressingOrLockerOrFittingRoomGymnasium |
| | enumeration | DressingOrLockerOrFittingRoomCourtHouse |
| | enumeration | DressingOrLockerOrFittingRoomPerformingArtsTheatre |
| | enumeration | DressingOrLockerOrFittingRoomAuditorium |
| | enumeration | DressingOrLockerOrFittingRoomExerciseCenter |
| | enumeration | ElectricalOrMechanical |
| | enumeration | ElevatorLobbies |
| | enumeration | EmergencyHospitalOrHealthcare |
| | enumeration | EquipmentRoomManufacturingFacility |
| | enumeration | ExamOrTreatmentHospitalOrHealthcare |
| | enumeration | ExcerciseAreaExerciseCenter |
| | enumeration | ExcerciseAreaGymnasium |
| | enumeration | ExhibitSpaceConventionCenter |
| | enumeration | FellowshipHallReligiousBuildings |
| | enumeration | FineMaterialWarehouse |
| | enumeration | FineMerchandiseSalesAreaRetail |
| | enumeration | FireStationEngineRoomPoliceOrFireStation |
| | enumeration | FoodPreparation |
| | enumeration | GarageServiceOrRepairAutomotiveFacility |
| | enumeration | GeneralHighBayManufacturingFacility |
| | enumeration | GeneralLowBayManufacturingFacility |
| | enumeration | GeneralExhibitionMuseum |
| | enumeration | HospitalNurseryHospitalOrHealthcare |
| | enumeration | HospitalOrMedicalSuppliesHospitalOrHealthcare |
| | enumeration | HospitalOrRadiologyHospitalOrHealthcare |
| | enumeration | HotelOrConferenceCenterConferenceOrMeeting |
| | enumeration | InactiveStorage |
| | enumeration | JudgesChambersCourtHouse |
| | enumeration | LaboratoryOffice |
| | enumeration | LaundryIroningAndSorting |
| | enumeration | LaundryWashingHospitalOrHealthcare |
| | enumeration | LibraryAudioVisualLibraryAudioVisual |
| | enumeration | LivingQuartersDormitory |
| | enumeration | LivingQuartersMotel |
| | enumeration | LivingQuartersHotel |
| | enumeration | Lobby |
| | enumeration | LobbyReligiousBuildings |
| | enumeration | LobbyMotionPictureTheatre |

| | | |
|---|---|---|
| | enumeration | LobbyAuditorium |
| | enumeration | LobbyPerformingArtsTheatre |
| | enumeration | LobbyPostOffice |
| | enumeration | LobbyHotel |
| | enumeration | LoungeOrRecreation |
| | enumeration | MallConcourseSalesAreaRetail |
| | enumeration | MassMerchandisingSalesAreaRetail |
| | enumeration | MediumOrBulkyMaterialWarehouse |
| | enumeration | MerchandisingSalesAreaRetail |
| | enumeration | MuseumAndGalleryStorage |
| | enumeration | NurseStationHospitalOrHealthcare |
| | enumeration | OfficeEnclosed |
| | enumeration | OfficeOpenPlan |
| | enumeration | OfficeCommonActivityAreasInactiveStorage |
| | enumeration | OperatingRoomHospitalOrHealthcare |
| | enumeration | OtherTelevisedPlayingAreaSportsArena |
| | enumeration | ParkingAreaAttendantOnlyParkingGarage |
| | enumeration | ParkingAreaPedestrianParkingGarage |
| | enumeration | PatientRoomHospitalOrHealthcare |
| | enumeration | PersonalServicesSalesAreaRetail |
| | enumeration | PharmacyHospitalOrHealthcare |
| | enumeration | PhysicalTherapyHospitalOrHealthcare |
| | enumeration | PlayingAreaGymnasium |
| | enumeration | Plenum |
| | enumeration | PoliceStationLaboratoryPoliceOrFireStations |
| | enumeration | PublicAndStaffLoungeHospitalOrHealthcare |
| | enumeration | ReadingAreaLibrary |
| | enumeration | ReceptionOrWaitingTransportation |
| | enumeration | ReceptionOrWaitingMotel |
| | enumeration | ReceptionOrWaitingHotel |
| | enumeration | RecoveryHospitalOrHealthcare |
| | enumeration | RestorationMuseum |
| | enumeration | Restrooms |
| | enumeration | RingSportsAreaSportsArena |
| | enumeration | ServerRoom |
| | enumeration | SleepingQuartersPoliceOrFireStation |
| | enumeration | SortingAreaPostOffice |
| | enumeration | SpecialtyStoreSalesAreaRetail |
| | enumeration | StacksLibrary |
| | enumeration | StairsInactive |
| | enumeration | Stairway |
| | enumeration | SupermarketSalesAreaRetail |
| | enumeration | TerminalTicketCounterTransportation |
| | enumeration | WorkshopWorkshop |
| | enumeration | WorshipPulpitChoirReligious |
| source | <element name="type" type="ineruni:spaceTypeEnum" minOccurs="1" maxOccurs="1"/> | |

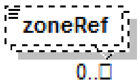
element **tSpace/floorSurface**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tAreaValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="floorSurface" type="ineruni:tAreaValue" minOccurs="0" maxOccurs="1"/> |

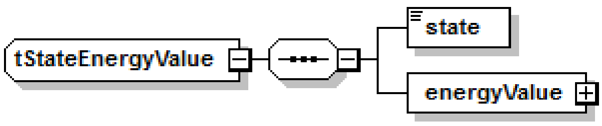
element **tSpace/maxOccupancyCapacity**

| | |
|---|---|
| diagram |  |
| type | **int** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="maxOccupancyCapacity" type="int" minOccurs="0" maxOccurs="1"/> |

element **tSpace/zoneRef**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | minOcc 0<br>maxOcc unbounded<br>content simple |
| source | <element name="zoneRef" type="string" minOccurs="0" maxOccurs="unbounded"/> |

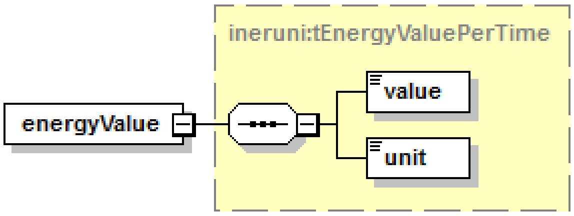
complexType **tStateEnergyValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/bim/ |
| children | **state energyValue** |
| used by | elements **tEquipmentTypeCharacteristics/energyConsumptionPerTime**<br>**tEquipmentTypeCharacteristics/heatProduction** |
| source | <complexType name="tStateEnergyValue"><br><sequence><br><element name="state" type="string" minOccurs="1" maxOccurs="1"/><br><element name="energyValue" type="ineruni:tEnergyValuePerTime" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tStateEnergyValue/state**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="state" type="string" minOccurs="1" maxOccurs="1"/> |

element **tStateEnergyValue/energyValue**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValuePerTime** |
| properties | content complex |
| children | **value unit** |
| source | <element name="energyValue" type="ineruni:tEnergyValuePerTime" minOccurs="1" maxOccurs="1"/> |

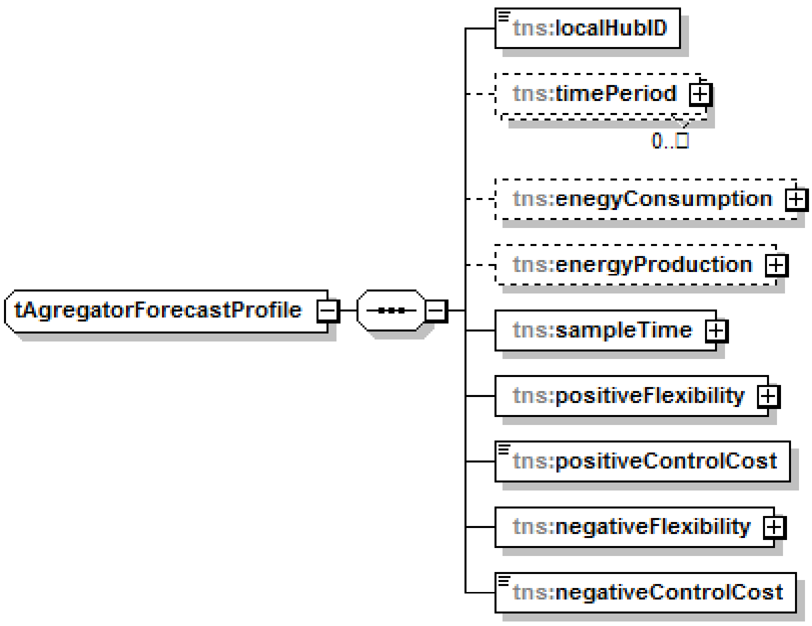
XML Schema documentation generated by **[XMLSpy](#)** Schema Editor **[http://www.altova.com/xmlspy](http://www.altova.com/xmlspy)**

# Schema inertiaMASHolisticFlexibilityAndProsumersControl.xsd

| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaMASHolisticFlexibilityAndProsumersControl.xsd** |
| attributeFormDefault: | |
| elementFormDefault: | **qualified** |
| targetNamespace: | **http://www.example.org/inertiaProsumersControlAndOptimization** |

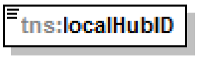
| Complex types | Attr. groups |
|---|---|
| **tAgregatorForecastProfile** | **NewAttributeGroup** |
| **tLocalHubForecastProfile** | |

complexType **tAgregatorForecastProfile**

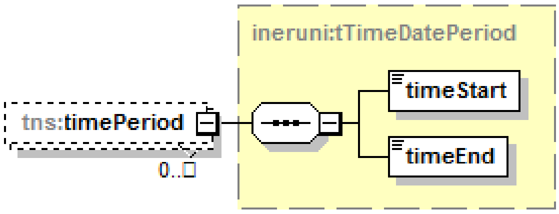
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| children | **tns:localHubID tns:timePeriod tns:enegyConsumption tns:energyProduction tns:sampleTime tns:positiveFlexibility tns:positiveControlCost tns:negativeFlexibility tns:negativeControlCost** |

| | |
|---|---|
| source | <complexType name="tAgregatorForecastProfile"><br><sequence><br><element name="localHubID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="unbounded"/><br><element name="enegyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/><br><element name="energyProduction" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/><br><element name="sampleTime" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/><br><element name="positiveFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/><br><element name="positiveControlCost" type="float" minOccurs="1" maxOccurs="1"/><br><element name="negativeFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/><br><element name="negativeControlCost" type="float" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

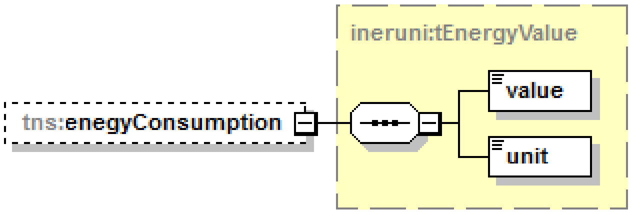
element **tAgregatorForecastProfile/localHubID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **string** |
| properties | content simple |
| source | <element name="localHubID" type="string" minOccurs="1" maxOccurs="1"/> |

element **tAgregatorForecastProfile/timePeriod**

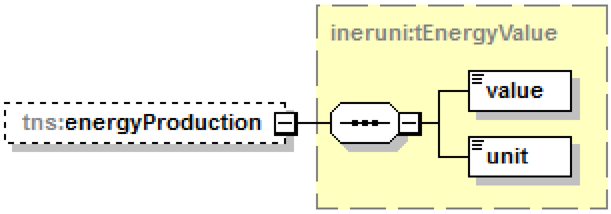
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="unbounded"/> |

element **tAgregatorForecastProfile/enegyConsumption**

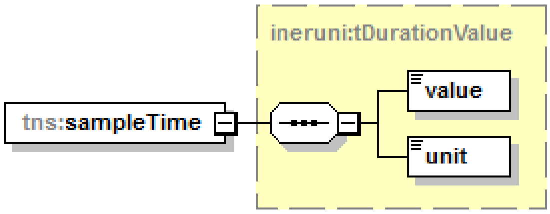
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |

| | |
|---|---|
| source | <element name="enegyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

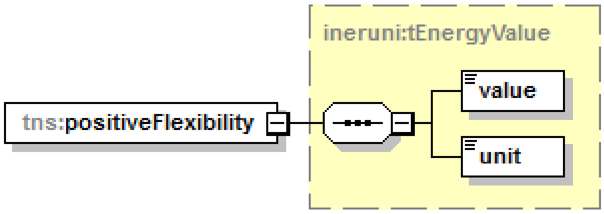
element **tAgregatorForecastProfile/energyProduction**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="energyProduction" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tAgregatorForecastProfile/sampleTime**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tDurationValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="sampleTime" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/> |

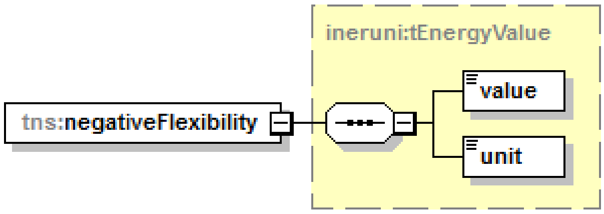
element **tAgregatorForecastProfile/positiveFlexibility**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="positiveFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/> |

element **tAgregatorForecastProfile/positiveControlCost**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **float** |
| properties | content simple |
| source | <element name="positiveControlCost" type="float" minOccurs="1" maxOccurs="1"/> |

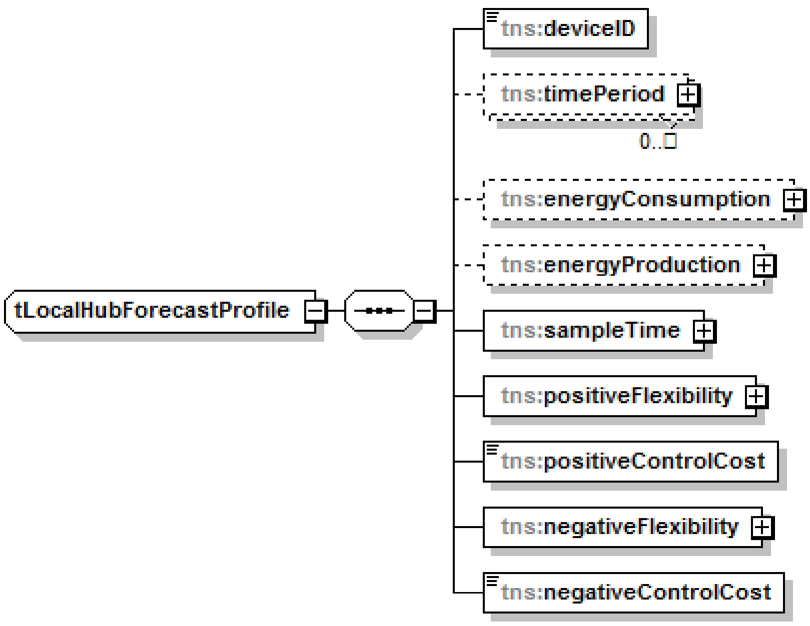
element **tAgregatorForecastProfile/negativeFlexibility**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="negativeFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/> |

element **tAgregatorForecastProfile/negativeControlCost**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **float** |
| properties | content simple |
| source | <element name="negativeControlCost" type="float" minOccurs="1" maxOccurs="1"/> |

complexType **tLocalHubForecastProfile**

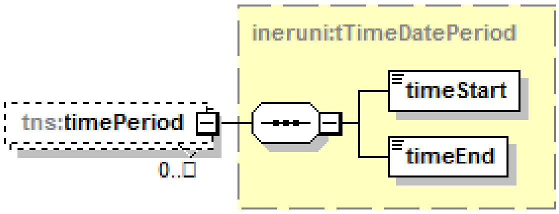
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| children | **tns:deviceID tns:timePeriod tns:energyConsumption tns:energyProduction tns:sampleTime tns:positiveFlexibility tns:positiveControlCost tns:negativeFlexibility tns:negativeControlCost** |
| source | <complexType name="tLocalHubForecastProfile"><br><sequence><br><element name="deviceID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="unbounded"/><br><element name="energyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/><br><element name="energyProduction" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/><br><element name="sampleTime" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/><br><element name="positiveFlexibility" type="ineruni:tEnergyValue" |

| | |
|---|---|
| | minOccurs="1" maxOccurs="1"/> <element name="positiveControlCost" type="float" minOccurs="1" maxOccurs="1"/> <element name="negativeFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/> <element name="negativeControlCost" type="float" minOccurs="1" maxOccurs="1"/> </sequence> </complexType> |

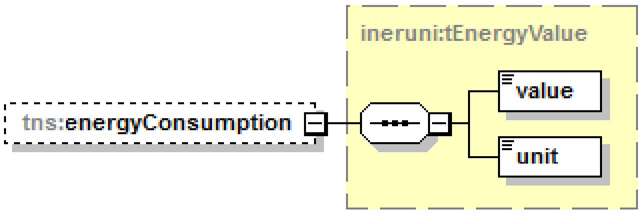
element **tLocalHubForecastProfile/deviceID**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **string** |
| properties | content simple |
| source | <element name="deviceID" type="string" minOccurs="1" maxOccurs="1"/> |

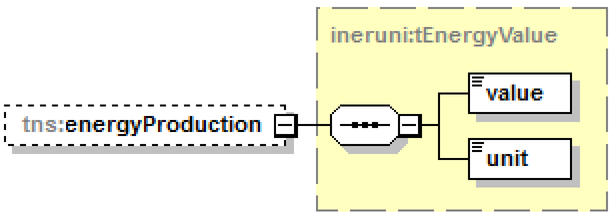
element **tLocalHubForecastProfile/timePeriod**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="timePeriod" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="unbounded"/> |

element **tLocalHubForecastProfile/energyConsumption**

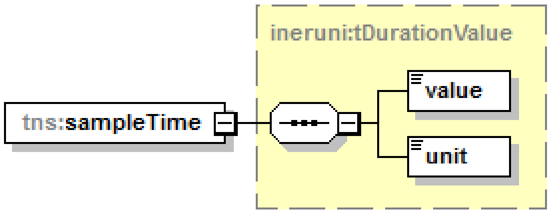
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="energyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tLocalHubForecastProfile/energyProduction**

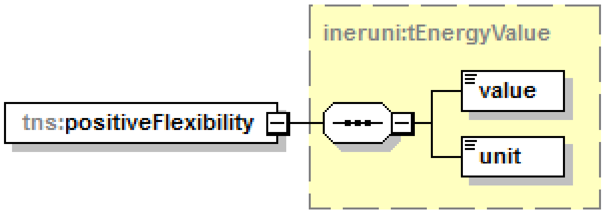
| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |

| | |
|---|---|
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="energyProduction" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tLocalHubForecastProfile/sampleTime**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tDurationValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="sampleTime" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/> |

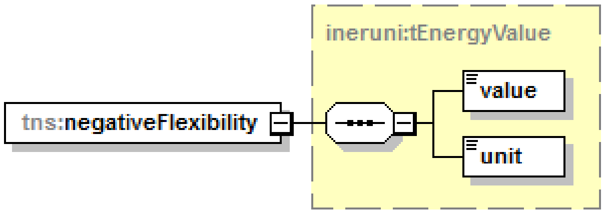
element **tLocalHubForecastProfile/positiveFlexibility**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="positiveFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/> |

element **tLocalHubForecastProfile/positiveControlCost**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **float** |
| properties | content simple |
| source | <element name="positiveControlCost" type="float" minOccurs="1" maxOccurs="1"/> |

element **tLocalHubForecastProfile/negativeFlexibility**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **ineruni:tEnergyValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="negativeFlexibility" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="1"/> |

element **tLocalHubForecastProfile/negativeControlCost**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| type | **float** |
| properties | content simple |
| source | <element name="negativeControlCost" type="float" minOccurs="1" maxOccurs="1"/> |

attributeGroup **NewAttributeGroup**

| | |
|---|---|
| namespace | http://www.example.org/inertiaProsumersControlAndOptimization |
| source | <attributeGroup name="NewAttributeGroup"/> |

XML Schema documentation generated by **XMLSpy** Schema Editor **http://www.altova.com/xmlspy**

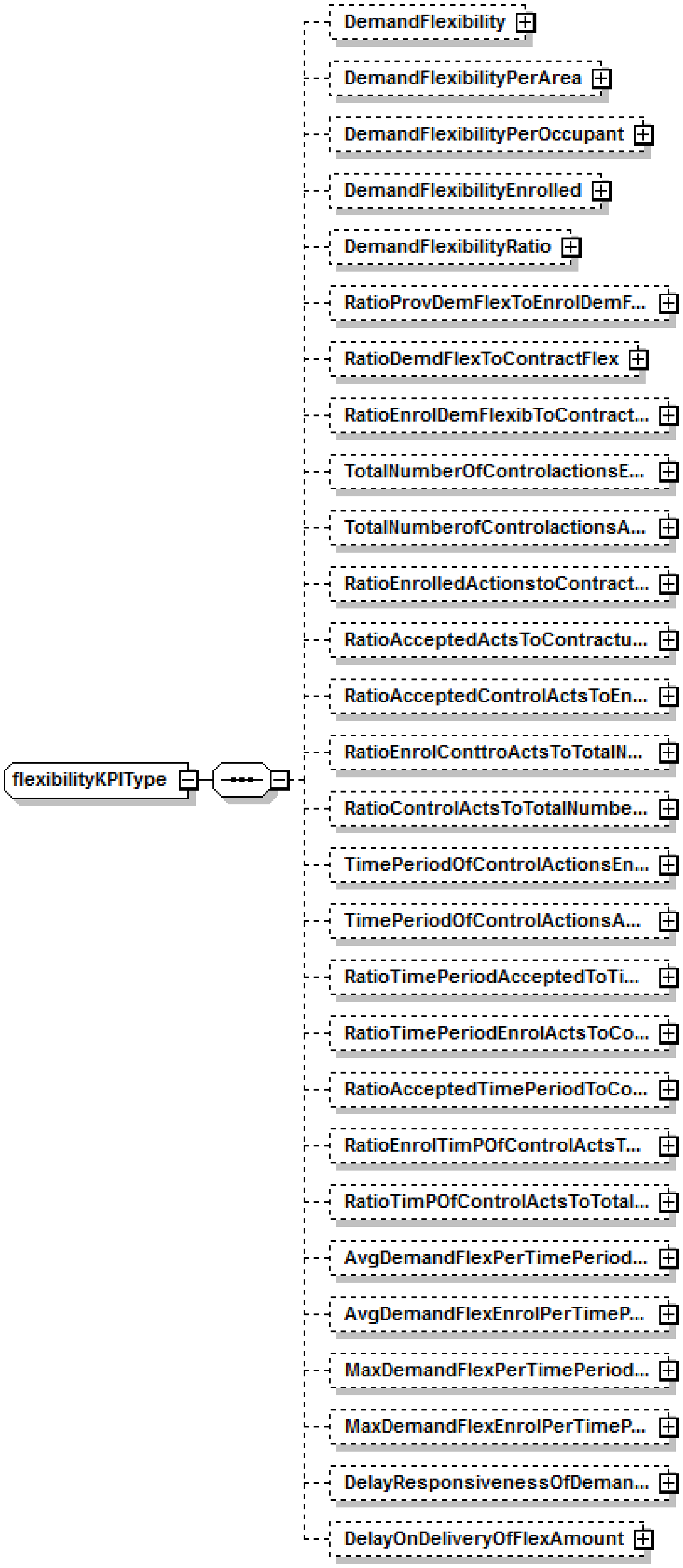
## Schema inertiaKPI.xsd

| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaKPI.xsd** |
| attributeFormDefault: | |
| elementFormDefault: | |
| targetNamespace: | **http://www.inertia.eu/2013/schema/kpi/** |

| Complex types | Simple types |
|---|---|
| **flexibilityKPIType** | **ActivityEnumerator** |
| **gridKPIType** | |
| **tAgentBasedSimulationModuleOutput** | |
| **tEventTrace** | |
| **tKeyPerformanceIndicator** | |

complexType **flexibilityKPIType**

diagram




| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/kpi/ |
| children | **DemandFlexibility** **DemandFlexibilityPerArea** **DemandFlexibilityPerOccupant** **DemandFlexibilityEnrolled** **DemandFlexibilityRatio** **RatioProvDemFlexToEnrolDemFlex** **RatioDemdFlexToContractFlex** **RatioEnrolDemFlexibToContractFlex** **TotalNumberOfControlactionsEnrolled** **TotalNumberofControlactionsAccepted** |

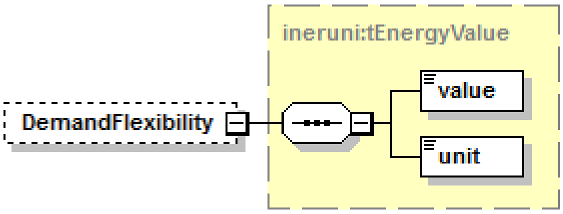
| | |
|---|---|
| | RatioEnrolledActionstoContractualActions RatioAcceptedActsToContractualActsAgreement RatioAcceptedControlActsToEnrolledControlActs RatioEnrolConttroActsToTotalNumOfEnrolControActs RatioControlActsToTotalNumberofControlActs TimePeriodOfControlActionsEnrolled TimePeriodOfControlActionsAccepted RatioTimePeriodAcceptedToTimeperiodEnrolled RatioTimePeriodEnrolActsToContractTimePeriod RatioAcceptedTimePeriodToContractTimePeriod RatioEnrolTimPOfControlActsToTotalTimPOfEnrolControlActs RatioTimPOfControlActsToTotalTimPOfControlActs AvgDemandFlexPerTimePeriodOfEvent AvgDemandFlexEnrolPerTimePeriodOfEvent MaxDemandFlexPerTimePeriodOfEvent MaxDemandFlexEnrolPerTimePeriodOfEvent DelayResponsivenessOfDemandFlex DelayOnDeliveryOfFlexAmount |
| used by | element tKeyPerformanceIndicator/flexibilityKPI |
| source | (see below) |

source:

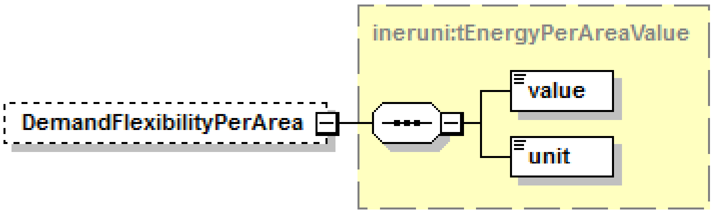
```
<complexType name="flexibilityKPIType">
  <sequence>
    <element name="DemandFlexibility" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
    <element name="DemandFlexibilityPerArea" type="ineruni:tEnergyPerAreaValue" minOccurs="0" maxOccurs="1"/>
    <element name="DemandFlexibilityPerOccupant" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
    <element name="DemandFlexibilityEnrolled" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
    <element name="DemandFlexibilityRatio" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioProvDemFlexToEnrolDemFlex" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioDemdFlexToContractFlex" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioEnrolDemFlexibToContractFlex" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="TotalNumberOfControlactionsEnrolled" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="TotalNumberofControlactionsAccepted" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioEnrolledActionstoContractualActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioAcceptedActsToContractualActsAgreement" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioAcceptedControlActsToEnrolledControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioEnrolConttroActsToTotalNumOfEnrolControActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioControlActsToTotalNumberofControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="TimePeriodOfControlActionsEnrolled" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/>
    <element name="TimePeriodOfControlActionsAccepted" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/>
    <element name="RatioTimePeriodAcceptedToTimeperiodEnrolled" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioTimePeriodEnrolActsToContractTimePeriod" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioAcceptedTimePeriodToContractTimePeriod" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element
name="RatioEnrolTimPOfControlActsToTotalTimPOfEnrolControlActs"
type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioTimPOfControlActsToTotalTimPOfControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="AvgDemandFlexPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/>
    <element name="AvgDemandFlexEnrolPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/>
    <element name="MaxDemandFlexPerTimePeriodOfEvent"
```

| | |
|---|---|
| | type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/><br><element name="MaxDemandFlexEnrolPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/><br><element name="DelayResponsivenessOfDemandFlex" type="ineruni:tDurationValue" minOccurs="0" maxOccurs="1"/><br><element name="DelayOnDeliveryOfFlexAmount" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

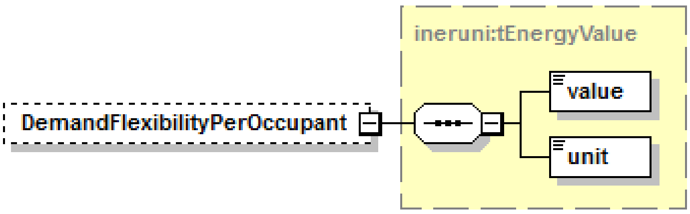
element **flexibilityKPIType/DemandFlexibility**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DemandFlexibility" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/DemandFlexibilityPerArea**

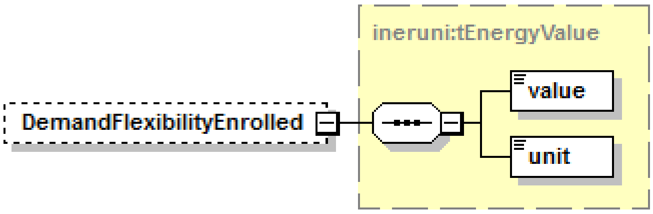
| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyPerAreaValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DemandFlexibilityPerArea" type="ineruni:tEnergyPerAreaValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/DemandFlexibilityPerOccupant**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DemandFlexibilityPerOccupant" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/DemandFlexibilityEnrolled**

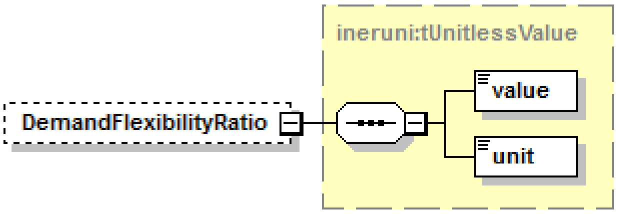
diagram




| | |
|---|---|
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DemandFlexibilityEnrolled" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/DemandFlexibilityRatio**

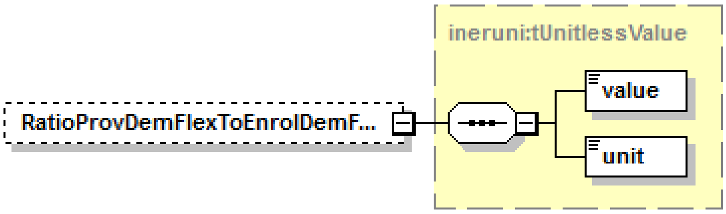
diagram




| | |
|---|---|
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DemandFlexibilityRatio" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioProvDemFlexToEnrolDemFlex**

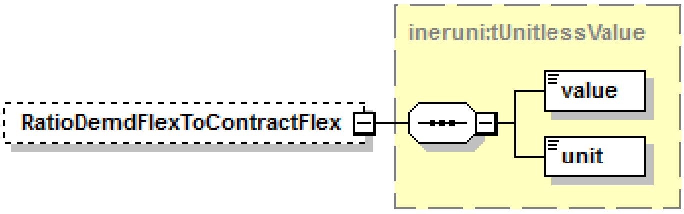
diagram




| | |
|---|---|
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioProvDemFlexToEnrolDemFlex" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioDemdFlexToContractFlex**

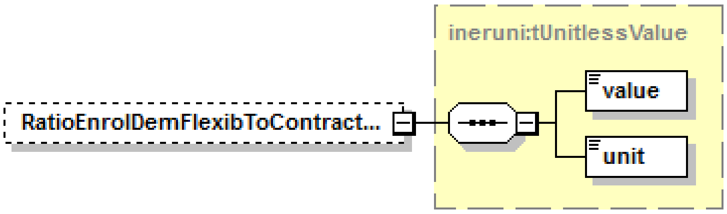
diagram



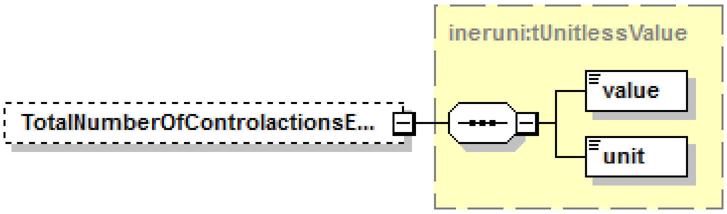

| | |
|---|---|
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioDemdFlexToContractFlex" type="ineruni:tUnitlessValue" |

| | |
|---|---|
| | minOccurs="0" maxOccurs="1"/> |

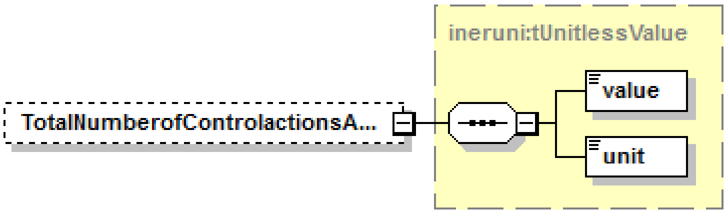
element **flexibilityKPIType/RatioEnrolDemFlexibToContractFlex**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioEnrolDemFlexibToContractFlex" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

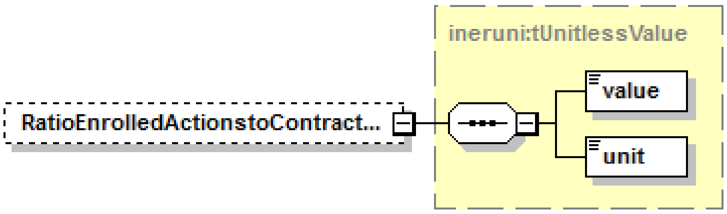
element **flexibilityKPIType/TotalNumberOfControlactionsEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="TotalNumberOfControlactionsEnrolled" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/TotalNumberofControlactionsAccepted**

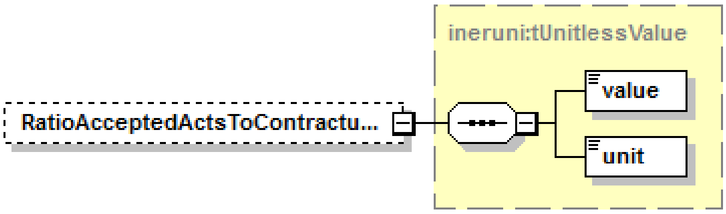
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="TotalNumberofControlactionsAccepted" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioEnrolledActionstoContractualActions**

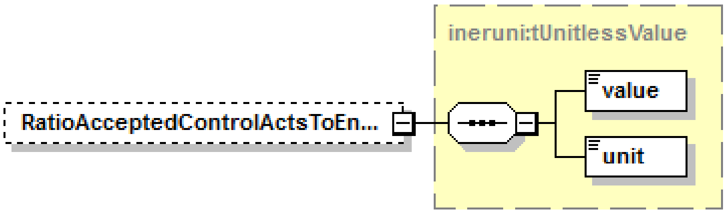
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |

| | |
|---|---|
| children | **value unit** |
| source | <element name="RatioEnrolledActionstoContractualActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

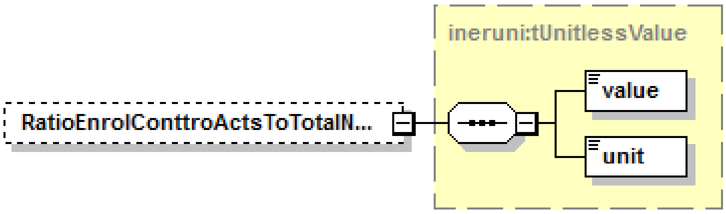
element **flexibilityKPIType/RatioAcceptedActsToContractualActsAgreement**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioAcceptedActsToContractualActsAgreement" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

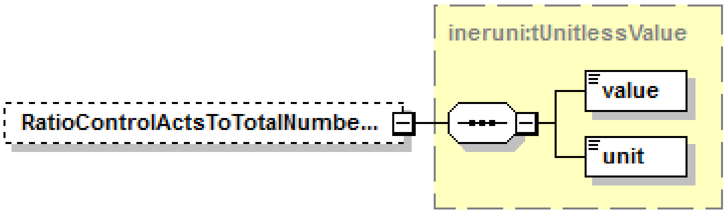
element **flexibilityKPIType/RatioAcceptedControlActsToEnrolledControlActs**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioAcceptedControlActsToEnrolledControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioEnrolConttroActsToTotalNumOfEnrolControActs**

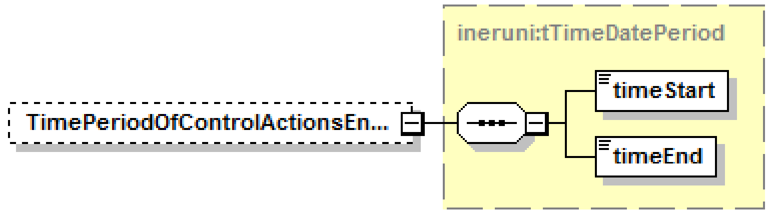
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioEnrolConttroActsToTotalNumOfEnrolControActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioControlActsToTotalNumberofControlActs**

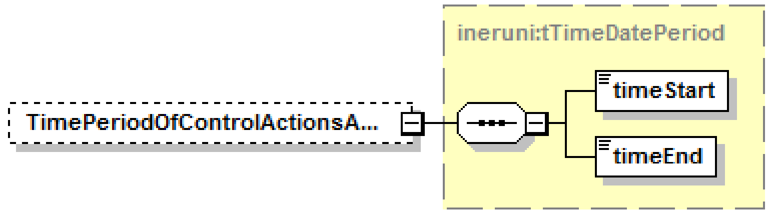
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |

| | |
|---|---|
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioControlActsToTotalNumberofControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

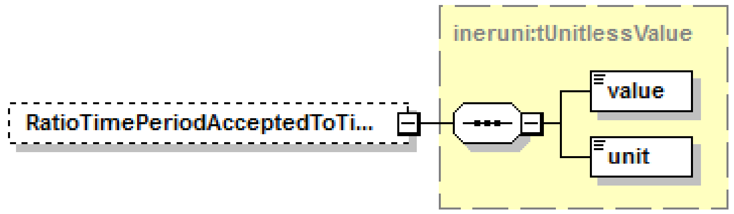
element **flexibilityKPIType/TimePeriodOfControlActionsEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="TimePeriodOfControlActionsEnrolled" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/> |

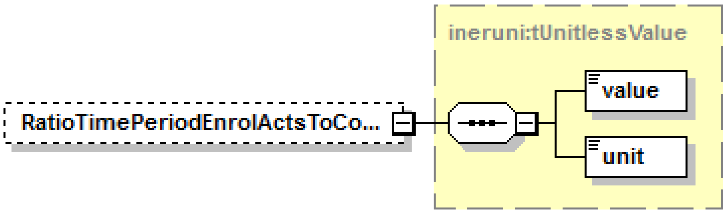
element **flexibilityKPIType/TimePeriodOfControlActionsAccepted**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeDatePeriod** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **timeStart timeEnd** |
| source | <element name="TimePeriodOfControlActionsAccepted" type="ineruni:tTimeDatePeriod" minOccurs="0" maxOccurs="1"/> |

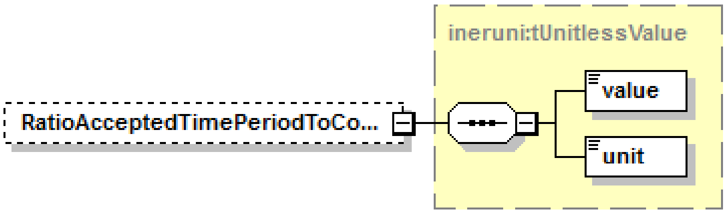
element **flexibilityKPIType/RatioTimePeriodAcceptedToTimeperiodEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioTimePeriodAcceptedToTimeperiodEnrolled" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

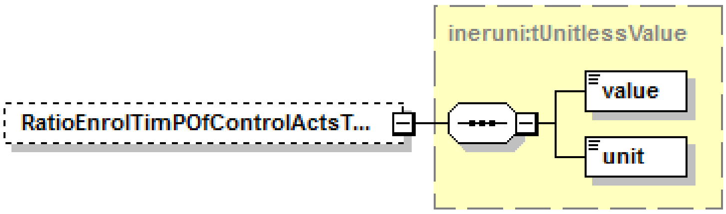
element **flexibilityKPIType/RatioTimePeriodEnrolActsToContractTimePeriod**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioTimePeriodEnrolActsToContractTimePeriod" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

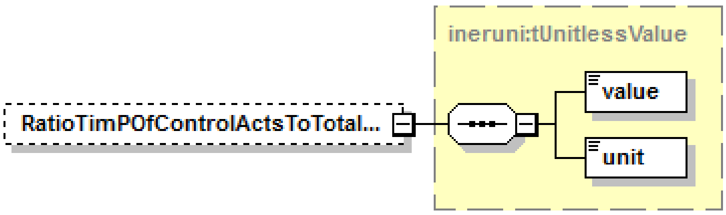
element **flexibilityKPIType/RatioAcceptedTimePeriodToContractTimePeriod**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioAcceptedTimePeriodToContractTimePeriod" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioEnrolTimPOfControlActsToTotalTimPOfEnrolControlActs**

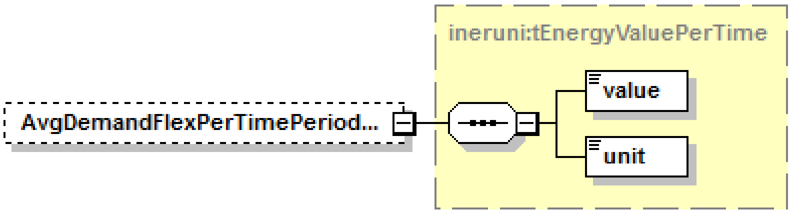
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioEnrolTimPOfControlActsToTotalTimPOfEnrolControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/RatioTimPOfControlActsToTotalTimPOfControlActs**

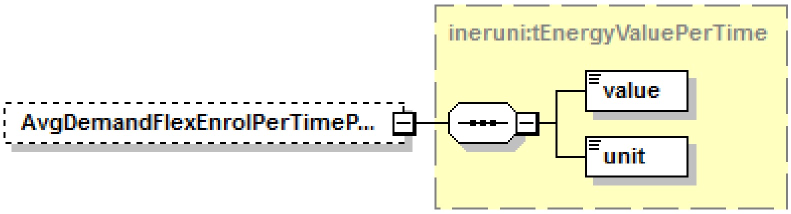
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |

| | |
|---|---|
| source | <element name="RatioTimPOfControlActsToTotalTimPOfControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

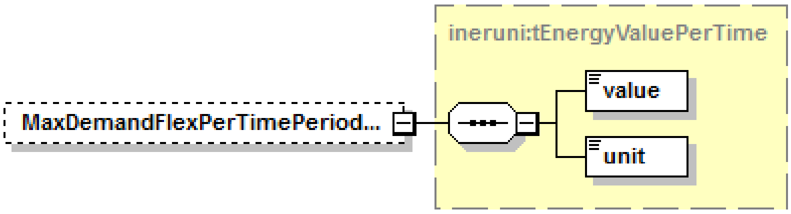
element **flexibilityKPIType/AvgDemandFlexPerTimePeriodOfEvent**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValuePerTime** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgDemandFlexPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/> |

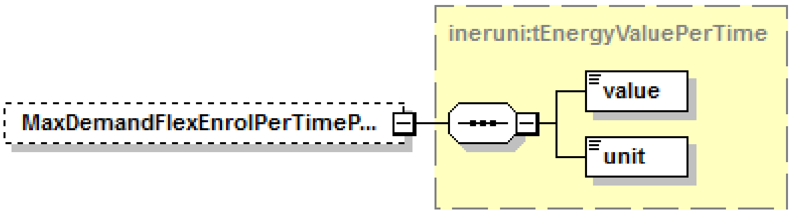
element **flexibilityKPIType/AvgDemandFlexEnrolPerTimePeriodOfEvent**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValuePerTime** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgDemandFlexEnrolPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/MaxDemandFlexPerTimePeriodOfEvent**

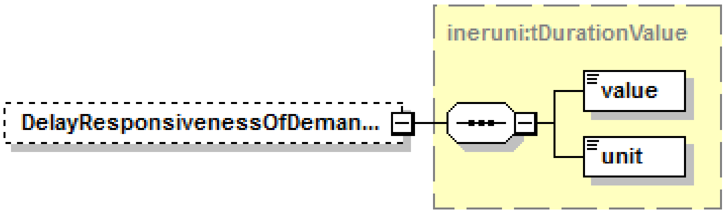
| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValuePerTime** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="MaxDemandFlexPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/MaxDemandFlexEnrolPerTimePeriodOfEvent**

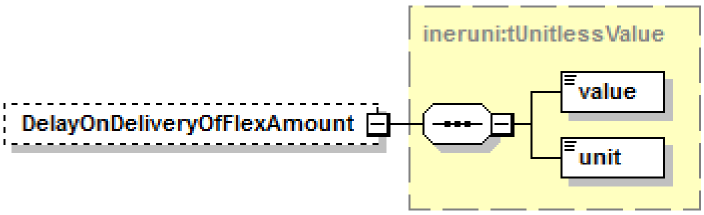
| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValuePerTime** |
| properties | minOcc 0 |

| | |
|---|---|
| | maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="MaxDemandFlexEnrolPerTimePeriodOfEvent" type="ineruni:tEnergyValuePerTime" minOccurs="0" maxOccurs="1"/> |

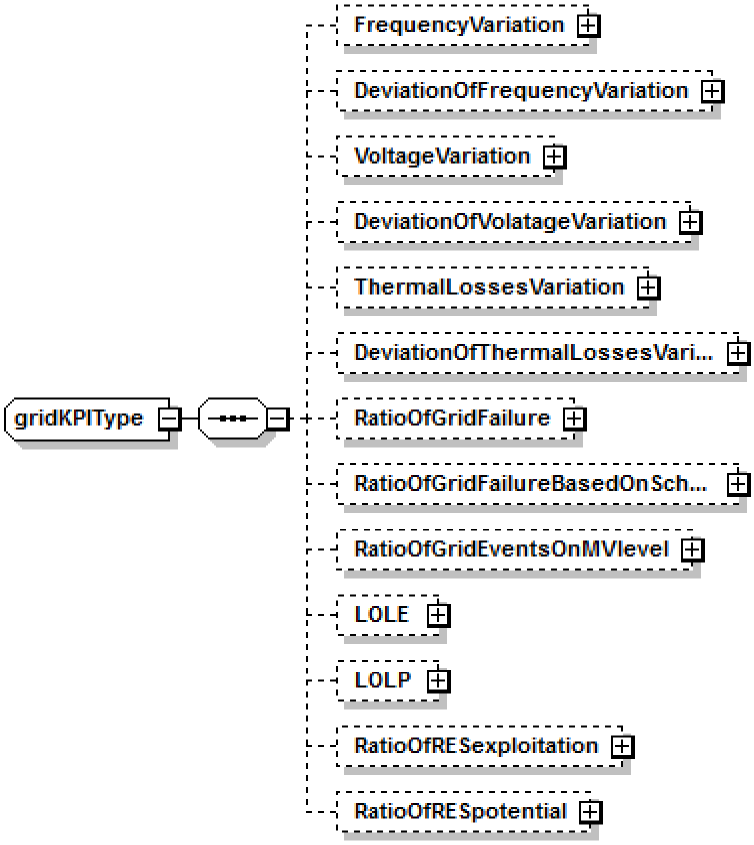
element **flexibilityKPIType/DelayResponsivenessOfDemandFlex**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tDurationValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DelayResponsivenessOfDemandFlex" type="ineruni:tDurationValue" minOccurs="0" maxOccurs="1"/> |

element **flexibilityKPIType/DelayOnDeliveryOfFlexAmount**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DelayOnDeliveryOfFlexAmount" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

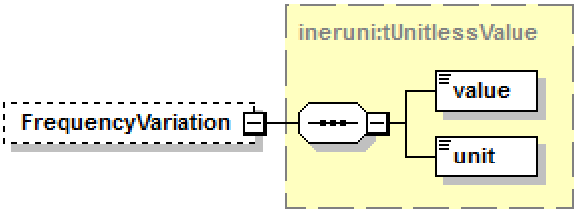
complexType **gridKPIType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/kpi/ |
| children | **FrequencyVariation** **DeviationOfFrequencyVariation** **VoltageVariation** **DeviationOfVolatageVariation** **ThermalLossesVariation** **DeviationOfThermalLossesVariation** **RatioOfGridFailure** **RatioOfGridFailureBasedOnScheduledAnalysis** **RatioOfGridEventsOnMVlevel** **LOLE** **LOLP** **RatioOfRESexploitation** **RatioOfRESpotential** |
| used by | element **tKeyPerformanceIndicator/gridKPI** |
| source | |

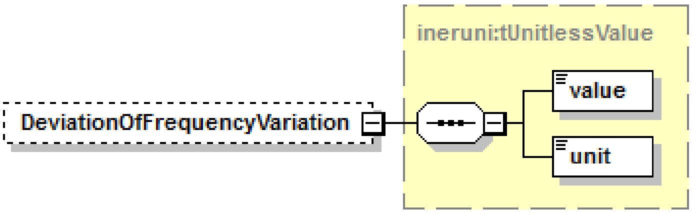
```
<complexType name="gridKPIType">
  <sequence>
    <element name="FrequencyVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="DeviationOfFrequencyVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="VoltageVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="DeviationOfVolatageVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="ThermalLossesVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="DeviationOfThermalLossesVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioOfGridFailure" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioOfGridFailureBasedOnScheduledAnalysis" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioOfGridEventsOnMVlevel" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="LOLE" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="LOLP" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioOfRESexploitation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    <element name="RatioOfRESpotential" type="ineruni:tUnitlessValue"
```

| | |
|---|---|
| | minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

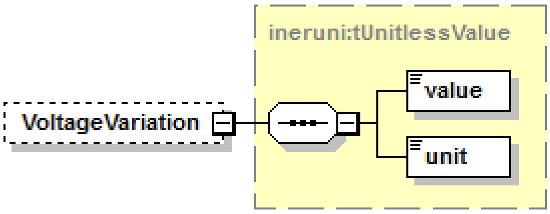
element **gridKPIType/FrequencyVariation**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="FrequencyVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

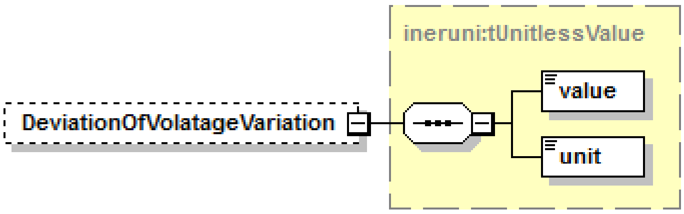
element **gridKPIType/DeviationOfFrequencyVariation**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DeviationOfFrequencyVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **gridKPIType/VoltageVariation**

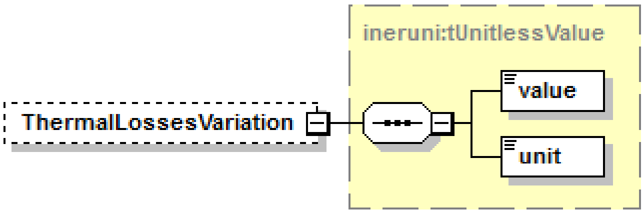
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="VoltageVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **gridKPIType/DeviationOfVolatageVariation**

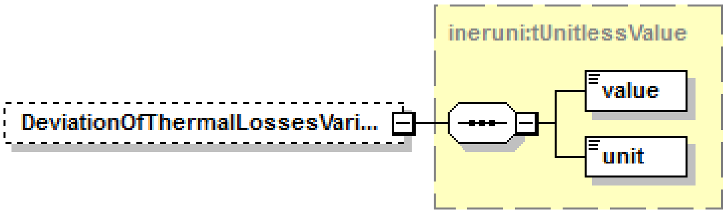
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |

| | |
|---|---|
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DeviationOfVolatageVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

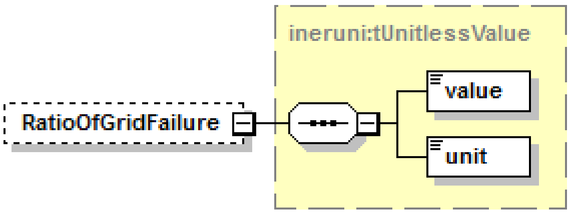
element **gridKPIType/ThermalLossesVariation**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="ThermalLossesVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

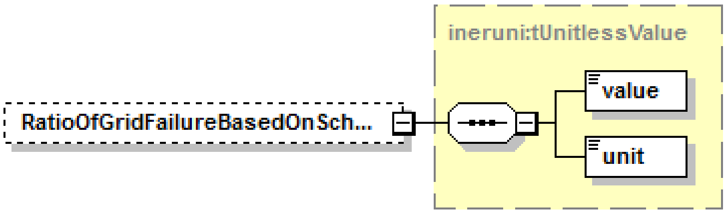
element **gridKPIType/DeviationOfThermalLossesVariation**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="DeviationOfThermalLossesVariation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

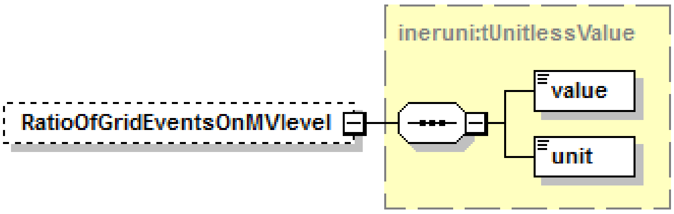
element **gridKPIType/RatioOfGridFailure**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioOfGridFailure" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

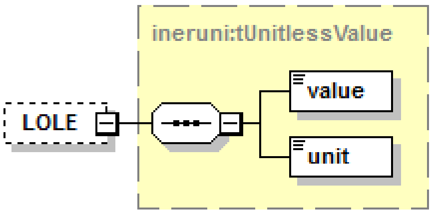
element **gridKPIType/RatioOfGridFailureBasedOnScheduledAnalysis**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioOfGridFailureBasedOnScheduledAnalysis" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

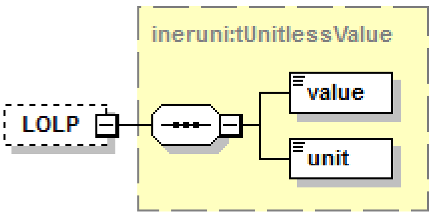
element **gridKPIType/RatioOfGridEventsOnMVlevel**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioOfGridEventsOnMVlevel" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **gridKPIType/LOLE**

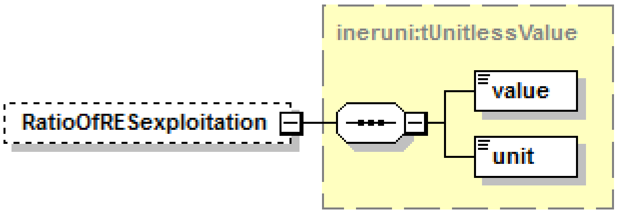
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="LOLE" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **gridKPIType/LOLP**

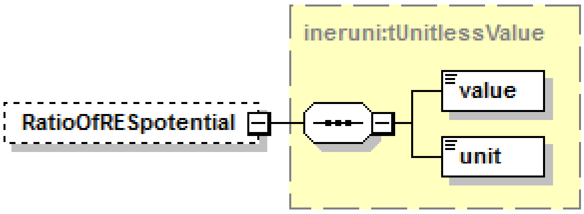
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="LOLP" type="ineruni:tUnitlessValue" minOccurs="0" |

| | |
|---|---|
| | maxOccurs="1"/> |

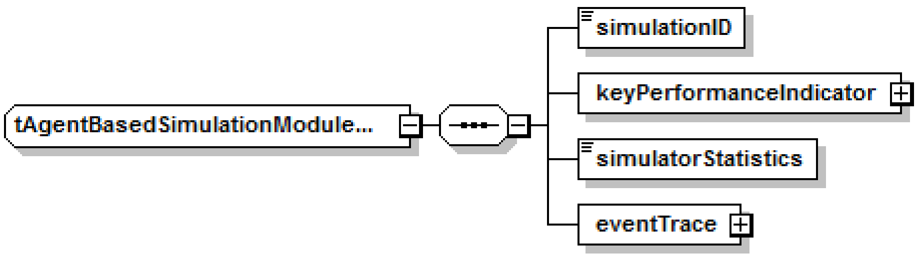
element **gridKPIType/RatioOfRESexploitation**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioOfRESexploitation" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

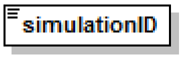
element **gridKPIType/RatioOfRESpotential**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioOfRESpotential" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

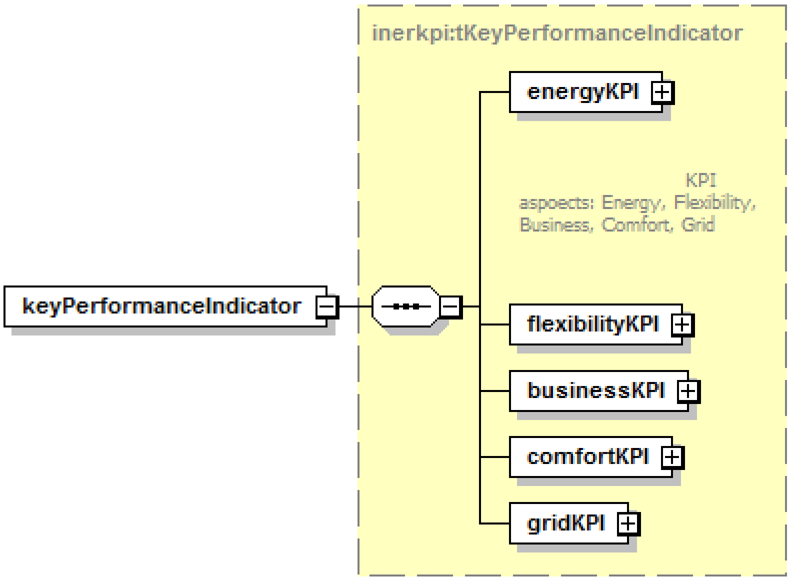
complexType **tAgentBasedSimulationModuleOutput**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/kpi/ |
| children | **simulationID keyPerformanceIndicator simulatorStatistics eventTrace** |
| source | <complexType name="tAgentBasedSimulationModuleOutput"><br><sequence><br><element name="simulationID" type="string" minOccurs="1" maxOccurs="1"/><br><element name="keyPerformanceIndicator" type="inerkpi:tKeyPerformanceIndicator" minOccurs="1" maxOccurs="1"/><br><element name="simulatorStatistics" type="string" minOccurs="1" maxOccurs="1"/><br><element name="eventTrace" type="inerkpi:tEventTrace" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tAgentBasedSimulationModuleOutput/simulationID**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="simulationID" type="string" minOccurs="1" maxOccurs="1"/> |

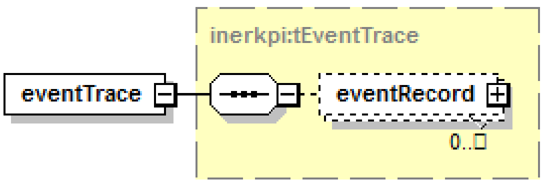
element **tAgentBasedSimulationModuleOutput/keyPerformanceIndicator**

| | |
|---|---|
| diagram |  |
| type | **inerkpi:tKeyPerformanceIndicator** |
| properties | content complex |
| children | **energyKPI** **flexibilityKPI** **businessKPI** **comfortKPI** **gridKPI** |
| source | <element name="keyPerformanceIndicator" type="inerkpi:tKeyPerformanceIndicator" minOccurs="1" maxOccurs="1"/> |

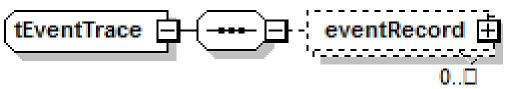
element **tAgentBasedSimulationModuleOutput/simulatorStatistics**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="simulatorStatistics" type="string" minOccurs="1" maxOccurs="1"/> |

element **tAgentBasedSimulationModuleOutput/eventTrace**

| | |
|---|---|
| diagram |  |
| type | **inerkpi:tEventTrace** |
| properties | content complex |
| children | **eventRecord** |
| source | <element name="eventTrace" type="inerkpi:tEventTrace" minOccurs="1" maxOccurs="1"/> |

complexType **tEventTrace**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/kpi/ |
| children | **eventRecord** |

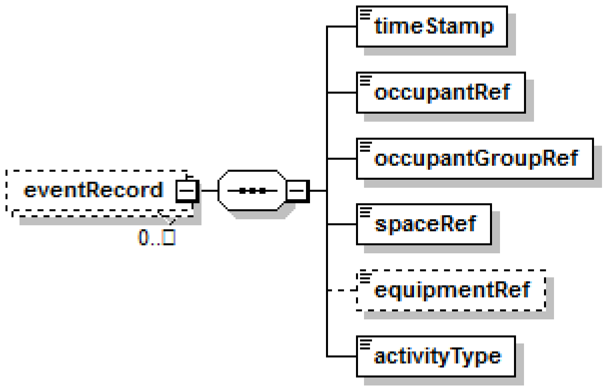
| | |
|---|---|
| used by | element **tAgentBasedSimulationModuleOutput/eventTrace** |

source

```xml
<complexType name="tEventTrace">
  <sequence>
    <element name="eventRecord" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/>
          <element name="occupantRef" type="string" minOccurs="1" maxOccurs="1"/>
          <element name="occupantGroupRef" type="string" minOccurs="1" maxOccurs="1"/>
          <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/>
          <element name="equipmentRef" type="string" minOccurs="0" maxOccurs="1"/>
          <element name="activityType" type="inerkpi:ActivityEnumerator"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

element **tEventTrace/eventRecord**

diagram




| | | |
|---|---|---|
| properties | minOcc | 0 |
| | maxOcc | unbounded |
| | content | complex |
| children | **timeStamp occupantRef occupantGroupRef spaceRef equipmentRef activityType** | |

source

```xml
<element name="eventRecord" minOccurs="0" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/>
      <element name="occupantRef" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="occupantGroupRef" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="equipmentRef" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="activityType" type="inerkpi:ActivityEnumerator"/>
    </sequence>
  </complexType>
</element>
```

element **tEventTrace/eventRecord/timeStamp**

| | |
|---|---|
| diagram | timeStamp |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/> |

element **tEventTrace/eventRecord/occupantRef**

| | |
|---|---|
| diagram | occupantRef |
| type | **string** |
| properties | content simple |
| source | <element name="occupantRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tEventTrace/eventRecord/occupantGroupRef**

| | |
|---|---|
| diagram | occupantGroupRef |
| type | **string** |
| properties | content simple |
| source | <element name="occupantGroupRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tEventTrace/eventRecord/spaceRef**

| | |
|---|---|
| diagram | spaceRef |
| type | **string** |
| properties | content simple |
| source | <element name="spaceRef" type="string" minOccurs="1" maxOccurs="1"/> |

element **tEventTrace/eventRecord/equipmentRef**

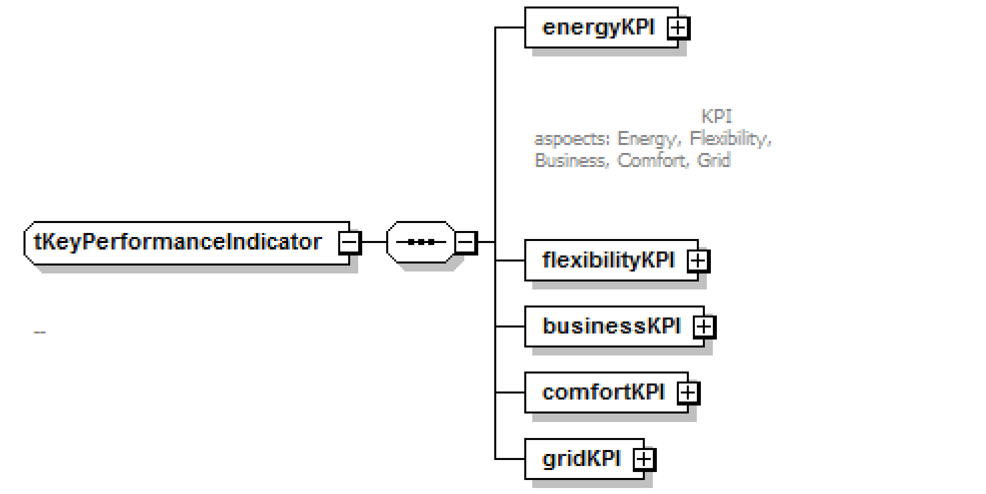
| | |
|---|---|
| diagram | equipmentRef |
| type | **string** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="equipmentRef" type="string" minOccurs="0" maxOccurs="1"/> |

element **tEventTrace/eventRecord/activityType**

| | | | |
|---|---|---|---|
| diagram | activityType | | |
| type | **inerkpi:ActivityEnumerator** | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | Occupant enters Building Element (BE corresponds to BE in Event Record) | |
| | enumeration | Occupant leaves Building Element (BE corresponds to BE in Event Record) | |
| | enumeration | Occupant starts using Equipment Artifact (EA corresponds to EA in Event Record) | |
| | enumeration | Occupants stops using Equipment Artifact (EA corresponds to EA in Event Record) | |
| | enumeration | Occupant starts executing some Business Activity | |
| | enumeration | Occupant stops executing some Business Activity | |
| | enumeration | Occupant turns on Lighting within Building Element | |
| | enumeration | Occupant turns off Lighting within Building Element | |
| source | <element name="activityType" type="inerkpi:ActivityEnumerator"/> | | |

complexType **tKeyPerformanceIndicator**

diagram




| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/kpi/ |
| children | **energyKPI** **flexibilityKPI** **businessKPI** **comfortKPI** **gridKPI** |
| used by | element **tAgentBasedSimulationModuleOutput/keyPerformanceIndicator** |
| annotation | documentation<br>-- |

source

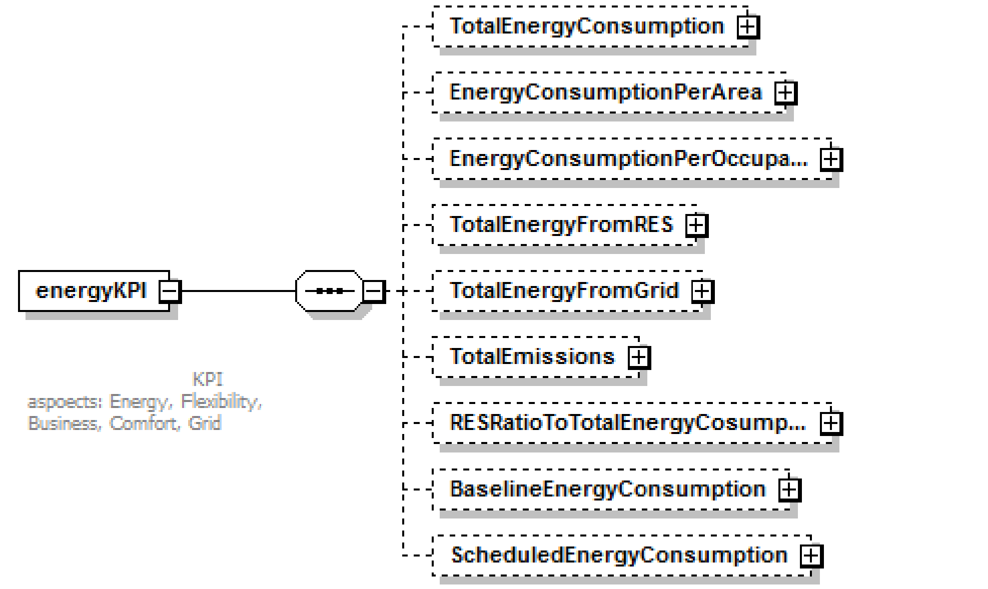
```
<complexType name="tKeyPerformanceIndicator">
  <annotation>
    <documentation>
      --
    </documentation>
  </annotation>
  <sequence>
    <element name="energyKPI" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation>
          KPI aspoects: Energy, Flexibility, Business, Comfort, Grid
        </documentation>
      </annotation>
      <complexType>
        <sequence>
          <element name="TotalEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
          <element name="EnergyConsumptionPerArea" type="ineruni:tEnergyPerAreaValue" minOccurs="0" maxOccurs="1"/>
          <element name="EnergyConsumptionPerOccupant" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
          <element name="TotalEnergyFromRES" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
          <element name="TotalEnergyFromGrid" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
          <element name="TotalEmissions" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/>
          <element name="RESRatioToTotalEnergyCosumption" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
          <element name="BaselineEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
          <element name="ScheduledEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </complexType>
    </element>
    <element name="flexibilityKPI" type="inerkpi:flexibilityKPIType"
```

```
minOccurs="1" maxOccurs="1"/>
  <element name="businessKPI" minOccurs="1" maxOccurs="1">
    <complexType>
      <sequence>
        <element name="CostOfEnergyConsumption" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
        <element name="CostOfEnergySavings" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
        <element name="CostOfEnergyConsumptionEnrolled" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
        <element name="CostOfEnergySavingstionEnrolled" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgCostOfEnergyConsumption" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgCostofConsumptionEnrolled" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgEstimationofEnergySavings" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgEstimationofEnergySavingsEnrolled" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
        <element name="CostOfEnergyReward" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgCostOfEnergyReward" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
        <element name="CostOfEnergyPenalty" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgCostOfEnergyPenalty" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
        <element name="CO2EmissionsSavings" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/>
        <element name="CO2EmissionsSavingsEnrolled" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/>
        <element name="FacilityManagerDirectControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="RatioOfFacilityManagerDirectControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      </sequence>
    </complexType>
  </element>
  <element name="comfortKPI" minOccurs="1" maxOccurs="1">
    <complexType>
      <sequence>
        <element name="ThermalDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="LightingDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="OccupancyFactor" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/>
        <element name="VariationFromStandardBasedThermalBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="VariationFromStandardBasedBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgValueOfThermalPreferences" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="AvgValueOfVisualPreferences" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
        <element name="VariationofThermPrefFromLHBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="VariationOfVisualPreferencesFromL.H.Baseline" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
        <element name="NumberOfNotAcceptedControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
```

```
        <element name="RatioOfNotAcceptedControlActionsToTotalNumber" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="RatioOfIndivNotAcceptedControlActionsToTotalNumber" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="RatioOfIndivPercentageOfNotAcceptedControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="ElectricVehiclesDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
        <element name="DishwasherDiscomfortFactor" type="time" minOccurs="0" maxOccurs="1"/>
      </sequence>
    </complexType>
  </element>
  <element name="gridKPI" type="inerkpi:gridKPIType" minOccurs="1" maxOccurs="1"/>
 </sequence>
</complexType>
```

element **tKeyPerformanceIndicator/energyKPI**

diagram




| | |
|---|---|
| properties | content complex |
| children | TotalEnergyConsumption EnergyConsumptionPerArea EnergyConsumptionPerOccupant TotalEnergyFromRES TotalEnergyFromGrid TotalEmissions RESRatioToTotalEnergyCosumption BaselineEnergyConsumption ScheduledEnergyConsumption |
| annotation | documentation<br>KPI aspoects: Energy, Flexibility, Business, Comfort, Grid |

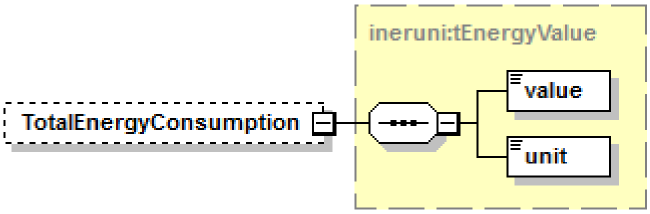
source

```
<element name="energyKPI" minOccurs="1" maxOccurs="1">
  <annotation>
    <documentation>
                  KPI aspoects: Energy, Flexibility, Business, Comfort, Grid
                </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="TotalEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="EnergyConsumptionPerArea" type="ineruni:tEnergyPerAreaValue" minOccurs="0" maxOccurs="1"/>
```

```
<element name="EnergyConsumptionPerOccupant" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
<element name="TotalEnergyFromRES" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
<element name="TotalEnergyFromGrid" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
<element name="TotalEmissions" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/>
<element name="RESRatioToTotalEnergyCosumption" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
<element name="BaselineEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
<element name="ScheduledEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/>
</sequence>
</complexType>
</element>
```

element **tKeyPerformanceIndicator/energyKPI/TotalEnergyConsumption**

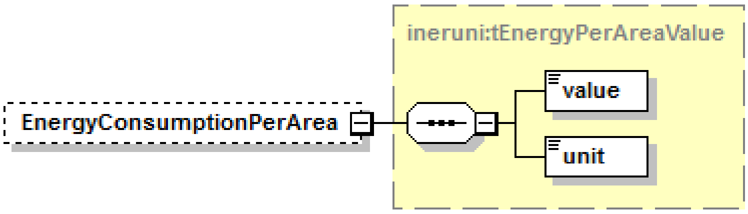
diagram




| | |
|---|---|
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="TotalEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/energyKPI/EnergyConsumptionPerArea**

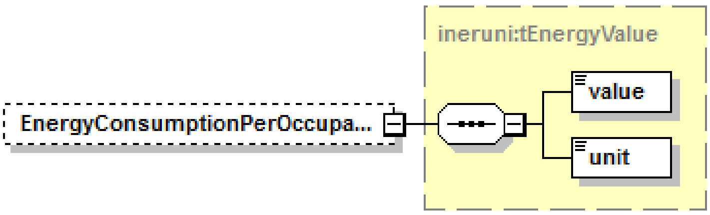
diagram




| | |
|---|---|
| type | **ineruni:tEnergyPerAreaValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="EnergyConsumptionPerArea" type="ineruni:tEnergyPerAreaValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/energyKPI/EnergyConsumptionPerOccupant**

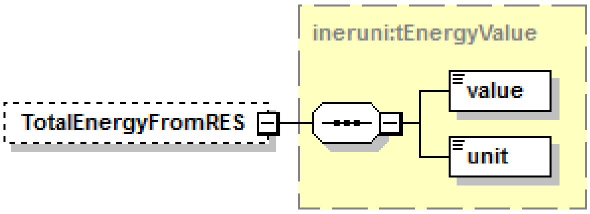
diagram



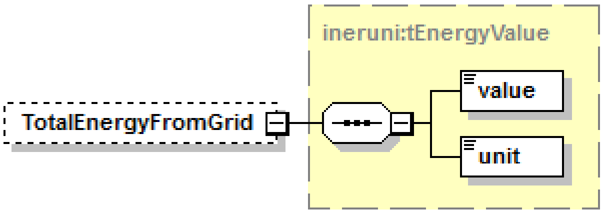

| | |
|---|---|
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |

| | |
|---|---|
| children | **value unit** |
| source | <element name="EnergyConsumptionPerOccupant" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

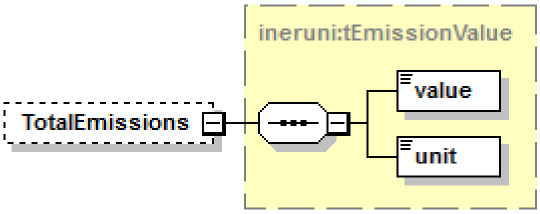
element **tKeyPerformanceIndicator/energyKPI/TotalEnergyFromRES**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="TotalEnergyFromRES" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

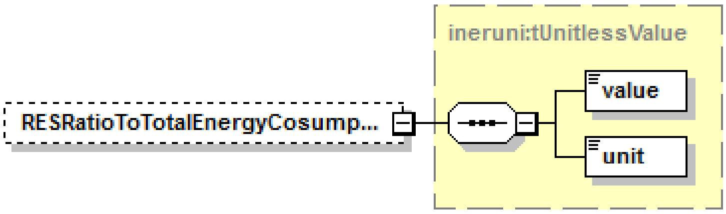
element **tKeyPerformanceIndicator/energyKPI/TotalEnergyFromGrid**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="TotalEnergyFromGrid" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/energyKPI/TotalEmissions**

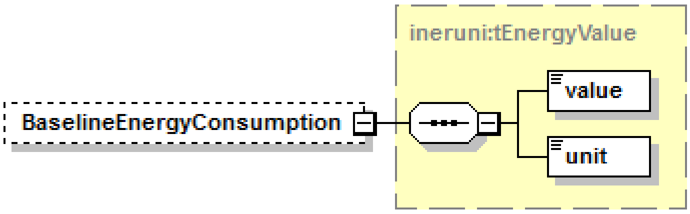
| | |
|---|---|
| diagram |  |
| type | **ineruni:tEmissionValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="TotalEmissions" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/energyKPI/RESRatioToTotalEnergyCosumption**

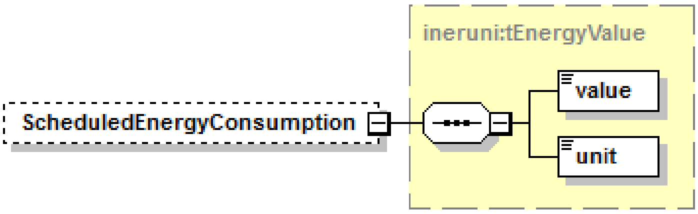
| | |
|---|---|
| diagram |  |

| | |
|---|---|
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RESRatioToTotalEnergyCosumption" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/energyKPI/BaselineEnergyConsumption**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="BaselineEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/energyKPI/ScheduledEnergyConsumption**

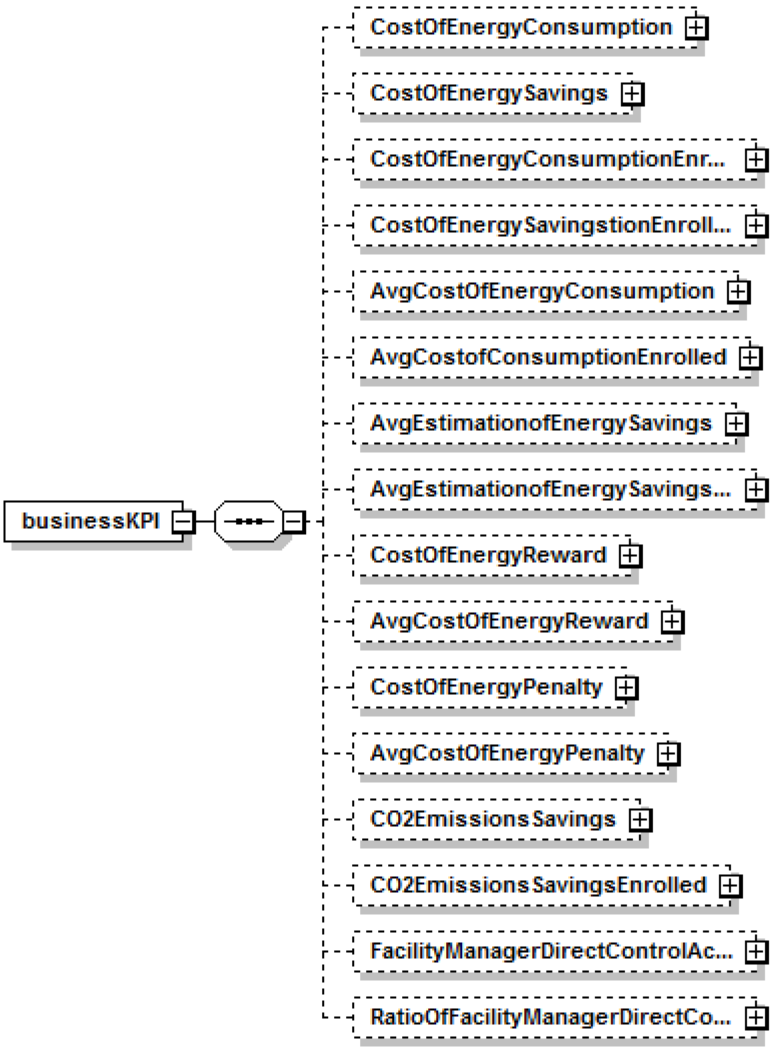
| | |
|---|---|
| diagram |  |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="ScheduledEnergyConsumption" type="ineruni:tEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/flexibilityKPI**

| | |
|---|---|
| diagram |   |
| type | **inerkpi:flexibilityKPIType** |
| properties | content complex |
| children | **DemandFlexibility** **DemandFlexibilityPerArea** **DemandFlexibilityPerOccupant** **DemandFlexibilityEnrolled** **DemandFlexibilityRatio** **RatioProvDemFlexToEnrolDemFlex** **RatioDemdFlexToContractFlex** **RatioEnrolDemFlexibToContractFlex** |

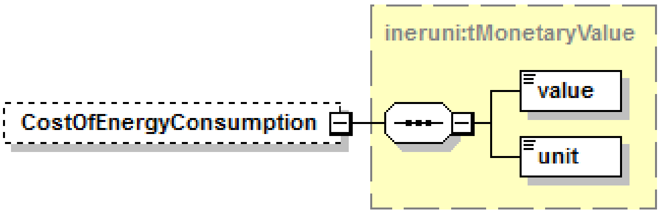
| | |
|---|---|
| | TotalNumberOfControlactionsEnrolled TotalNumberofControlactionsAccepted RatioEnrolledActionstoContractualActions RatioAcceptedActsToContractualActsAgreement RatioAcceptedControlActsToEnrolledControlActs RatioEnrolConttroActsToTotalNumOfEnrolControActs RatioControlActsToTotalNumberofControlActs TimePeriodOfControlActionsEnrolled TimePeriodOfControlActionsAccepted RatioTimePeriodAcceptedToTimeperiodEnrolled RatioTimePeriodEnrolActsToContractTimePeriod RatioAcceptedTimePeriodToContractTimePeriod RatioEnrolTimPOfControlActsToTotalTimPOfEnrolControlActs RatioTimPOfControlActsToTotalTimPOfControlActs AvgDemandFlexPerTimePeriodOfEvent AvgDemandFlexEnrolPerTimePeriodOfEvent MaxDemandFlexPerTimePeriodOfEvent MaxDemandFlexEnrolPerTimePeriodOfEvent DelayResponsivenessOfDemandFlex DelayOnDeliveryOfFlexAmount |
| source | <element name="flexibilityKPI" type="inerkpi:flexibilityKPIType" minOccurs="1" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI**

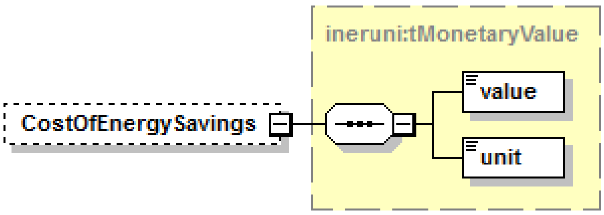
| | |
|---|---|
| diagram |  |
| properties | content complex |
| children | CostOfEnergyConsumption CostOfEnergySavings CostOfEnergyConsumptionEnrolled CostOfEnergySavingstionEnrolled AvgCostOfEnergyConsumption AvgCostofConsumptionEnrolled AvgEstimationofEnergySavings AvgEstimationofEnergySavingsEnrolled CostOfEnergyReward AvgCostOfEnergyReward CostOfEnergyPenalty AvgCostOfEnergyPenalty CO2EmissionsSavings CO2EmissionsSavingsEnrolled FacilityManagerDirectControlActions RatioOfFacilityManagerDirectControlActions |
| source | <element name="businessKPI" minOccurs="1" maxOccurs="1"><br><complexType><br><sequence><br><element name="CostOfEnergyConsumption" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

```xml
      <element name="CostOfEnergySavings" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
      <element name="CostOfEnergyConsumptionEnrolled" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
      <element name="CostOfEnergySavingstionEnrolled" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgCostOfEnergyConsumption" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgCostofConsumptionEnrolled" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgEstimationofEnergySavings" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgEstimationofEnergySavingsEnrolled" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="CostOfEnergyReward" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgCostOfEnergyReward" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="CostOfEnergyPenalty" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgCostOfEnergyPenalty" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/>
      <element name="CO2EmissionsSavings" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/>
      <element name="CO2EmissionsSavingsEnrolled" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/>
      <element name="FacilityManagerDirectControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="RatioOfFacilityManagerDirectControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
```

element **tKeyPerformanceIndicator/businessKPI/CostOfEnergyConsumption**

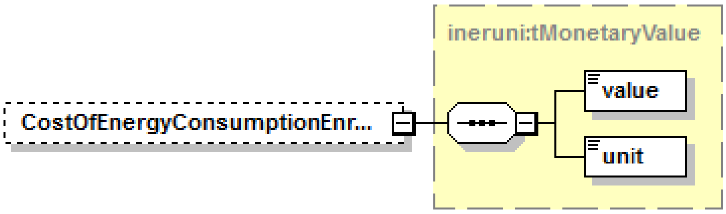
| | |
|---|---|
| diagram |  |
| type | **ineruni:tMonetaryValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CostOfEnergyConsumption" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI/CostOfEnergySavings**

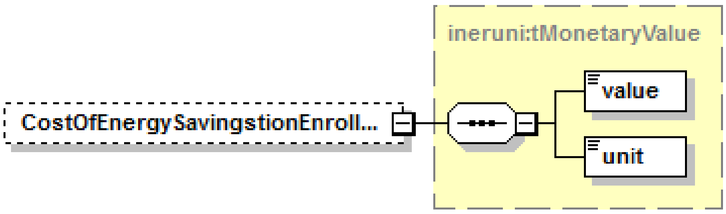
| | |
|---|---|
| diagram |  |
| type | **ineruni:tMonetaryValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |

| | |
|---|---|
| children | **value unit** |
| source | <element name="CostOfEnergySavings" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

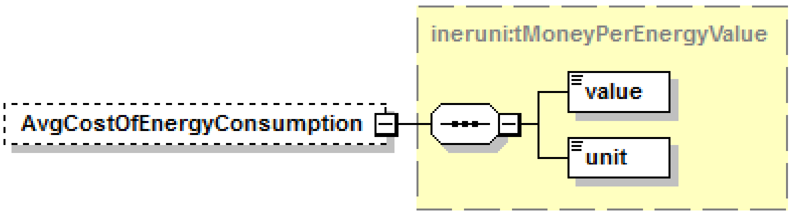
element **tKeyPerformanceIndicator/businessKPI/CostOfEnergyConsumptionEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMonetaryValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CostOfEnergyConsumptionEnrolled" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

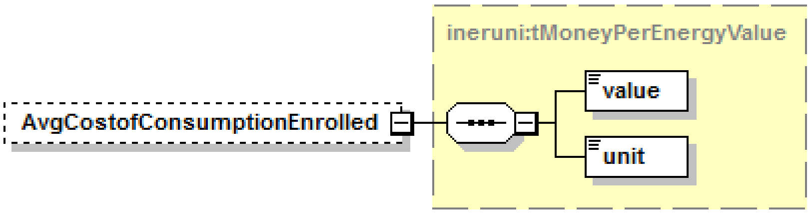
element **tKeyPerformanceIndicator/businessKPI/CostOfEnergySavingstionEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMonetaryValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CostOfEnergySavingstionEnrolled" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

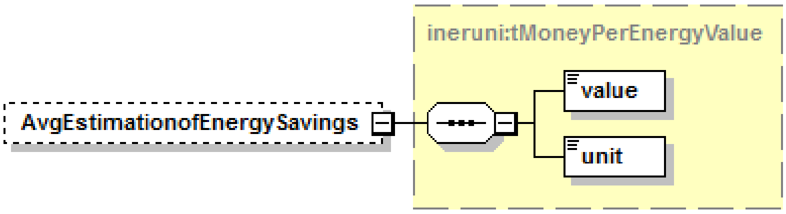
element **tKeyPerformanceIndicator/businessKPI/AvgCostOfEnergyConsumption**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgCostOfEnergyConsumption" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/> |

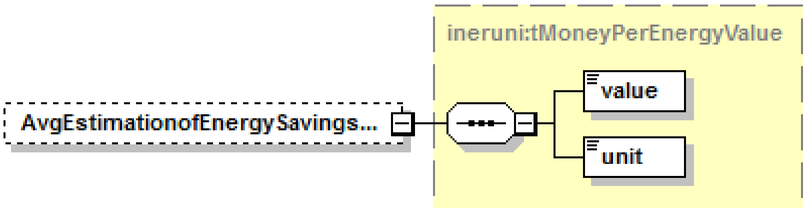
element **tKeyPerformanceIndicator/businessKPI/AvgCostofConsumptionEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgCostofConsumptionEnrolled" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/> |

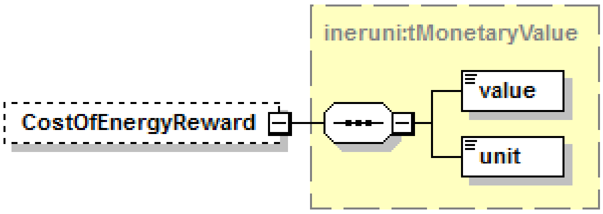
element **tKeyPerformanceIndicator/businessKPI/AvgEstimationofEnergySavings**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgEstimationofEnergySavings" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/> |

element
**tKeyPerformanceIndicator/businessKPI/AvgEstimationofEnergySavingsEnrolled**

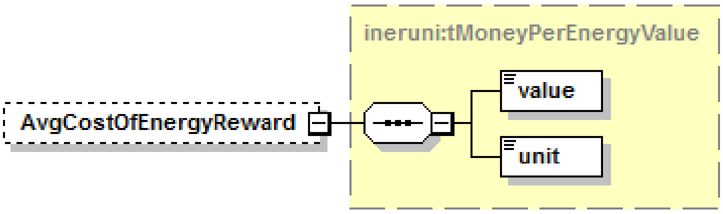
| | |
|---|---|
| diagram |  |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgEstimationofEnergySavingsEnrolled" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI/CostOfEnergyReward**

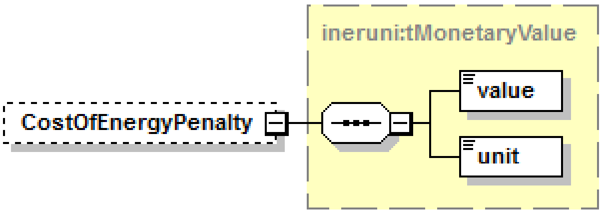
| | |
|---|---|
| diagram |  |
| type | **ineruni:tMonetaryValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |

| | |
|---|---|
| source | <element name="CostOfEnergyReward" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

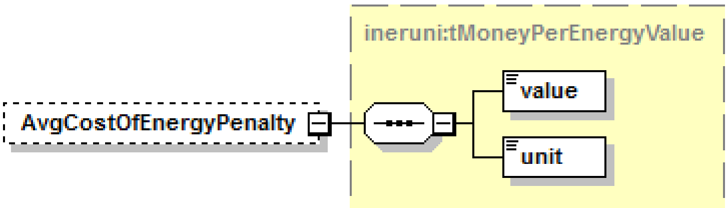
element **tKeyPerformanceIndicator/businessKPI/AvgCostOfEnergyReward**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgCostOfEnergyReward" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/> |

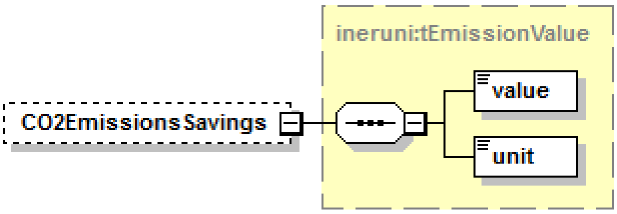
element **tKeyPerformanceIndicator/businessKPI/CostOfEnergyPenalty**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tMonetaryValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CostOfEnergyPenalty" type="ineruni:tMonetaryValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI/AvgCostOfEnergyPenalty**

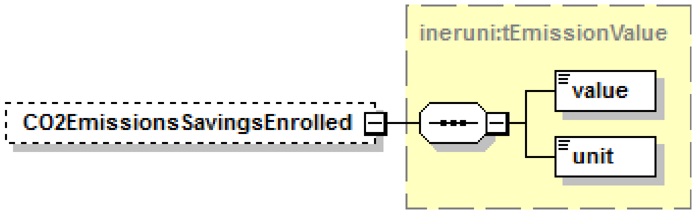
| | |
|---|---|
| diagram |  |
| type | **ineruni:tMoneyPerEnergyValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgCostOfEnergyPenalty" type="ineruni:tMoneyPerEnergyValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI/CO2EmissionsSavings**

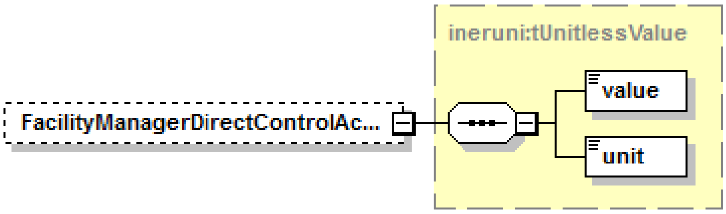
| | |
|---|---|
| diagram |  |
| type | **ineruni:tEmissionValue** |
| properties | minOcc 0 |

| | |
|---|---|
| | maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CO2EmissionsSavings" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/> |

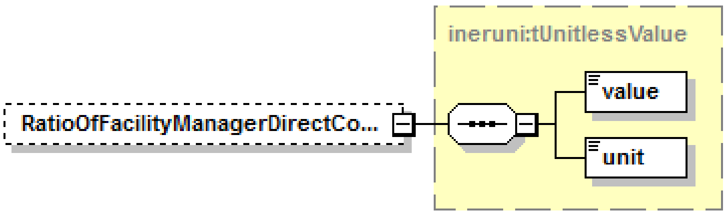
element **tKeyPerformanceIndicator/businessKPI/CO2EmissionsSavingsEnrolled**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tEmissionValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CO2EmissionsSavingsEnrolled" type="ineruni:tEmissionValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI/FacilityManagerDirectControlActions**

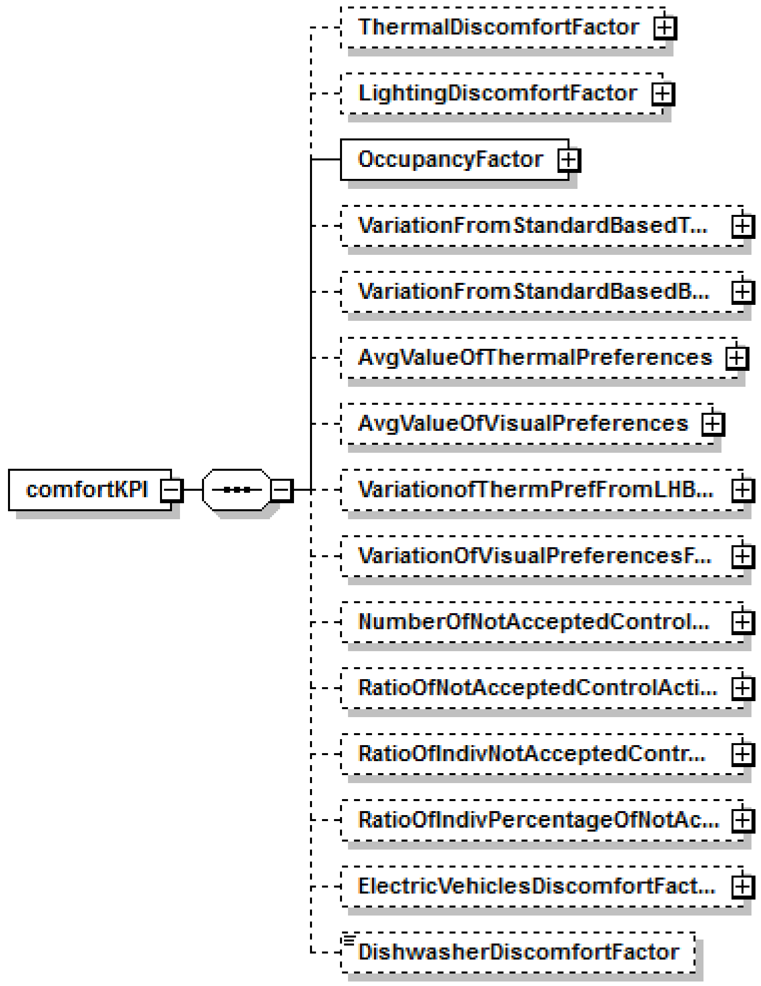
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="FacilityManagerDirectControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/businessKPI/RatioOfFacilityManagerDirectControlActions**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="RatioOfFacilityManagerDirectControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/comfortKPI**

diagram



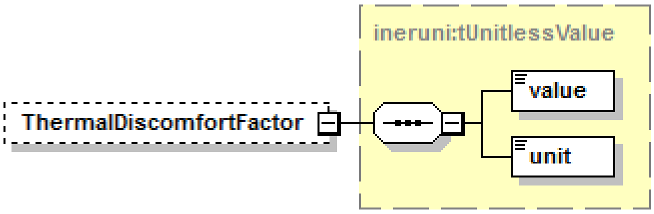

| | |
|---|---|
| properties | content complex |
| children | ThermalDiscomfortFactor LightingDiscomfortFactor OccupancyFactor VariationFromStandardBasedThermalBaseline VariationFromStandardBasedBaseline AvgValueOfThermalPreferences AvgValueOfVisualPreferences VariationofThermPrefFromLHBaseline VariationOfVisualPreferencesFromL.H.Baseline NumberOfNotAcceptedControlActions RatioOfNotAcceptedControlActionsToTotalNumber RatioOfIndivNotAcceptedControlActionsToTotalNumber RatioOfIndivPercentageOfNotAcceptedControlActs ElectricVehiclesDiscomfortFactor DishwasherDiscomfortFactor |

source

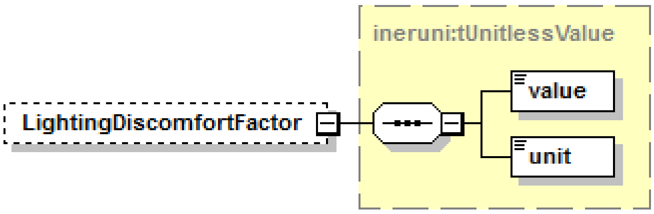
```
<element name="comfortKPI" minOccurs="1" maxOccurs="1">
  <complexType>
    <sequence>
      <element name="ThermalDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="LightingDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="OccupancyFactor" type="ineruni:tUnitlessValue" minOccurs="1" maxOccurs="1"/>
      <element name="VariationFromStandardBasedThermalBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="VariationFromStandardBasedBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgValueOfThermalPreferences" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="AvgValueOfVisualPreferences" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
      <element name="VariationofThermPrefFromLHBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/>
      <element name="VariationOfVisualPreferencesFromL.H.Baseline"
```

| | |
|---|---|
| | type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/><br><element name="NumberOfNotAcceptedControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="RatioOfNotAcceptedControlActionsToTotalNumber" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="RatioOfIndivNotAcceptedControlActionsToTotalNumber" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="RatioOfIndivPercentageOfNotAcceptedControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="ElectricVehiclesDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/><br><element name="DishwasherDiscomfortFactor" type="time" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType><br></element> |

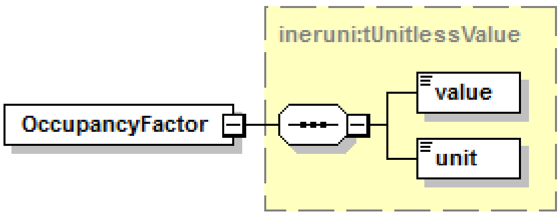
element **tKeyPerformanceIndicator/comfortKPI/ThermalDiscomfortFactor**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="ThermalDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/comfortKPI/LightingDiscomfortFactor**

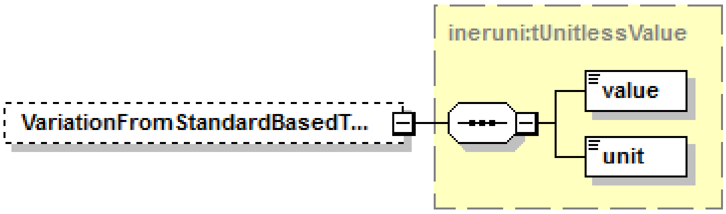
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="LightingDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/comfortKPI/OccupancyFactor**

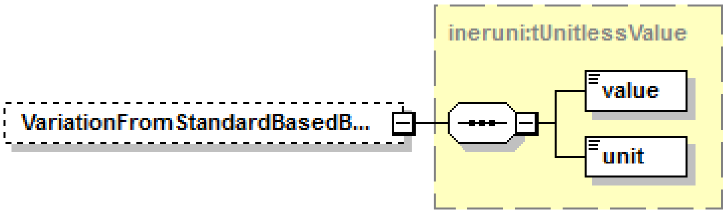
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="OccupancyFactor" type="ineruni:tUnitlessValue" |

| | |
|---|---|
| | minOccurs="1" maxOccurs="1"/> |

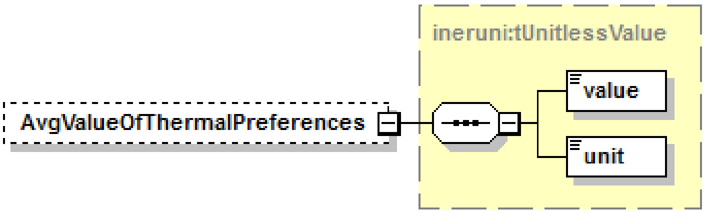
element **tKeyPerformanceIndicator/comfortKPI/VariationFromStandardBasedThermalBaseline**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="VariationFromStandardBasedThermalBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

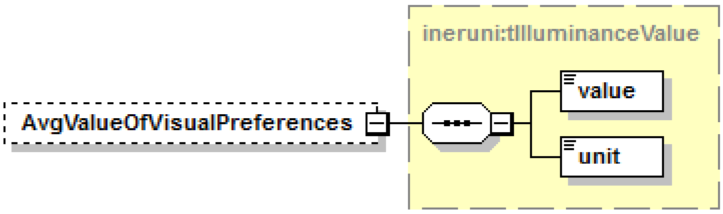
element **tKeyPerformanceIndicator/comfortKPI/VariationFromStandardBasedBaseline**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="VariationFromStandardBasedBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/comfortKPI/AvgValueOfThermalPreferences**

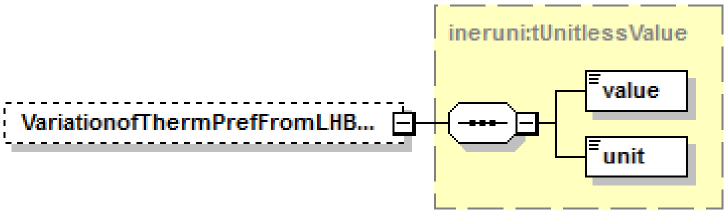
| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgValueOfThermalPreferences" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/comfortKPI/AvgValueOfVisualPreferences**

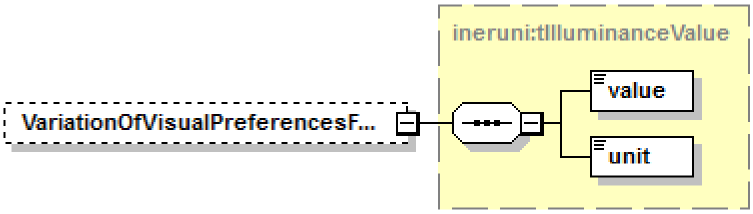
| | |
|---|---|
| diagram |  |

| | |
|---|---|
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="AvgValueOfVisualPreferences" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

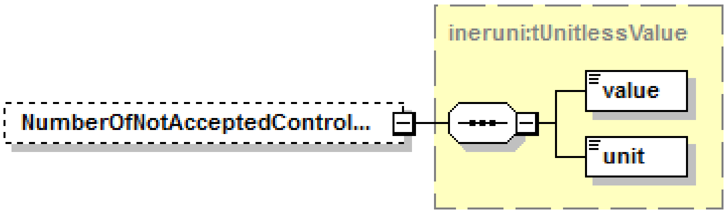
element **tKeyPerformanceIndicator/comfortKPI/VariationofThermPrefFromLHBaseline**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="VariationofThermPrefFromLHBaseline" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

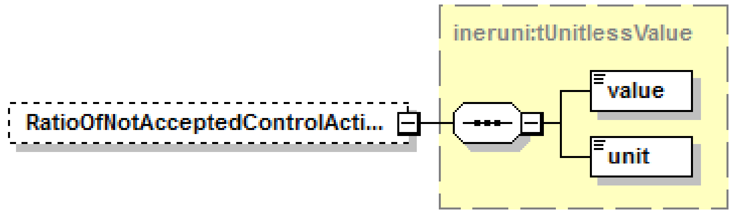
element **tKeyPerformanceIndicator/comfortKPI/VariationOfVisualPreferencesFromL.H.Baseline**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="VariationOfVisualPreferencesFromL.H.Baseline" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

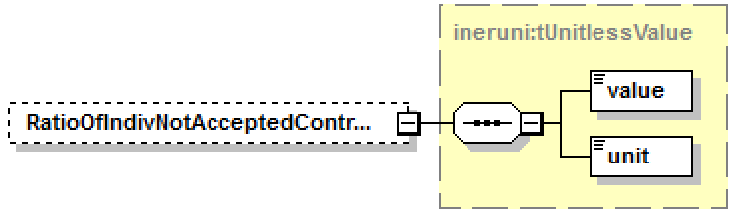
element **tKeyPerformanceIndicator/comfortKPI/NumberOfNotAcceptedControlActions**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="NumberOfNotAcceptedControlActions" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

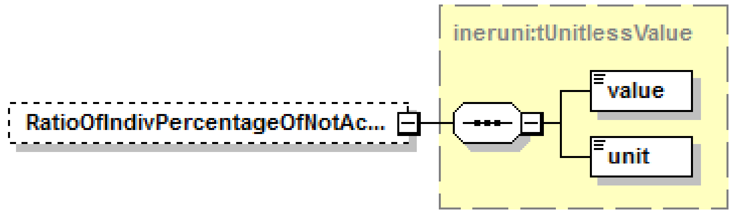
element **tKeyPerformanceIndicator/comfortKPI/RatioOfNotAcceptedControlActionsToTotalNumber**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td>ineruni:tUnitlessValue</td></tr>
<tr><td>properties</td><td>minOcc 0<br>maxOcc 1<br>content complex</td></tr>
<tr><td>children</td><td>value unit</td></tr>
<tr><td>source</td><td><element name="RatioOfNotAcceptedControlActionsToTotalNumber" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/></td></tr>
</table>

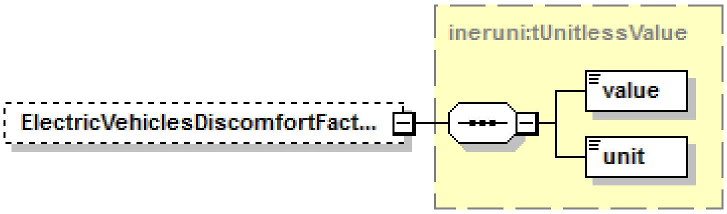
element **tKeyPerformanceIndicator/comfortKPI/RatioOfIndivNotAcceptedControlActionsToTotalNumber**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td>ineruni:tUnitlessValue</td></tr>
<tr><td>properties</td><td>minOcc 0<br>maxOcc 1<br>content complex</td></tr>
<tr><td>children</td><td>value unit</td></tr>
<tr><td>source</td><td><element name="RatioOfIndivNotAcceptedControlActionsToTotalNumber" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/></td></tr>
</table>

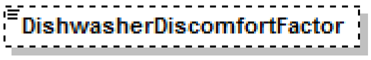
element **tKeyPerformanceIndicator/comfortKPI/RatioOfIndivPercentageOfNotAcceptedControlActs**

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td>ineruni:tUnitlessValue</td></tr>
<tr><td>properties</td><td>minOcc 0<br>maxOcc 1<br>content complex</td></tr>
<tr><td>children</td><td>value unit</td></tr>
<tr><td>source</td><td><element name="RatioOfIndivPercentageOfNotAcceptedControlActs" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/></td></tr>
</table>

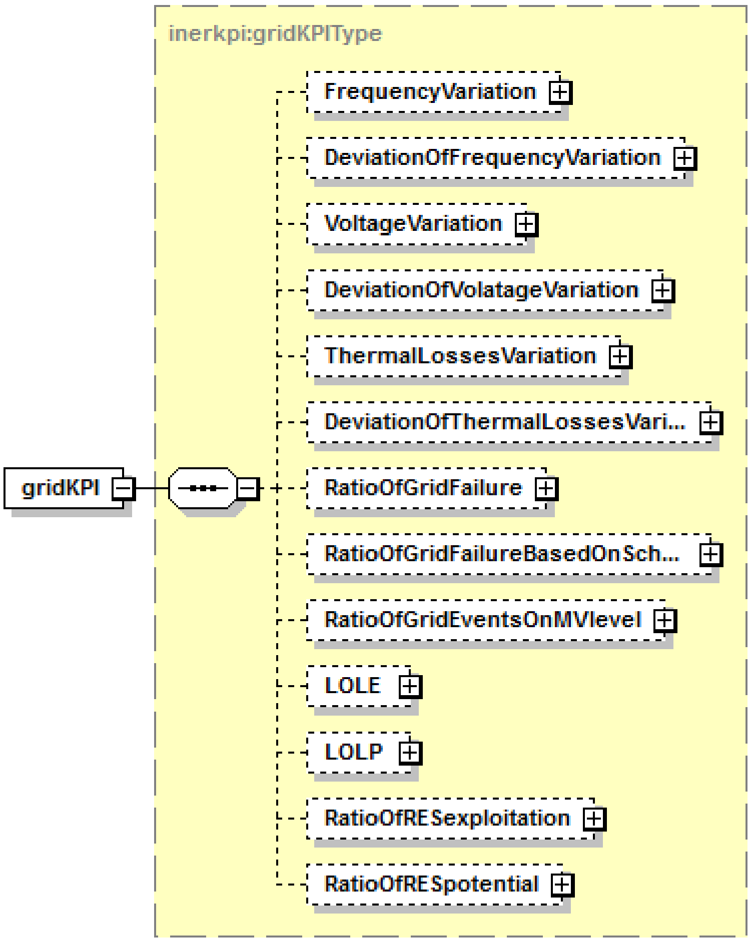
element **tKeyPerformanceIndicator/comfortKPI/ElectricVehiclesDiscomfortFactor**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tUnitlessValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="ElectricVehiclesDiscomfortFactor" type="ineruni:tUnitlessValue" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/comfortKPI/DishwasherDiscomfortFactor**

| | |
|---|---|
| diagram |  |
| type | **time** |
| properties | minOcc 0<br>maxOcc 1<br>content simple |
| source | <element name="DishwasherDiscomfortFactor" type="time" minOccurs="0" maxOccurs="1"/> |

element **tKeyPerformanceIndicator/gridKPI**

| | |
|---|---|
| diagram |  |
| type | **inerkpi:gridKPIType** |
| properties | content complex |
| children | **FrequencyVariation** **DeviationOfFrequencyVariation** **VoltageVariation** |

| | |
|---|---|
| | DeviationOfVolatageVariation ThermalLossesVariation DeviationOfThermalLossesVariation RatioOfGridFailure RatioOfGridFailureBasedOnScheduledAnalysis RatioOfGridEventsOnMVlevel LOLE LOLP RatioOfRESexploitation RatioOfRESpotential |
| source | `<element name="gridKPI" type="inerkpi:gridKPIType" minOccurs="1" maxOccurs="1"/>` |

simpleType **ActivityEnumerator**

| | | | |
|---|---|---|---|
| namespace | http://www.inertia.eu/2013/schema/kpi/ | | |
| type | restriction of **string** | | |
| properties | base string | | |
| used by | element **tEventTrace/eventRecord/activityType** | | |
| facets | Kind | Value | Annotation |
| | enumeration | Occupant enters Building Element (BE corresponds to BE in Event Record) | |
| | enumeration | Occupant leaves Building Element (BE corresponds to BE in Event Record) | |
| | enumeration | Occupant starts using Equipment Artifact (EA corresponds to EA in Event Record) | |
| | enumeration | Occupants stops using Equipment Artifact (EA corresponds to EA in Event Record) | |
| | enumeration | Occupant starts executing some Business Activity | |
| | enumeration | Occupant stops executing some Business Activity | |
| | enumeration | Occupant turns on Lighting within Building Element | |
| | enumeration | Occupant turns off Lighting within Building Element | |

source

```
<simpleType name="ActivityEnumerator">
  <restriction base="string">
    <enumeration value="Occupant enters Building Element (BE corresponds to BE in Event Record)"/>
    <enumeration value="Occupant leaves Building Element (BE corresponds to BE in Event Record)"/>
    <enumeration value="Occupant starts using Equipment Artifact (EA corresponds to EA in Event Record)"/>
    <enumeration value="Occupants stops using Equipment Artifact (EA corresponds to EA in Event Record)"/>
    <enumeration value="Occupant starts executing some Business Activity"/>
    <enumeration value="Occupant stops executing some Business Activity"/>
    <enumeration value="Occupant turns on Lighting within Building Element"/>
    <enumeration value="Occupant turns off Lighting within Building Element"/>
  </restriction>
</simpleType>
```

XML Schema documentation generated by **XMLSpy** Schema Editor **http://www.altova.com/xmlspy**

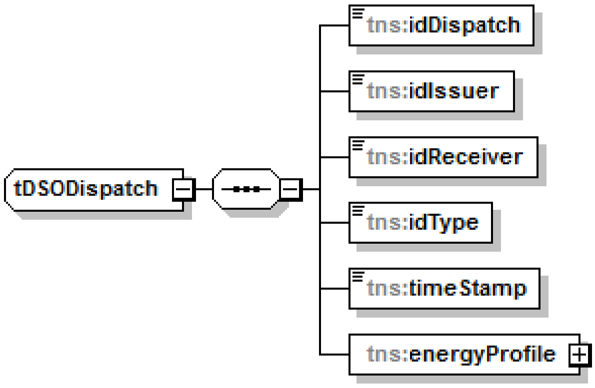
# Schema inertiaGridCoordinationAndGridActivation.xsd

| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaGridCoordinationAndGridActivation.xsd** |
| attributeFormDefault: | |
| elementFormDefault: | **qualified** |
| targetNamespace: | **http://www.example.org/inertiaGridCoordinationAndGridActivation** |

Complex types
**tDSODispatch**
**tMV2LCHMapping**

### complexType **tDSODispatch**

diagram




| | |
|---|---|
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| children | **tns:idDispatch tns:idIssuer tns:idReceiver tns:idType tns:timeStamp tns:energyProfile** |

source

```
<complexType name="tDSODispatch">
  <sequence>
    <element name="idDispatch" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="idIssuer" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="idReceiver" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="idType" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/>
    <element name="energyProfile" minOccurs="1" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="idSupply" type="string" minOccurs="1" maxOccurs="1"/>
          <element name="timeStampBegin" type="dateTime" minOccurs="1" maxOccurs="1"/>
          <element name="timeStampEnd" type="dateTime" minOccurs="1" maxOccurs="1"/>
          <element name="timeStampStep" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/>
          <element name="energyValues" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

### element **tDSODispatch/idDispatch**

diagram


| | |
|---|---|
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |

source

```
<element name="idDispatch" type="string" minOccurs="1" maxOccurs="1"/>
```

### element **tDSODispatch/idIssuer**

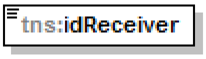
diagram


| | |
|---|---|
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |

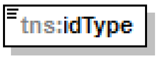
source

```
<element name="idIssuer" type="string" minOccurs="1" maxOccurs="1"/>
```

element **tDSODispatch/idReceiver**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |
| source | <element name="idReceiver" type="string" minOccurs="1" maxOccurs="1"/> |

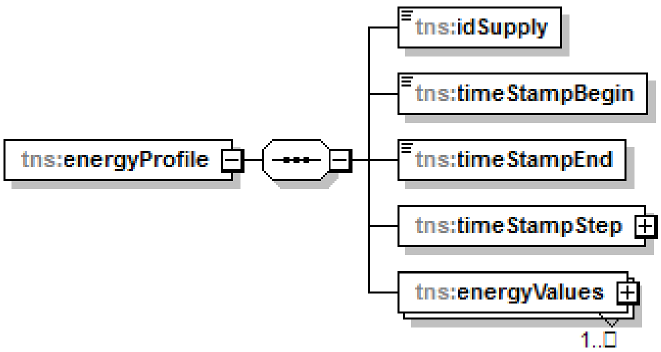
element **tDSODispatch/idType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |
| source | <element name="idType" type="string" minOccurs="1" maxOccurs="1"/> |

element **tDSODispatch/timeStamp**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStamp" type="dateTime" minOccurs="1" maxOccurs="1"/> |

element **tDSODispatch/energyProfile**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| properties | content complex |
| children | **tns:idSupply tns:timeStampBegin tns:timeStampEnd tns:timeStampStep tns:energyValues** |
| source | <element name="energyProfile" minOccurs="1" maxOccurs="1"><br><complexType><br><sequence><br><element name="idSupply" type="string" minOccurs="1" maxOccurs="1"/><br><element name="timeStampBegin" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="timeStampEnd" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="timeStampStep" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/><br><element name="energyValues" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType><br></element> |

element **tDSODispatch/energyProfile/idSupply**

| | |
|---|---|
| diagram |  |

| | |
|---|---|
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |
| source | <element name="idSupply" type="string" minOccurs="1" maxOccurs="1"/> |

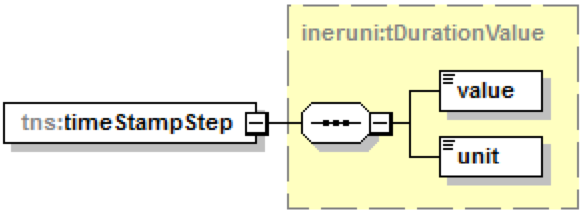
element **tDSODispatch/energyProfile/timeStampBegin**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStampBegin" type="dateTime" minOccurs="1" maxOccurs="1"/> |

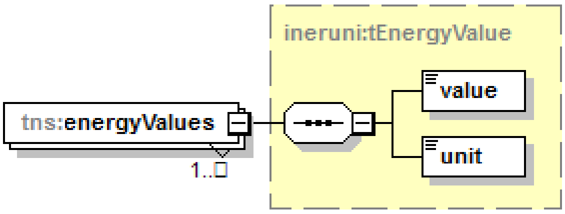
element **tDSODispatch/energyProfile/timeStampEnd**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStampEnd" type="dateTime" minOccurs="1" maxOccurs="1"/> |

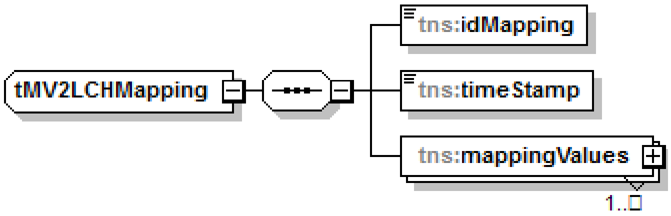
element **tDSODispatch/energyProfile/timeStampStep**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **ineruni:tDurationValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="timeStampStep" type="ineruni:tDurationValue" minOccurs="1" maxOccurs="1"/> |

element **tDSODispatch/energyProfile/energyValues**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **ineruni:tEnergyValue** |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **value unit** |
| source | <element name="energyValues" type="ineruni:tEnergyValue" minOccurs="1" maxOccurs="unbounded"/> |

complexType **tMV2LCHMapping**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| children | **tns:idMapping tns:timeStamp tns:mappingValues** |

source

```
<complexType name="tMV2LCHMapping">
  <sequence>
    <element name="idMapping" type="string" minOccurs="1"
maxOccurs="1"/>
    <element name="timeStamp" type="dateTime" minOccurs="1"
maxOccurs="1"/>
    <element name="mappingValues" minOccurs="1"
maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="idSupply" type="string" minOccurs="1"
maxOccurs="1"/>
          <element name="idLocalHubs" type="string" minOccurs="1"
maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

element **tMV2LCHMapping/idMapping**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |

source

```
<element name="idMapping" type="string" minOccurs="1" maxOccurs="1"/>
```

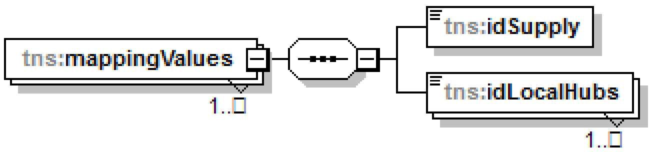
element **tMV2LCHMapping/timeStamp**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **dateTime** |
| properties | content simple |

source

```
<element name="timeStamp" type="dateTime" minOccurs="1"
maxOccurs="1"/>
```

element **tMV2LCHMapping/mappingValues**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| properties | minOcc 1<br>maxOcc unbounded<br>content complex |
| children | **tns:idSupply tns:idLocalHubs** |

source

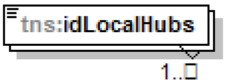
```
<element name="mappingValues" minOccurs="1" maxOccurs="unbounded">
  <complexType>
    <sequence>
```

| | |
|---|---|
| | <element name="idSupply" type="string" minOccurs="1" maxOccurs="1"/><br><element name="idLocalHubs" type="string" minOccurs="1" maxOccurs="unbounded"/><br></sequence><br></complexType><br></element> |

element **tMV2LCHMapping/mappingValues/idSupply**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | content simple |
| source | <element name="idSupply" type="string" minOccurs="1" maxOccurs="1"/> |

element **tMV2LCHMapping/mappingValues/idLocalHubs**

| | |
|---|---|
| diagram |  |
| namespace | http://www.example.org/inertiaGridCoordinationAndGridActivation |
| type | **string** |
| properties | minOcc 1<br>maxOcc unbounded<br>content simple |
| source | <element name="idLocalHubs" type="string" minOccurs="1" maxOccurs="unbounded"/> |

XML Schema documentation generated by **XMLSpy** Schema Editor **http://www.altova.com/xmlspy**

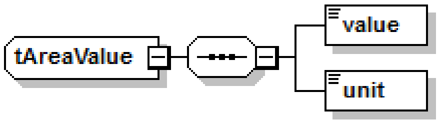
# Schema inertiaUnits.xsd

| | |
|---|---|
| schema location: | **C:\Users\Marek Skokan\Dropbox\Projekty_TUKE\Inertia\Deliverables\xsd\xsd_23_9\inertiaUnits.xsd** |
| attributeFormDefault: | **unqualified** |
| elementFormDefault: | **unqualified** |
| targetNamespace: | **http://www.inertia.eu/2013/schema/units/** |

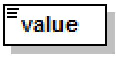
| Complex types | Simple types |
|---|---|
| **tAreaValue** | **areaUnitEnum** |
| **tConcentrationOfGasValue** | **buildingTypeEnum** |
| **tCurrentValue** | **clothingUnitEnum** |
| **tDayType** | **concentrationOfGasUnitEnum** |
| **tDurationValue** | **CurrentUnitEnum** |
| **tEfficiencyValue** | **efficiencyTypeEnum** |
| **tEmissionValue** | **emissionUnitEnum** |
| **tEnergyPerAreaValue** | **energyPerAreaUnitEnum** |
| **tEnergyValue** | **energyPerTimeUnitEnum** |
| **tEnergyValuePerTime** | **energyUnitEnum** |
| **tEnvironmentComfortValues** | **eventTypeEnum** |
| **tEnvironmentValues** | **illuminanceUnitEnum** |
| **tFrequencyValue** | **intEquipTypeEnum** |
| **tIlluminanceValue** | **lengthUnitEnum** |
| **tLengthValue** | **monetaryUnitEnum** |
| **tMonetaryValue** | **moneyPerEnergyUnitEnum** |
| **tMoneyPerEnergyValue** | **noiseUnitEnum** |
| **tNoiseValue** | **peopleHeatGainUnitEnum** |
| **tOccupantPositionValue** | **peopleNumberUnitEnum** |
| **tOtherValue** | **PMVUnitEnum** |
| **tPeopleClothingValue** | **resourceTypeEnum** |
| **tPeopleHeatGainValue** | **seasonUnitEnum** |
| **tPeopleNumberValue** | **spaceTypeEnum** |
| **tPossitionValue** | **stateTypeEnum** |
| **tRelativeHumidityValue** | **stationIdTypeEnum** |
| **tSeasonValue** | **temperatureUnitEnum** |

tStationIdValue
tStatisticValues
tTemperatureValue
tTimeDatePeriod
tTimePeriod
tTimeSerieNumberValue
tTimeSeriePercentageValue
tTimeSeriesOfNumbersValue
tTimeSeriesOfPercentagesValue
tTimeSeriesOfStatisticsValue
tTimeSerieStatisticValue
tTrajectory
tUnitlessValue
tVelocityValue
tVoltageValue
tVolumeValue
timeUnitEnum
typesOfEventUserProfilingUnitEnum
unitlessUnitEnum
velocityUnitEnum
VoltageUnitEnum
volumeUnitEnum
weekDayUnitEnum

complexType **tAreaValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | value unit |
| source | <complexType name="tAreaValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:areaUnitEnum"/><br></sequence><br></complexType> |

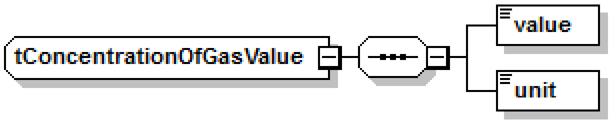
element **tAreaValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

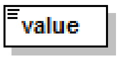
element **tAreaValue/unit**

| | | | |
|---|---|---|---|
| diagram |  | | |
| type | ineruni:areaUnitEnum | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | SquareKilometers | |
| | enumeration | SquareMeters | |
| | enumeration | SquareCentimeters | |
| | enumeration | SquareMillimeters | |
| | enumeration | SquareMiles | |
| | enumeration | SquareYards | |
| | enumeration | SquareFeet | |
| | enumeration | SquareInches | |
| source | <element name="unit" type="ineruni:areaUnitEnum"/> | | |

complexType **tConcentrationOfGasValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| used by | elements **tEnvironmentValues/CO2** **tEnvironmentComfortValues/maxCO2** **tEnvironmentComfortValues/minCO2** |
| source | <complexType name="tConcentrationOfGasValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:concentrationOfGasUnitEnum"/><br></sequence><br></complexType> |

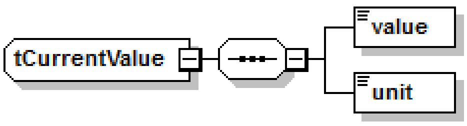
element **tConcentrationOfGasValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

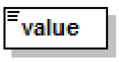
element **tConcentrationOfGasValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:concentrationOfGasUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration PartsPerMillion |
| source | <element name="unit" type="ineruni:concentrationOfGasUnitEnum"/> |

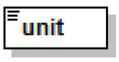
complexType **tCurrentValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tCurrentValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:CurrentUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tCurrentValue/value**

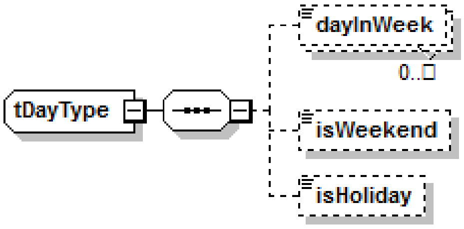
| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

element **tCurrentValue/unit**

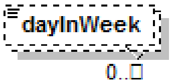
| | |
|---|---|
| diagram |  |
| type | **ineruni:CurrentUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation |

| | | |
|---|---|---|
| | enumeration | Ampere |
| | enumeration | milliAmpere |
| source | <element name="unit" type="ineruni:CurrentUnitEnum" minOccurs="1" maxOccurs="1"/> | |

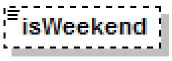
complexType **tDayType**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **dayInWeek isWeekend isHoliday** |
| source | <complexType name="tDayType"><br><sequence><br><element name="dayInWeek" type="ineruni:weekDayUnitEnum" minOccurs="0" maxOccurs="unbounded"/><br><element name="isWeekend" type="boolean" minOccurs="0" maxOccurs="1"/><br><element name="isHoliday" type="boolean" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

element **tDayType/dayInWeek**

| | | | |
|---|---|---|---|
| diagram |  | | |
| type | **ineruni:weekDayUnitEnum** | | |
| properties | minOcc | 0 | |
| | maxOcc | unbounded | |
| | content | simple | |
| facets | Kind | Value | Annotation |
| | enumeration | Monday | |
| | enumeration | Tuesday | |
| | enumeration | Wednesday | |
| | enumeration | Thursday | |
| | enumeration | Friday | |
| | enumeration | Saturday | |
| | enumeration | Sunday | |
| source | <element name="dayInWeek" type="ineruni:weekDayUnitEnum" minOccurs="0" maxOccurs="unbounded"/> | | |

element **tDayType/isWeekend**

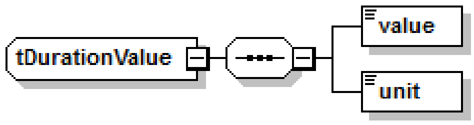
| | | |
|---|---|---|
| diagram |  | |
| type | **boolean** | |
| properties | minOcc | 0 |
| | maxOcc | 1 |
| | content | simple |
| source | <element name="isWeekend" type="boolean" minOccurs="0" maxOccurs="1"/> | |

element **tDayType/isHoliday**

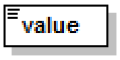
| | | |
|---|---|---|
| diagram |  | |
| type | **boolean** | |
| properties | minOcc | 0 |

| | |
|---|---|
| | maxOcc 1<br>content simple |
| source | <element name="isHoliday" type="boolean" minOccurs="0" maxOccurs="1"/> |

complexType **tDurationValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | <complexType name="tDurationValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:timeUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

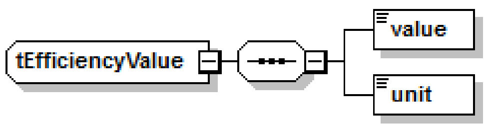
element **tDurationValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

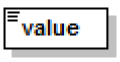
element **tDurationValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:timeUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration MilliSeconds<br>enumeration Seconds<br>enumeration Minutes<br>enumeration Hours<br>enumeration Days |
| source | <element name="unit" type="ineruni:timeUnitEnum" minOccurs="1" maxOccurs="1"/> |

complexType **tEfficiencyValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | <complexType name="tEfficiencyValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:efficiencyTypeEnum"/><br></sequence><br></complexType> |

element **tEfficiencyValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |

| | |
|---|---|
| source | <element name="value" type="float"/> |

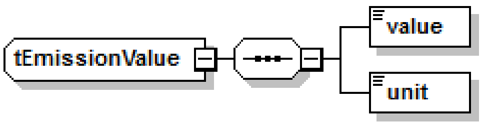
element **tEfficiencyValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:efficiencyTypeEnum** |
| properties | content simple |

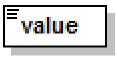
| facets | Kind | Value | Annotation |
|---|---|---|---|
| | enumeration | COP | documentation<br>Coefficient of Performance |
| | enumeration | EER | documentation<br>Energy Efficiency Ratio |
| | enumeration | SEER | documentation<br>Seasonal Energy Efficiency Ratio |
| | enumeration | AFUE | documentation<br>Annual Fuel Utilization Efficiency |
| | enumeration | HSPF | documentation<br>Heating Seasonal Performance Factor |
| | enumeration | effectiveness | |
| | enumeration | EnergyFactor | |
| | enumeration | ThermalEff | documentation<br>Thermal Efficiency |
| | enumeration | MotorEff | documentation<br>Motor Efficiency |
| | enumeration | FanEff | documentation<br>Fan Efficiency |
| | enumeration | MechanicalEff | documentation<br>Mechanical Efficiency |
| | enumeration | BoilerEff | documentation<br>Boiler Efficiency |
| | enumeration | kWPerTon | |
| | enumeration | kWPerkW | |
| | enumeration | BTUPerHourPerF | |
| | enumeration | kWPerC | |

| | |
|---|---|
| source | <element name="unit" type="ineruni:efficiencyTypeEnum"/> |

complexType **tEmissionValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | <complexType name="tEmissionValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:emissionUnitEnum"/><br></sequence><br></complexType> |

element **tEmissionValue/value**

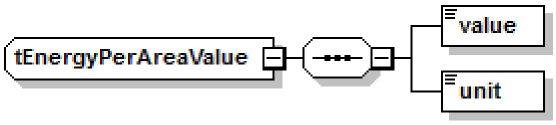
| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

element **tEmissionValue/unit**

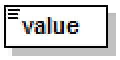
| | |
|---|---|
| diagram |  |
| type | **ineruni:emissionUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation |

| | | |
|---|---|---|
| | enumeration | KilogramsPerkWh |
| | enumeration | PoundsPerHour |
| | enumeration | KilogramsPerHour |
| | enumeration | GramsPerHour |
| source | <element name="unit" type="ineruni:emissionUnitEnum"/> | |

complexType **tEnergyPerAreaValue**

| | |
|---|---|
| diagram |   |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tEnergyPerAreaValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:energyPerAreaUnitEnum"/><br></sequence><br></complexType> |

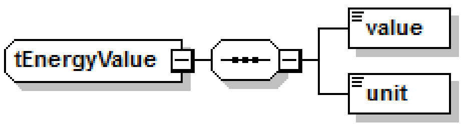
element **tEnergyPerAreaValue/value**

| | |
|---|---|
| diagram |   |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

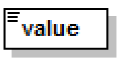
element **tEnergyPerAreaValue/unit**

| | | | |
|---|---|---|---|
| diagram |   | | |
| type | **ineruni:energyPerAreaUnitEnum** | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | KilowattHoursPerSquareMeter | |
| | enumeration | JoulesPerSquareMeter | |
| source | <element name="unit" type="ineruni:energyPerAreaUnitEnum"/> | | |

complexType **tEnergyValue**

| | |
|---|---|
| diagram |   |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tEnergyValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:energyUnitEnum"/><br></sequence><br></complexType> |

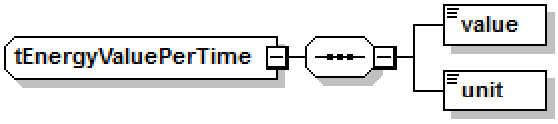
element **tEnergyValue/value**

| | |
|---|---|
| diagram |   |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

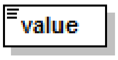
element **tEnergyValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:energyUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration KilowattHours<br>enumeration Joules<br>enumeration BTU<br>enumeration Therms |
| source | <element name="unit" type="ineruni:energyUnitEnum"/> |

complexType **tEnergyValuePerTime**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tEnergyValuePerTime"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:energyPerTimeUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tEnergyValuePerTime/value**

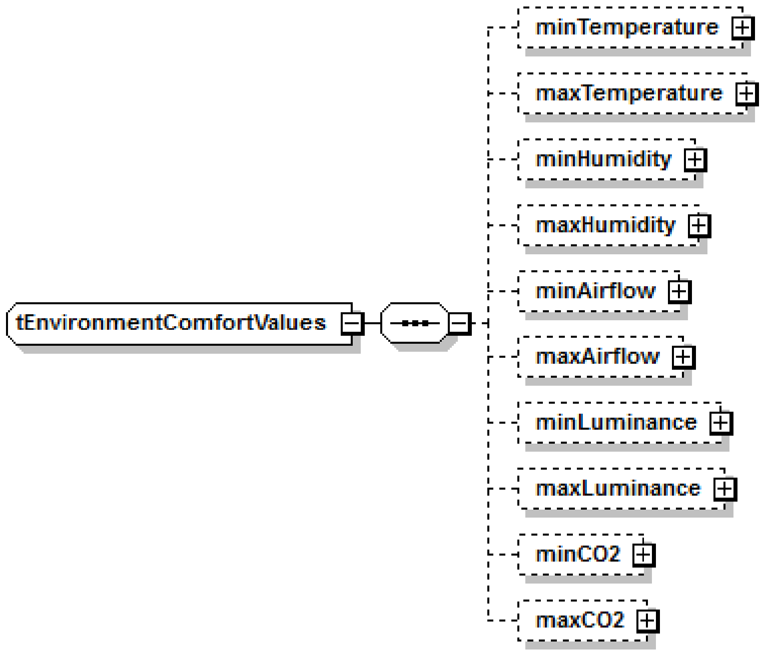
| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

element **tEnergyValuePerTime/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:energyPerTimeUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration KilowattHoursPerMinute<br>enumeration KilowattHoursPersecond<br>enumeration JoulesPerSecond<br>enumeration Watt |
| source | <element name="unit" type="ineruni:energyPerTimeUnitEnum" minOccurs="1" maxOccurs="1"/> |

## complexType **tEnvironmentComfortValues**

diagram



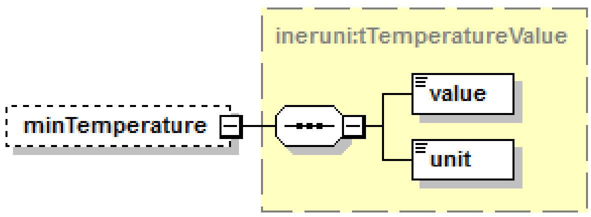

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **minTemperature** **maxTemperature** **minHumidity** **maxHumidity** **minAirflow** **maxAirflow** **minLuminance** **maxLuminance** **minCO2** **maxCO2** |

source

```
<complexType name="tEnvironmentComfortValues">
  <sequence>
    <element name="minTemperature" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/>
    <element name="maxTemperature" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/>
    <element name="minHumidity" type="ineruni:tRelativeHumidityValue" minOccurs="0" maxOccurs="1"/>
    <element name="maxHumidity" type="ineruni:tRelativeHumidityValue" minOccurs="0" maxOccurs="1"/>
    <element name="minAirflow" type="ineruni:tVelocityValue" minOccurs="0" maxOccurs="1"/>
    <element name="maxAirflow" type="ineruni:tVelocityValue" minOccurs="0" maxOccurs="1"/>
    <element name="minLuminance" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
    <element name="maxLuminance" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/>
    <element name="minCO2" type="ineruni:tConcentrationOfGasValue" minOccurs="0" maxOccurs="1"/>
    <element name="maxCO2" type="ineruni:tConcentrationOfGasValue" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

## element **tEnvironmentComfortValues/minTemperature**

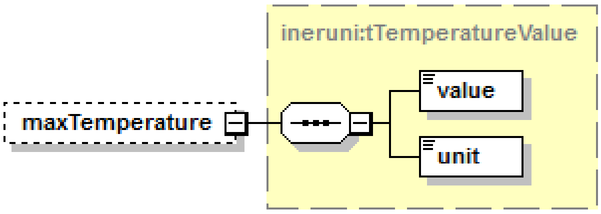
diagram



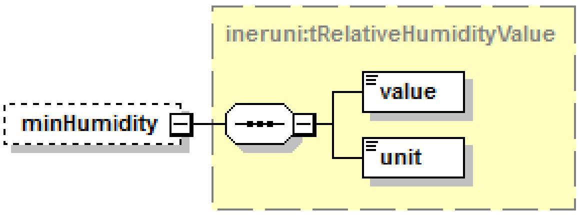

| | |
|---|---|
| type | **ineruni:tTemperatureValue** |
| properties | minOcc 0 |

| | |
|---|---|
| | maxOcc 1<br>content complex |
| children | **value** **unit** |
| source | <element name="minTemperature" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/> |

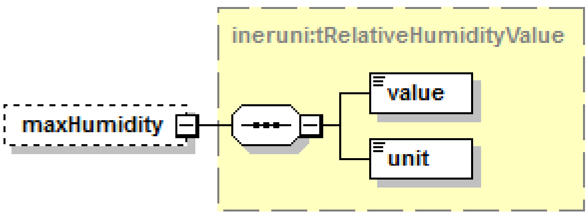
element **tEnvironmentComfortValues/maxTemperature**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTemperatureValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value** **unit** |
| source | <element name="maxTemperature" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/> |

element **tEnvironmentComfortValues/minHumidity**

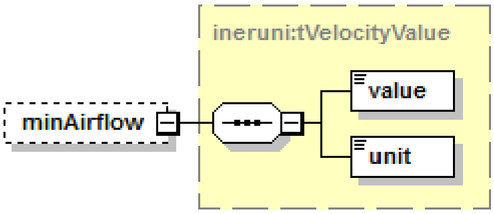
| | |
|---|---|
| diagram |  |
| type | **ineruni:tRelativeHumidityValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value** **unit** |
| source | <element name="minHumidity" type="ineruni:tRelativeHumidityValue" minOccurs="0" maxOccurs="1"/> |

element **tEnvironmentComfortValues/maxHumidity**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tRelativeHumidityValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value** **unit** |
| source | <element name="maxHumidity" type="ineruni:tRelativeHumidityValue" minOccurs="0" maxOccurs="1"/> |

### element **tEnvironmentComfortValues/minAirflow**

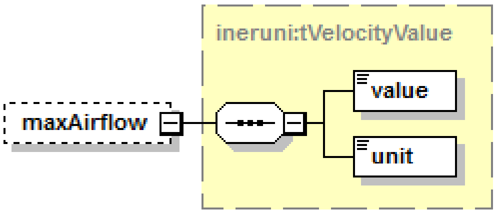
diagram




| | |
|---|---|
| type | **ineruni:tVelocityValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="minAirflow" type="ineruni:tVelocityValue" minOccurs="0" maxOccurs="1"/> |

### element **tEnvironmentComfortValues/maxAirflow**

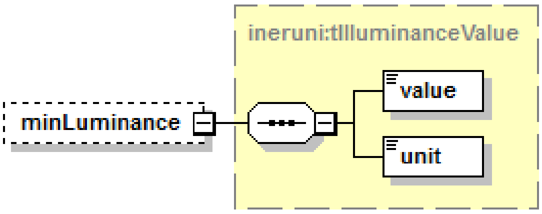
diagram




| | |
|---|---|
| type | **ineruni:tVelocityValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="maxAirflow" type="ineruni:tVelocityValue" minOccurs="0" maxOccurs="1"/> |

### element **tEnvironmentComfortValues/minLuminance**

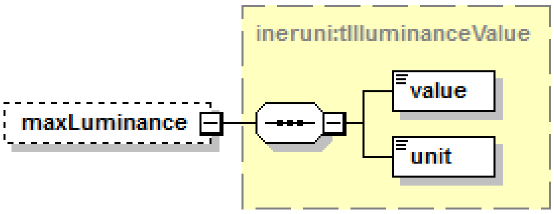
diagram




| | |
|---|---|
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="minLuminance" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

### element **tEnvironmentComfortValues/maxLuminance**

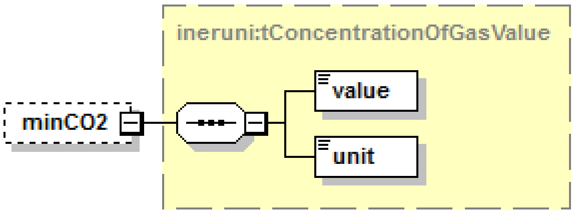
diagram



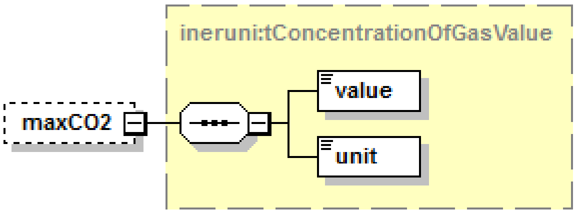

| | |
|---|---|
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="maxLuminance" type="ineruni:tIlluminanceValue" |

| | |
|---|---|
| | minOccurs="0" maxOccurs="1"/> |

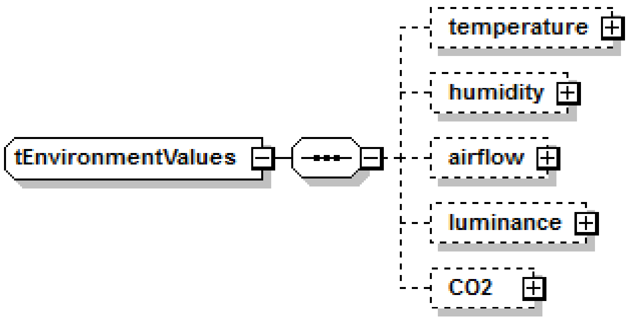
element **tEnvironmentComfortValues/minCO2**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tConcentrationOfGasValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="minCO2" type="ineruni:tConcentrationOfGasValue" minOccurs="0" maxOccurs="1"/> |

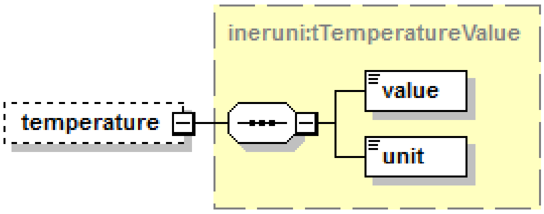
element **tEnvironmentComfortValues/maxCO2**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tConcentrationOfGasValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="maxCO2" type="ineruni:tConcentrationOfGasValue" minOccurs="0" maxOccurs="1"/> |

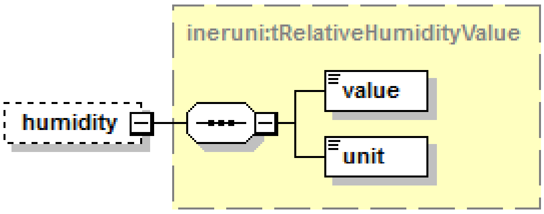
complexType **tEnvironmentValues**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **temperature humidity airflow luminance CO2** |
| source | <complexType name="tEnvironmentValues"><br><sequence><br><element name="temperature" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/><br><element name="humidity" type="ineruni:tRelativeHumidityValue" minOccurs="0" maxOccurs="1"/><br><element name="airflow" type="ineruni:tVelocityValue" minOccurs="0" maxOccurs="1"/><br><element name="luminance" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/><br><element name="CO2" type="ineruni:tConcentrationOfGasValue" minOccurs="0" maxOccurs="1"/><br></sequence><br></complexType> |

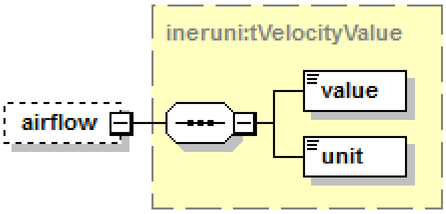
element **tEnvironmentValues/temperature**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTemperatureValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="temperature" type="ineruni:tTemperatureValue" minOccurs="0" maxOccurs="1"/> |

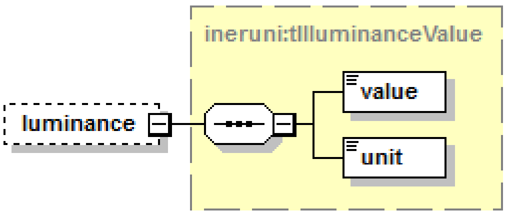
element **tEnvironmentValues/humidity**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tRelativeHumidityValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="humidity" type="ineruni:tRelativeHumidityValue" minOccurs="0" maxOccurs="1"/> |

element **tEnvironmentValues/airflow**

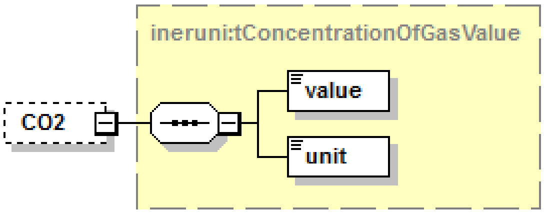
| | |
|---|---|
| diagram |  |
| type | **ineruni:tVelocityValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="airflow" type="ineruni:tVelocityValue" minOccurs="0" maxOccurs="1"/> |

element **tEnvironmentValues/luminance**

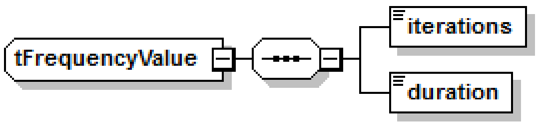
| | |
|---|---|
| diagram |  |
| type | **ineruni:tIlluminanceValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |

| | |
|---|---|
| source | <element name="luminance" type="ineruni:tIlluminanceValue" minOccurs="0" maxOccurs="1"/> |

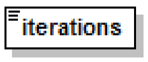
element **tEnvironmentValues/CO2**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tConcentrationOfGasValue** |
| properties | minOcc 0<br>maxOcc 1<br>content complex |
| children | **value unit** |
| source | <element name="CO2" type="ineruni:tConcentrationOfGasValue" minOccurs="0" maxOccurs="1"/> |

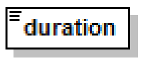
complexType **tFrequencyValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **iterations duration** |
| source | <complexType name="tFrequencyValue"><br><sequence><br><element name="iterations" type="int" minOccurs="1" maxOccurs="1"/><br><element name="duration" type="duration" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

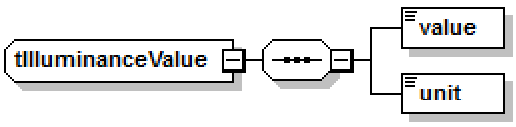
element **tFrequencyValue/iterations**

| | |
|---|---|
| diagram |  |
| type | **int** |
| properties | content simple |
| source | <element name="iterations" type="int" minOccurs="1" maxOccurs="1"/> |

element **tFrequencyValue/duration**

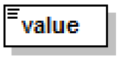
| | |
|---|---|
| diagram |  |
| type | **duration** |
| properties | content simple |
| source | <element name="duration" type="duration" minOccurs="1" maxOccurs="1"/> |

complexType **tIlluminanceValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| used by | elements **tEnvironmentValues/luminance** **tEnvironmentComfortValues/maxLuminance** **tEnvironmentComfortValues/minLuminance** |
| source | <complexType name="tIlluminanceValue"><br><sequence><br><element name="value" type="float"/> |

| | |
|---|---|
| | <element name="unit" type="ineruni:illuminanceUnitEnum"/><br></sequence><br></complexType> |

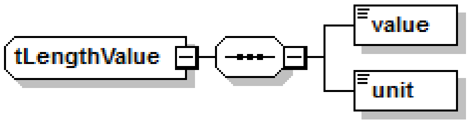
element **tIlluminanceValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

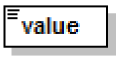
element **tIlluminanceValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:illuminanceUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration Footcandles<br>enumeration Lux<br>enumeration Phot |
| source | <element name="unit" type="ineruni:illuminanceUnitEnum"/> |

complexType **tLengthValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| used by | elements **tPossitionValue/x tPossitionValue/y tPossitionValue/z** |
| source | <complexType name="tLengthValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:lengthUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tLengthValue/value**

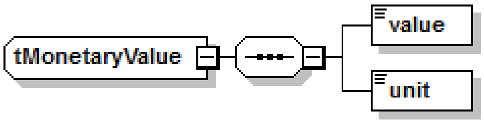
| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

element **tLengthValue/unit**

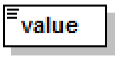
| | |
|---|---|
| diagram |  |
| type | **ineruni:lengthUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration Kilometers<br>enumeration Meters<br>enumeration Centimeters<br>enumeration Millimeters<br>enumeration Miles<br>enumeration Yards<br>enumeration Feet |

| | |
|---|---|
| | enumeration Inches |
| source | <element name="unit" type="ineruni:lengthUnitEnum" minOccurs="1" maxOccurs="1"/> |

complexType **tMonetaryValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tMonetaryValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:monetaryUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

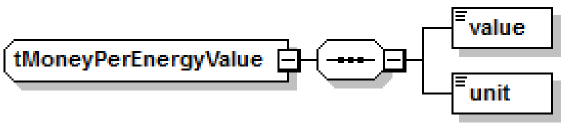
element **tMonetaryValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

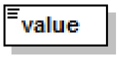
element **tMonetaryValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:monetaryUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration EUR<br>enumeration SEK<br>enumeration CZK<br>enumeration DKK<br>enumeration NOK<br>enumeration CHF<br>enumeration HRK |
| source | <element name="unit" type="ineruni:monetaryUnitEnum" minOccurs="1" maxOccurs="1"/> |

complexType **tMoneyPerEnergyValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tMoneyPerEnergyValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:moneyPerEnergyUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

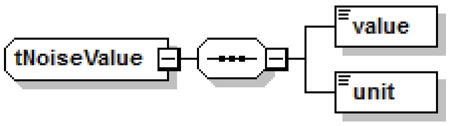
element **tMoneyPerEnergyValue/value**

| | |
|---|---|
| diagram | <br> |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

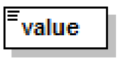
element **tMoneyPerEnergyValue/unit**

| | |
|---|---|
| diagram | <br> |
| type | **ineruni:moneyPerEnergyUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration deviceUsed<br>enumeration spaceEnvironmentChange<br>enumeration externalEnvironmentChanged<br>enumeration HVACChange<br>enumeration LightingChange<br>enumeration SpaceOccupancy<br>enumeration SpaceMovement<br>enumeration SpaceOccupantPossitions<br>enumeration Current<br>enumeration Voltage<br>enumeration Energy |
| source | <element name="unit" type="ineruni:moneyPerEnergyUnitEnum" minOccurs="1" maxOccurs="1"/> |

complexType **tNoiseValue**

| | |
|---|---|
| diagram | <br> |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | <complexType name="tNoiseValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:noiseUnitEnum"/><br></sequence><br></complexType> |

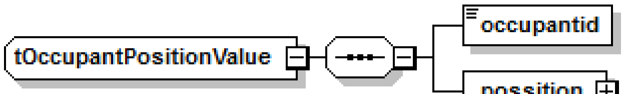
element **tNoiseValue/value**

| | |
|---|---|
| diagram | <br> |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

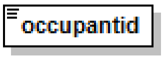
element **tNoiseValue/unit**

| | |
|---|---|
| diagram | <br> |
| type | **ineruni:noiseUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration dB |
| source | <element name="unit" type="ineruni:noiseUnitEnum"/> |

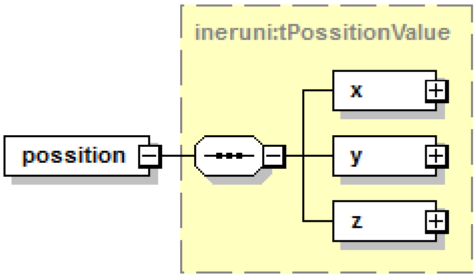
complexType **tOccupantPositionValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **occupantid** **possition** |
| source | <complexType name="tOccupantPositionValue"><br><sequence><br><element name="occupantid" type="int" minOccurs="1" maxOccurs="1"/><br><element name="possition" type="ineruni:tPositionValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

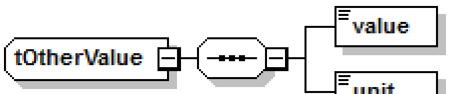
element **tOccupantPositionValue/occupantid**

| | |
|---|---|
| diagram |  |
| type | **int** |
| properties | content simple |
| source | <element name="occupantid" type="int" minOccurs="1" maxOccurs="1"/> |

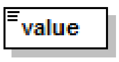
element **tOccupantPositionValue/possition**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tPossitionValue** |
| properties | content complex |
| children | **x y z** |
| source | <element name="possition" type="ineruni:tPossitionValue" minOccurs="1" maxOccurs="1"/> |

complexType **tOtherValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| source | <complexType name="tOtherValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="string"/><br></sequence><br></complexType> |

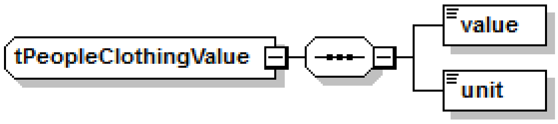
element **tOtherValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

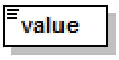
element **tOtherValue/unit**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="unit" type="string"/> |

complexType **tPeopleClothingValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | [value](#) [unit](#) |
| source | <complexType name="tPeopleClothingValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:clothingUnitEnum"/><br></sequence><br></complexType> |

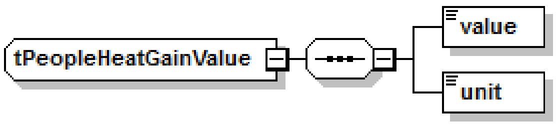
element **tPeopleClothingValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

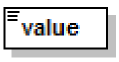
element **tPeopleClothingValue/unit**

| | |
|---|---|
| diagram |  |
| type | **[ineruni:clothingUnitEnum](#)** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration Clo |
| source | <element name="unit" type="ineruni:clothingUnitEnum"/> |

complexType **tPeopleHeatGainValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | [value](#) [unit](#) |
| source | <complexType name="tPeopleHeatGainValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:peopleHeatGainUnitEnum"/><br></sequence><br></complexType> |

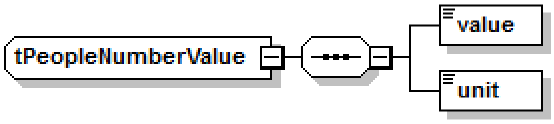
element **tPeopleHeatGainValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

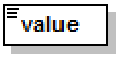
element **tPeopleHeatGainValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:peopleHeatGainUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration WattPerPerson<br>enumeration BtuPerHourPerson<br>enumeration WattPerSquareMeter<br>enumeration Met |
| source | <element name="unit" type="ineruni:peopleHeatGainUnitEnum"/> |

complexType **tPeopleNumberValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | <complexType name="tPeopleNumberValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:peopleNumberUnitEnum"/><br></sequence><br></complexType> |

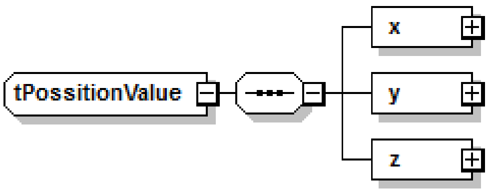
element **tPeopleNumberValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

element **tPeopleNumberValue/unit**

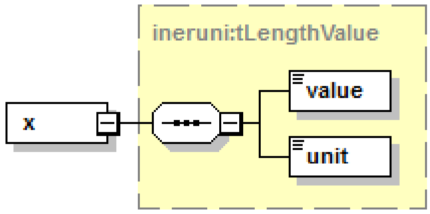
| | |
|---|---|
| diagram |  |
| type | **ineruni:peopleNumberUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration NumberOfPeople<br>enumeration SquareFtPerPerson<br>enumeration SquareMPerPerson |
| source | <element name="unit" type="ineruni:peopleNumberUnitEnum"/> |

complexType **tPossitionValue**

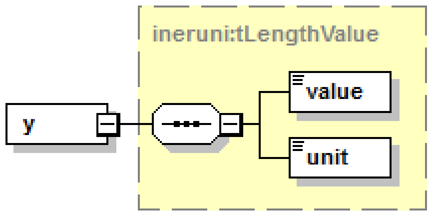
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **x y z** |
| used by | elements **tOccupantPositionValue/possition tTrajectory/possition** |
| source | <complexType name="tPossitionValue"><br><sequence><br><element name="x" type="ineruni:tLengthValue" minOccurs="1" maxOccurs="1"/> |

| | |
|---|---|
| | <element name="y" type="ineruni:tLengthValue" minOccurs="1" maxOccurs="1"/><br><element name="z" type="ineruni:tLengthValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

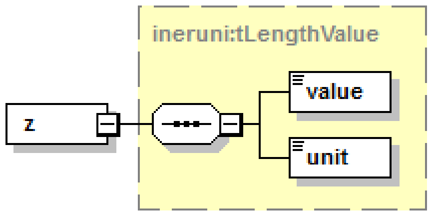
element **tPossitionValue/x**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tLengthValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="x" type="ineruni:tLengthValue" minOccurs="1" maxOccurs="1"/> |

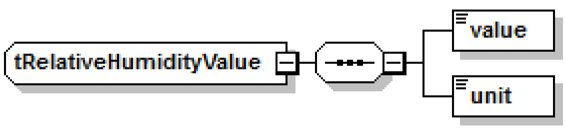
element **tPossitionValue/y**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tLengthValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="y" type="ineruni:tLengthValue" minOccurs="1" maxOccurs="1"/> |

element **tPossitionValue/z**

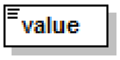
| | |
|---|---|
| diagram |  |
| type | **ineruni:tLengthValue** |
| properties | content complex |
| children | **value unit** |
| source | <element name="z" type="ineruni:tLengthValue" minOccurs="1" maxOccurs="1"/> |

complexType **tRelativeHumidityValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| used by | elements **tEnvironmentValues/humidity** **tEnvironmentComfortValues/maxHumidity** **tEnvironmentComfortValues/minHumidity** |
| source | <complexType name="tRelativeHumidityValue"><br><sequence> |

| | |
|---|---|
| | <element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:unitlessUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

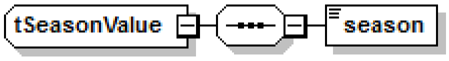
element **tRelativeHumidityValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

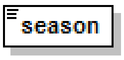
element **tRelativeHumidityValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:unitlessUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration Percent<br>enumeration Fraction |
| source | <element name="unit" type="ineruni:unitlessUnitEnum" minOccurs="1" maxOccurs="1"/> |

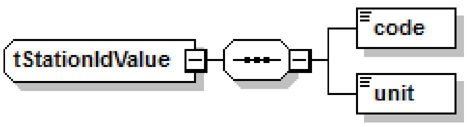
complexType **tSeasonValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **season** |
| source | <complexType name="tSeasonValue"><br><sequence><br><element name="season" type="ineruni:seasonUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

element **tSeasonValue/season**

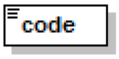
| | |
|---|---|
| diagram |  |
| type | **ineruni:seasonUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration Summer<br>enumeration Autumn<br>enumeration Winter<br>enumeration Spring |
| source | <element name="season" type="ineruni:seasonUnitEnum" minOccurs="1" maxOccurs="1"/> |

complexType **tStationIdValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **code unit** |
| source | <complexType name="tStationIdValue"><br><sequence><br><element name="code" type="string" minOccurs="1" maxOccurs="1"/> |

| | |
|---|---|
| | <element name="unit" type="ineruni:stationIdTypeEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

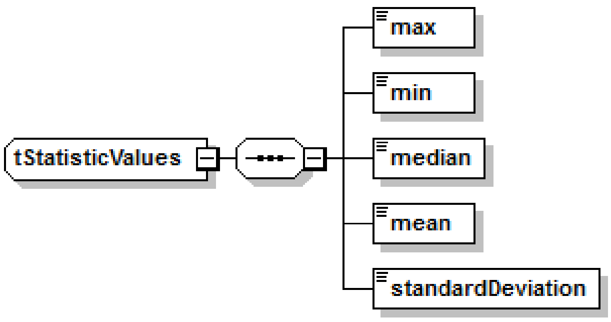
element **tStationIdValue/code**

| | |
|---|---|
| diagram |  |
| type | **string** |
| properties | content simple |
| source | <element name="code" type="string" minOccurs="1" maxOccurs="1"/> |

element **tStationIdValue/unit**

| | | | |
|---|---|---|---|
| diagram |  | | |
| type | **ineruni:stationIdTypeEnum** | | |
| properties | content simple | | |
| facets | Kind | Value | Annotation |
| | enumeration | WMO | documentation<br>A five-digit station identifier assigned by the World Meteorological Organization (WMO), used for international weather data exchange and station documentation. |
| | enumeration | WBAN | documentation<br>Weather-Bureau-Army-Navy station id. A five-digit station identifier used at National Climatic Data Center (NCDC) for digital data storage and general station identification purposes. |
| | enumeration | ICAO | documentation<br>A 4-letter indicator for geographical locations throughout the world, managed by the International Civil Aviation Organization (ICAO). |
| | enumeration | NWSLI | documentation<br>A 3- to 5-character alpha-numeric identifier assigned by the US National Weather Service (NWS), used for site identification in the NWS communications systems (e.g. AFOS, ROSA, SHEF, etc.). |
| | enumeration | FAACallSign | documentation<br>An alpha-numeric identifier of up to 4-characters managed by the USDT Federal Aviation Administration (FAA), used for site identification of airports, weather stations and other sites vital to navigation. |
| | enumeration | COOP | documentation<br>A 6-digit number identifying a station in the NWS Cooperative Network. |
| source | <element name="unit" type="ineruni:stationIdTypeEnum" minOccurs="1" maxOccurs="1"/> | | |

complexType **tStatisticValues**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **max min median mean standardDeviation** |
| used by | element **tTimeSerieStatisticValue/value** |
| source | <complexType name="tStatisticValues"><br><sequence><br><element name="max" type="float" minOccurs="1" maxOccurs="1"/><br><element name="min" type="float" minOccurs="1" maxOccurs="1"/><br><element name="median" type="float" minOccurs="1" maxOccurs="1"/><br><element name="mean" type="float" minOccurs="1" maxOccurs="1"/><br><element name="standardDeviation" type="float" minOccurs="1" |

| | |
|---|---|
| | maxOccurs="1"/><br></sequence><br></complexType> |

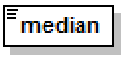
element **tStatisticValues/max**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="max" type="float" minOccurs="1" maxOccurs="1"/> |

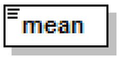
element **tStatisticValues/min**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="min" type="float" minOccurs="1" maxOccurs="1"/> |

element **tStatisticValues/median**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="median" type="float" minOccurs="1" maxOccurs="1"/> |

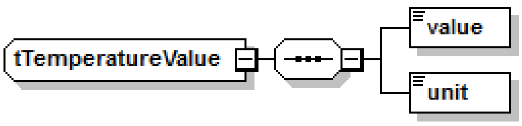
element **tStatisticValues/mean**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="mean" type="float" minOccurs="1" maxOccurs="1"/> |

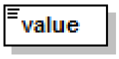
element **tStatisticValues/standardDeviation**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="standardDeviation" type="float" minOccurs="1" maxOccurs="1"/> |

complexType **tTemperatureValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| used by | elements **tEnvironmentComfortValues/maxTemperature**<br>**tEnvironmentComfortValues/minTemperature**<br>**tEnvironmentValues/temperature** |
| source | <complexType name="tTemperatureValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:temperatureUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

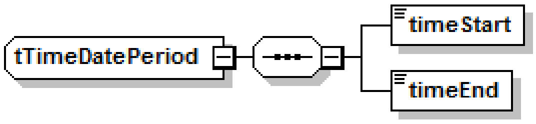
element **tTemperatureValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

element **tTemperatureValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:temperatureUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration F<br>enumeration C<br>enumeration K<br>enumeration R |
| source | <element name="unit" type="ineruni:temperatureUnitEnum" minOccurs="1" maxOccurs="1"/> |

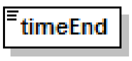
complexType **tTimeDatePeriod**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeStart** **timeEnd** |
| source | <complexType name="tTimeDatePeriod"><br><sequence><br><element name="timeStart" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="timeEnd" type="dateTime" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

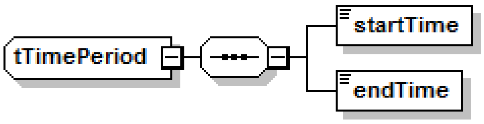
element **tTimeDatePeriod/timeStart**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStart" type="dateTime" minOccurs="1" maxOccurs="1"/> |

element **tTimeDatePeriod/timeEnd**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeEnd" type="dateTime" minOccurs="1" maxOccurs="1"/> |

complexType **tTimePeriod**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **startTime** **endTime** |
| source | <complexType name="tTimePeriod"> |

| | |
|---|---|
| | <sequence><br><element name="startTime" type="time" minOccurs="1" maxOccurs="1"/><br><element name="endTime" type="time" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

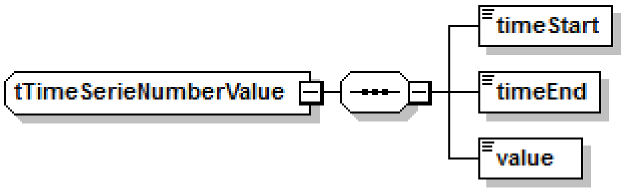
element **tTimePeriod/startTime**

| | |
|---|---|
| diagram |  |
| type | **time** |
| properties | content simple |
| source | <element name="startTime" type="time" minOccurs="1" maxOccurs="1"/> |

element **tTimePeriod/endTime**

| | |
|---|---|
| diagram |  |
| type | **time** |
| properties | content simple |
| source | <element name="endTime" type="time" minOccurs="1" maxOccurs="1"/> |

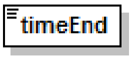
complexType **tTimeSerieNumberValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeStart** **timeEnd** **value** |
| used by | element **tTimeSeriesOfNumbersValue/timeSeriesValue** |
| source | <complexType name="tTimeSerieNumberValue"><br><sequence><br><element name="timeStart" type="dateTime"/><br><element name="timeEnd" type="dateTime"/><br><element name="value" type="int"/><br></sequence><br></complexType> |

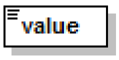
element **tTimeSerieNumberValue/timeStart**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStart" type="dateTime"/> |

element **tTimeSerieNumberValue/timeEnd**

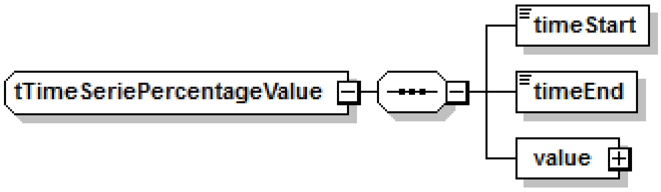
| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeEnd" type="dateTime"/> |

element **tTimeSerieNumberValue/value**

| | |
|---|---|
| diagram |  |
| type | **int** |
| properties | content simple |
| source | <element name="value" type="int"/> |

### complexType **tTimeSeriePercentageValue**

diagram




| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeStart** **timeEnd** **value** |
| used by | element **tTimeSeriesOfPercentagesValue/timeSeriesValue** |

source

```
<complexType name="tTimeSeriePercentageValue">
  <sequence>
    <element name="timeStart" type="dateTime"/>
    <element name="timeEnd" type="dateTime"/>
    <element name="value" type="ineruni:tUnitlessValue"/>
  </sequence>
</complexType>
```

### element **tTimeSeriePercentageValue/timeStart**

diagram

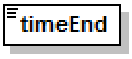

| | |
|---|---|
| type | **dateTime** |
| properties | content simple |

source

```
<element name="timeStart" type="dateTime"/>
```

### element **tTimeSeriePercentageValue/timeEnd**

diagram



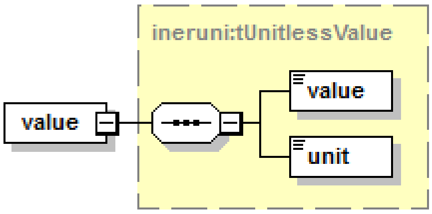

| | |
|---|---|
| type | **dateTime** |
| properties | content simple |

source

```
<element name="timeEnd" type="dateTime"/>
```

### element **tTimeSeriePercentageValue/value**

diagram



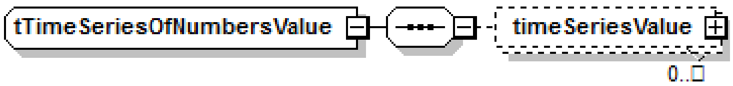

| | |
|---|---|
| type | **ineruni:tUnitlessValue** |
| properties | content complex |
| children | **value** **unit** |

source

```
<element name="value" type="ineruni:tUnitlessValue"/>
```

### complexType **tTimeSeriesOfNumbersValue**

diagram

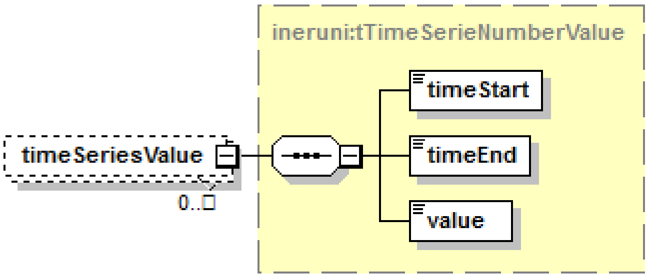



| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeSeriesValue** |

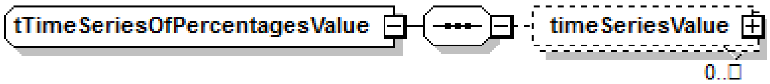
source

```
<complexType name="tTimeSeriesOfNumbersValue">
  <sequence>
    <element name="timeSeriesValue" type="ineruni:tTimeSerieNumberValue"
minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

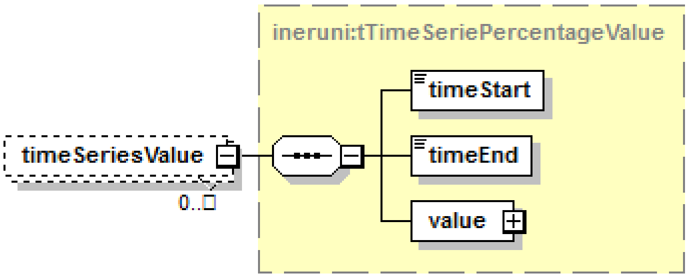
element **tTimeSeriesOfNumbersValue/timeSeriesValue**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeSerieNumberValue** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **timeStart** **timeEnd** **value** |
| source | <element name="timeSeriesValue" type="ineruni:tTimeSerieNumberValue" minOccurs="0" maxOccurs="unbounded"/> |

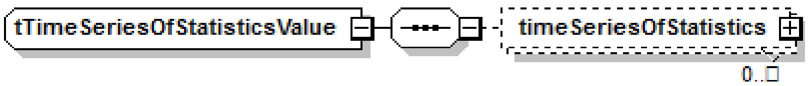
complexType **tTimeSeriesOfPercentagesValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeSeriesValue** |
| source | <complexType name="tTimeSeriesOfPercentagesValue"><br><sequence><br><element name="timeSeriesValue" type="ineruni:tTimeSeriePercentageValue" minOccurs="0" maxOccurs="unbounded"/><br></sequence><br></complexType> |

element **tTimeSeriesOfPercentagesValue/timeSeriesValue**

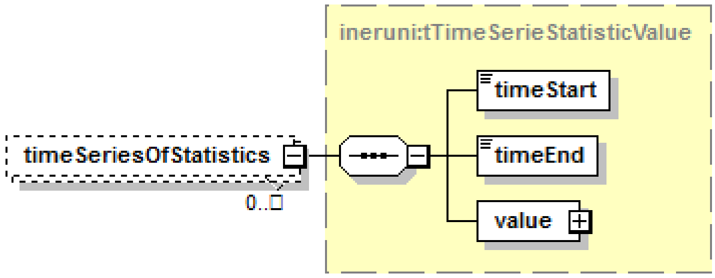
| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeSeriePercentageValue** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **timeStart** **timeEnd** **value** |
| source | <element name="timeSeriesValue" type="ineruni:tTimeSeriePercentageValue" minOccurs="0" maxOccurs="unbounded"/> |

complexType **tTimeSeriesOfStatisticsValue**

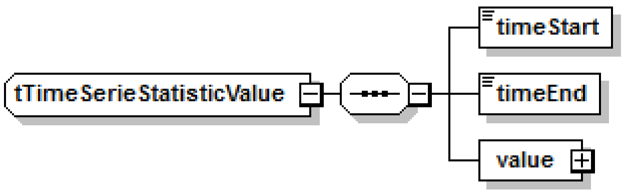
| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeSeriesOfStatistics** |
| source | <complexType name="tTimeSeriesOfStatisticsValue"><br><sequence><br><element name="timeSeriesOfStatistics" type="ineruni:tTimeSerieStatisticValue" minOccurs="0" maxOccurs="unbounded"/> |

| | |
|---|---|
| | </sequence><br></complexType> |

element **tTimeSeriesOfStatisticsValue/timeSeriesOfStatistics**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tTimeSerieStatisticValue** |
| properties | minOcc 0<br>maxOcc unbounded<br>content complex |
| children | **timeStart** **timeEnd** **value** |
| source | <element name="timeSeriesOfStatistics" type="ineruni:tTimeSerieStatisticValue" minOccurs="0" maxOccurs="unbounded"/> |

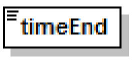
complexType **tTimeSerieStatisticValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeStart** **timeEnd** **value** |
| used by | element **tTimeSeriesOfStatisticsValue/timeSeriesOfStatistics** |
| source | <complexType name="tTimeSerieStatisticValue"><br><sequence><br><element name="timeStart" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="timeEnd" type="dateTime" minOccurs="1" maxOccurs="1"/><br><element name="value" type="ineruni:tStatisticValues" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

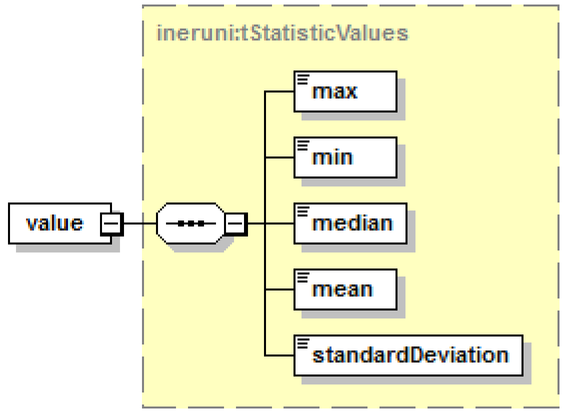
element **tTimeSerieStatisticValue/timeStart**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeStart" type="dateTime" minOccurs="1" maxOccurs="1"/> |

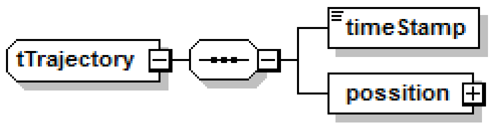
element **tTimeSerieStatisticValue/timeEnd**

| | |
|---|---|
| diagram |  |
| type | **dateTime** |
| properties | content simple |
| source | <element name="timeEnd" type="dateTime" minOccurs="1" maxOccurs="1"/> |

element **tTimeSerieStatisticValue/value**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tStatisticValues** |
| properties | content complex |
| children | **max min median mean standardDeviation** |
| source | <element name="value" type="ineruni:tStatisticValues" minOccurs="1" maxOccurs="1"/> |

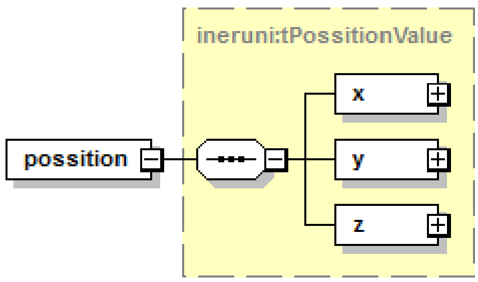
complexType **tTrajectory**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **timeStamp possition** |
| source | <complexType name="tTrajectory"><br><sequence><br><element name="timeStamp" type="double" minOccurs="1" maxOccurs="1"/><br><element name="possition" type="ineruni:tPossitionValue" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

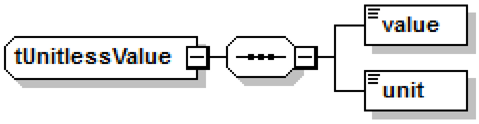
element **tTrajectory/timeStamp**

| | |
|---|---|
| diagram |  |
| type | **double** |
| properties | content simple |
| source | <element name="timeStamp" type="double" minOccurs="1" maxOccurs="1"/> |

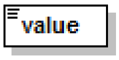
element **tTrajectory/possition**

| | |
|---|---|
| diagram |  |
| type | **ineruni:tPossitionValue** |
| properties | content complex |
| children | **x y z** |
| source | <element name="possition" type="ineruni:tPossitionValue" minOccurs="1" maxOccurs="1"/> |

complexType **tUnitlessValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| used by | element **tTimeSeriePercentageValue/value** |
| source | <complexType name="tUnitlessValue"><br><sequence><br><element name="value" type="float" minOccurs="1" maxOccurs="1"/><br><element name="unit" type="ineruni:unitlessUnitEnum" minOccurs="1" maxOccurs="1"/><br></sequence><br></complexType> |

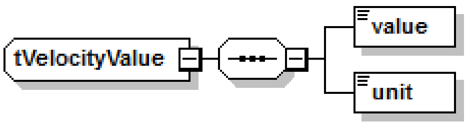
element **tUnitlessValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float" minOccurs="1" maxOccurs="1"/> |

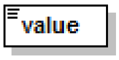
element **tUnitlessValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:unitlessUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration / Percent<br>enumeration / Fraction |
| source | <element name="unit" type="ineruni:unitlessUnitEnum" minOccurs="1" maxOccurs="1"/> |

complexType **tVelocityValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value** **unit** |
| used by | elements **tEnvironmentValues/airflow** **tEnvironmentComfortValues/maxAirflow** **tEnvironmentComfortValues/minAirflow** |
| source | <complexType name="tVelocityValue"><br><sequence><br><element name="value" type="float"/><br><element name="unit" type="ineruni:velocityUnitEnum"/><br></sequence><br></complexType> |

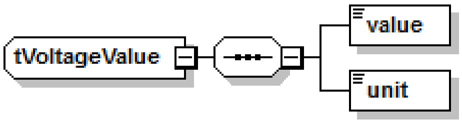
element **tVelocityValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | <element name="value" type="float"/> |

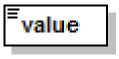
element **tVelocityValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:velocityUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration MetersPerSecond<br>enumeration MilesPerHour<br>enumeration Knots |
| source | `<element name="unit" type="ineruni:velocityUnitEnum"/>` |

complexType **tVoltageValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | `<complexType name="tVoltageValue">`<br>`<sequence>`<br>`<element name="value" type="float" minOccurs="1" maxOccurs="1"/>`<br>`<element name="unit" type="ineruni:VoltageUnitEnum" minOccurs="1" maxOccurs="1"/>`<br>`</sequence>`<br>`</complexType>` |

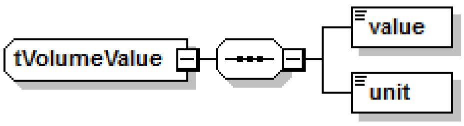
element **tVoltageValue/value**

| | |
|---|---|
| diagram |  |
| type | **float** |
| properties | content simple |
| source | `<element name="value" type="float" minOccurs="1" maxOccurs="1"/>` |

element **tVoltageValue/unit**

| | |
|---|---|
| diagram |  |
| type | **ineruni:VoltageUnitEnum** |
| properties | content simple |
| facets | Kind Value Annotation<br>enumeration Volts<br>enumeration KiloVolts |
| source | `<element name="unit" type="ineruni:VoltageUnitEnum" minOccurs="1" maxOccurs="1"/>` |

complexType **tVolumeValue**

| | |
|---|---|
| diagram |  |
| namespace | http://www.inertia.eu/2013/schema/units/ |
| children | **value unit** |
| source | `<complexType name="tVolumeValue">`<br>`<sequence>`<br>`<element name="value" type="float"/>`<br>`<element name="unit" type="ineruni:volumeUnitEnum"/>`<br>`</sequence>`<br>`</complexType>` |

element **tVolumeValue/value**

| | |
|---|---|
| diagram | value |
| type | **float** |
| properties | content simple |
| source | `<element name="value" type="float"/>` |

element **tVolumeValue/unit**

| | |
|---|---|
| diagram | unit |
| type | **ineruni:volumeUnitEnum** |
| properties | content simple |
| facets | Kind / Value / Annotation<br>enumeration CubicKilometers<br>enumeration CubicMeters<br>enumeration CubicCentimeters<br>enumeration CubicMillimeters<br>enumeration CubicMiles<br>enumeration CubicYards<br>enumeration CubicFeet<br>enumeration CubicInches |
| source | `<element name="unit" type="ineruni:volumeUnitEnum"/>` |

simpleType **areaUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tAreaValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration SquareKilometers<br>enumeration SquareMeters<br>enumeration SquareCentimeters<br>enumeration SquareMillimeters<br>enumeration SquareMiles<br>enumeration SquareYards<br>enumeration SquareFeet<br>enumeration SquareInches |

source

```
<simpleType name="areaUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="SquareKilometers"/>
    <enumeration value="SquareMeters"/>
    <enumeration value="SquareCentimeters"/>
    <enumeration value="SquareMillimeters"/>
    <enumeration value="SquareMiles"/>
    <enumeration value="SquareYards"/>
    <enumeration value="SquareFeet"/>
    <enumeration value="SquareInches"/>
  </restriction>
</simpleType>
```

simpleType **buildingTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| facets | Kind / Value / Annotation<br>enumeration AutomotiveFacility<br>enumeration ConventionCenter<br>enumeration Courthouse |

| | | | |
|---|---|---|---|
| | enumeration | DiningBarLoungeOrLeisure | |
| | enumeration | DiningCafeteriaFastFood | |
| | enumeration | DiningFamily | |
| | enumeration | Dormitory | |
| | enumeration | ExerciseCenter | |
| | enumeration | FireStation | |
| | enumeration | Gymnasium | |
| | enumeration | HospitalOrHealthcare | |
| | enumeration | Hotel | |
| | enumeration | Library | |
| | enumeration | Manufacturing | |
| | enumeration | Motel | |
| | enumeration | MotionPictureTheatre | |
| | enumeration | MultiFamily | |
| | enumeration | Museum | |
| | enumeration | Office | |
| | enumeration | ParkingGarage | |
| | enumeration | Penitentiary | |
| | enumeration | PerformingArtsTheater | |
| | enumeration | PoliceStation | |
| | enumeration | PostOffice | |
| | enumeration | ReligiousBuilding | |
| | enumeration | Retail | |
| | enumeration | SchoolOrUniversity | |
| | enumeration | SingleFamily | |
| | enumeration | SportsArena | |
| | enumeration | TownHall | |
| | enumeration | Transportation | |
| | enumeration | Unknown | documentation<br>Enumeration value to be used when there is no building type information available. |
| | enumeration | Warehouse | |
| | enumeration | Workshop | |
| annotation | documentation<br>Use to specify the most predominant building use type. | | |
| source | (see below) | | |

```
<simpleType name="buildingTypeEnum">
  <annotation>
    <documentation>Use to specify the most predominant building use type.</documentation>
  </annotation>
  <restriction base="NMTOKEN">
    <enumeration value="AutomotiveFacility"/>
    <enumeration value="ConventionCenter"/>
    <enumeration value="Courthouse"/>
    <enumeration value="DiningBarLoungeOrLeisure"/>
    <enumeration value="DiningCafeteriaFastFood"/>
    <enumeration value="DiningFamily"/>
    <enumeration value="Dormitory"/>
    <enumeration value="ExerciseCenter"/>
    <enumeration value="FireStation"/>
    <enumeration value="Gymnasium"/>
    <enumeration value="HospitalOrHealthcare"/>
    <enumeration value="Hotel"/>
    <enumeration value="Library"/>
    <enumeration value="Manufacturing"/>
    <enumeration value="Motel"/>
```

| | |
|---|---|
| | <enumeration value="MotionPictureTheatre"/><br><enumeration value="MultiFamily"/><br><enumeration value="Museum"/><br><enumeration value="Office"/><br><enumeration value="ParkingGarage"/><br><enumeration value="Penitentiary"/><br><enumeration value="PerformingArtsTheater"/><br><enumeration value="PoliceStation"/><br><enumeration value="PostOffice"/><br><enumeration value="ReligiousBuilding"/><br><enumeration value="Retail"/><br><enumeration value="SchoolOrUniversity"/><br><enumeration value="SingleFamily"/><br><enumeration value="SportsArena"/><br><enumeration value="TownHall"/><br><enumeration value="Transportation"/><br><enumeration value="Unknown"><br><annotation><br><documentation>Enumeration value to be used when there is no building type information available.</documentation><br></annotation><br></enumeration><br><enumeration value="Warehouse"/><br><enumeration value="Workshop"/><br></restriction><br></simpleType> |

simpleType **clothingUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tPeopleClothingValue/unit** |
| facets | Kind Value Annotation<br>enumeration Clo |
| source | <simpleType name="clothingUnitEnum"><br><restriction base="NMTOKEN"><br><enumeration value="Clo"/><br></restriction><br></simpleType> |

simpleType **concentrationOfGasUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tConcentrationOfGasValue/unit** |
| facets | Kind Value Annotation<br>enumeration PartsPerMillion |
| source | <simpleType name="concentrationOfGasUnitEnum"><br><restriction base="NMTOKEN"><br><enumeration value="PartsPerMillion"/><br></restriction><br></simpleType> |

simpleType **CurrentUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tCurrentValue/unit** |
| facets | Kind Value Annotation<br>enumeration Ampere<br>enumeration milliAmpere |
| source | <simpleType name="CurrentUnitEnum"><br><restriction base="NMTOKEN"> |

| | |
|---|---|
| | `<enumeration value="Ampere"/>`<br>`<enumeration value="milliAmpere"/>`<br>`</restriction>`<br>`</simpleType>` |

simpleType **efficiencyTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tEfficiencyValue/unit** |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | COP | documentation<br>Coefficient of Performance |
| enumeration | EER | documentation<br>Energy Efficiency Ratio |
| enumeration | SEER | documentation<br>Seasonal Energy Efficiency Ratio |
| enumeration | AFUE | documentation<br>Annual Fuel Utilization Efficiency |
| enumeration | HSPF | documentation<br>Heating Seasonal Performance Factor |
| enumeration | effectiveness | |
| enumeration | EnergyFactor | |
| enumeration | ThermalEff | documentation<br>Thermal Efficiency |
| enumeration | MotorEff | documentation<br>Motor Efficiency |
| enumeration | FanEff | documentation<br>Fan Efficiency |
| enumeration | MechanicalEff | documentation<br>Mechanical Efficiency |
| enumeration | BoilerEff | documentation<br>Boiler Efficiency |
| enumeration | kWPerTon | |
| enumeration | kWPerkW | |
| enumeration | BTUPerHourPerF | |
| enumeration | kWPerC | |

source

```
<simpleType name="efficiencyTypeEnum">
 <restriction base="NMTOKEN">
  <enumeration value="COP">
   <annotation>
    <documentation>Coefficient of Performance</documentation>
   </annotation>
  </enumeration>
  <enumeration value="EER">
   <annotation>
    <documentation>Energy Efficiency Ratio</documentation>
   </annotation>
  </enumeration>
  <enumeration value="SEER">
   <annotation>
    <documentation>Seasonal Energy Efficiency Ratio</documentation>
   </annotation>
  </enumeration>
  <enumeration value="AFUE">
   <annotation>
    <documentation>Annual Fuel Utilization Efficiency</documentation>
   </annotation>
  </enumeration>
  <enumeration value="HSPF">
   <annotation>
    <documentation>Heating Seasonal Performance Factor</documentation>
   </annotation>
  </enumeration>
  <enumeration value="effectiveness"/>
  <enumeration value="EnergyFactor"/>
```

| | |
|---|---|
| | ```<enumeration value="ThermalEff"><annotation><documentation>Thermal Efficiency</documentation></annotation></enumeration><enumeration value="MotorEff"><annotation><documentation>Motor Efficiency</documentation></annotation></enumeration><enumeration value="FanEff"><annotation><documentation>Fan Efficiency</documentation></annotation></enumeration><enumeration value="MechanicalEff"><annotation><documentation>Mechanical Efficiency</documentation></annotation></enumeration><enumeration value="BoilerEff"><annotation><documentation>Boiler Efficiency</documentation></annotation></enumeration><enumeration value="kWPerTon"/><enumeration value="kWPerkW"/><enumeration value="BTUPerHourPerF"/><enumeration value="kWPerC"/></restriction></simpleType>``` |

simpleType **emissionUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tEmissionValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration KilogramsPerkWh<br>enumeration PoundsPerHour<br>enumeration KilogramsPerHour<br>enumeration GramsPerHour |
| source | ```<simpleType name="emissionUnitEnum"><restriction base="NMTOKEN"><enumeration value="KilogramsPerkWh"/><enumeration value="PoundsPerHour"/><enumeration value="KilogramsPerHour"/><enumeration value="GramsPerHour"/></restriction></simpleType>``` |

simpleType **energyPerAreaUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tEnergyPerAreaValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration KilowattHoursPerSquareMeter<br>enumeration JoulesPerSquareMeter |
| source | ```<simpleType name="energyPerAreaUnitEnum"><restriction base="NMTOKEN"><enumeration value="KilowattHoursPerSquareMeter"/>``` |

```xml
    <enumeration value="JoulesPerSquareMeter"/>
  </restriction>
</simpleType>
```

simpleType **energyPerTimeUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tEnergyValuePerTime/unit** |
| facets | Kind / Value / Annotation |
| | enumeration / KilowattHoursPerMinute / |
| | enumeration / KilowattHoursPersecond / |
| | enumeration / JoulesPerSecond / |
| | enumeration / Watt / |

source:

```xml
<simpleType name="energyPerTimeUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="KilowattHoursPerMinute"/>
    <enumeration value="KilowattHoursPersecond"/>
    <enumeration value="JoulesPerSecond"/>
    <enumeration value="Watt"/>
  </restriction>
</simpleType>
```

simpleType **energyUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tEnergyValue/unit** |
| facets | Kind / Value / Annotation |
| | enumeration / KilowattHours / |
| | enumeration / Joules / |
| | enumeration / BTU / |
| | enumeration / Therms / |

source:

```xml
<simpleType name="energyUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="KilowattHours"/>
    <enumeration value="Joules"/>
    <enumeration value="BTU"/>
    <enumeration value="Therms"/>
  </restriction>
</simpleType>
```

simpleType **eventTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| facets | Kind / Value / Annotation |
| | enumeration / EURPerkWh / |
| | enumeration / SEKPerkWh / |
| | enumeration / CZKPerkWh / |
| | enumeration / DKKPerkWh / |
| | enumeration / NOKPerkWh / |
| | enumeration / CHFPerkWh / |
| | enumeration / HRKPerkWh / |

source:

```xml
<simpleType name="eventTypeEnum">
  <restriction base="NMTOKEN">
    <enumeration value="EURPerkWh"/>
    <enumeration value="SEKPerkWh"/>
    <enumeration value="CZKPerkWh"/>
    <enumeration value="DKKPerkWh"/>
```

| | |
|---|---|
| | <enumeration value="NOKPerkWh"/><br><enumeration value="CHFPerkWh"/><br><enumeration value="HRKPerkWh"/><br></restriction><br></simpleType> |

simpleType **illuminanceUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tIlluminanceValue/unit** |
| facets | Kind Value Annotation<br>enumeration Footcandles<br>enumeration Lux<br>enumeration Phot |
| source | <simpleType name="illuminanceUnitEnum"><br><restriction base="NMTOKEN"><br><enumeration value="Footcandles"/><br><enumeration value="Lux"/><br><enumeration value="Phot"/><br></restriction><br></simpleType> |

simpleType **intEquipTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| facets | Kind Value Annotation<br>enumeration GeneralPlugload<br>enumeration Computer<br>enumeration Copier<br>enumeration Refrigerator<br>enumeration Stove<br>enumeration Cooktop<br>enumeration Fryer<br>enumeration Freezer<br>enumeration ClothesDryer<br>enumeration ClothesWasher<br>enumeration Dishwasher<br>enumeration TV<br>enumeration VCR<br>enumeration Microwave<br>enumeration Fan<br>enumeration AudioEquip<br>enumeration MiscEquip<br>enumeration Toilet<br>enumeration Urinal<br>enumeration Shower<br>enumeration Sink<br>enumeration Pool<br>enumeration HotTub<br>enumeration Sauna<br>enumeration BatteryCharger<br>enumeration Furnace<br>enumeration IndustrialEquip<br>enumeration Printer |

| | | |
|---|---|---|
| | enumeration | Vending |
| | enumeration | Elevator |
| | enumeration | Escalator |
| | enumeration | GeneralLaundryEquip |
| | enumeration | GeneralKitchenEquip |
| | enumeration | GeneralMedicalEquip |
| | enumeration | GeneralOfficeEquip |
| | enumeration | GeneralEntertainmentEquip |
| | enumeration | GeneralRefrigerationEquip |
| | enumeration | GeneralFitnessEquip |
| | enumeration | GeneralVideoEquip |
| | enumeration | GeneralToilet |

source

```
<simpleType name="intEquipTypeEnum">
  <restriction base="NMTOKEN">
    <enumeration value="GeneralPlugload"/>
    <enumeration value="Computer"/>
    <enumeration value="Copier"/>
    <enumeration value="Refrigerator"/>
    <enumeration value="Stove"/>
    <enumeration value="Cooktop"/>
    <enumeration value="Fryer"/>
    <enumeration value="Freezer"/>
    <enumeration value="ClothesDryer"/>
    <enumeration value="ClothesWasher"/>
    <enumeration value="Dishwasher"/>
    <enumeration value="TV"/>
    <enumeration value="VCR"/>
    <enumeration value="Microwave"/>
    <enumeration value="Fan"/>
    <enumeration value="AudioEquip"/>
    <enumeration value="MiscEquip"/>
    <enumeration value="Toilet"/>
    <enumeration value="Urinal"/>
    <enumeration value="Shower"/>
    <enumeration value="Sink"/>
    <enumeration value="Pool"/>
    <enumeration value="HotTub"/>
    <enumeration value="Sauna"/>
    <enumeration value="BatteryCharger"/>
    <enumeration value="Furnace"/>
    <enumeration value="IndustrialEquip"/>
    <enumeration value="Printer"/>
    <enumeration value="Vending"/>
    <enumeration value="Elevator"/>
    <enumeration value="Escalator"/>
    <enumeration value="GeneralLaundryEquip"/>
    <enumeration value="GeneralKitchenEquip"/>
    <enumeration value="GeneralMedicalEquip"/>
    <enumeration value="GeneralOfficeEquip"/>
    <enumeration value="GeneralEntertainmentEquip"/>
    <enumeration value="GeneralRefrigerationEquip"/>
    <enumeration value="GeneralFitnessEquip"/>
    <enumeration value="GeneralVideoEquip"/>
    <enumeration value="GeneralToilet"/>
  </restriction>
</simpleType>
```

## simpleType **lengthUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |

| | |
|---|---|
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tLengthValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration Kilometers<br>enumeration Meters<br>enumeration Centimeters<br>enumeration Millimeters<br>enumeration Miles<br>enumeration Yards<br>enumeration Feet<br>enumeration Inches |
| source | see below |

```
<simpleType name="lengthUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Kilometers"/>
    <enumeration value="Meters"/>
    <enumeration value="Centimeters"/>
    <enumeration value="Millimeters"/>
    <enumeration value="Miles"/>
    <enumeration value="Yards"/>
    <enumeration value="Feet"/>
    <enumeration value="Inches"/>
  </restriction>
</simpleType>
```

## simpleType **monetaryUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tMonetaryValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration EUR<br>enumeration SEK<br>enumeration CZK<br>enumeration DKK<br>enumeration NOK<br>enumeration CHF<br>enumeration HRK |
| source | see below |

```
<simpleType name="monetaryUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="EUR"/>
    <enumeration value="SEK"/>
    <enumeration value="CZK"/>
    <enumeration value="DKK"/>
    <enumeration value="NOK"/>
    <enumeration value="CHF"/>
    <enumeration value="HRK"/>
  </restriction>
</simpleType>
```

## simpleType **moneyPerEnergyUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tMoneyPerEnergyValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration deviceUsed<br>enumeration spaceEnvironmentChange<br>enumeration externalEnvironmentChanged<br>enumeration HVACChange |

| | | | |
|---|---|---|---|
| | enumeration | LightingChange | |
| | enumeration | SpaceOccupancy | |
| | enumeration | SpaceMovement | |
| | enumeration | SpaceOccupantPossitions | |
| | enumeration | Current | |
| | enumeration | Voltage | |
| | enumeration | Energy | |

source

```
<simpleType name="moneyPerEnergyUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="deviceUsed"/>
    <enumeration value="spaceEnvironmentChange"/>
    <enumeration value="externalEnvironmentChanged"/>
    <enumeration value="HVACChange"/>
    <enumeration value="LightingChange"/>
    <enumeration value="SpaceOccupancy"/>
    <enumeration value="SpaceMovement"/>
    <enumeration value="SpaceOccupantPossitions"/>
    <enumeration value="Current"/>
    <enumeration value="Voltage"/>
    <enumeration value="Energy"/>
  </restriction>
</simpleType>
```

## simpleType **noiseUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tNoiseValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration dB |

source

```
<simpleType name="noiseUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="dB"/>
  </restriction>
</simpleType>
```

## simpleType **peopleHeatGainUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tPeopleHeatGainValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration WattPerPerson<br>enumeration BtuPerHourPerson<br>enumeration WattPerSquareMeter<br>enumeration Met |

source

```
<simpleType name="peopleHeatGainUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="WattPerPerson"/>
    <enumeration value="BtuPerHourPerson"/>
    <enumeration value="WattPerSquareMeter"/>
    <enumeration value="Met"/>
  </restriction>
</simpleType>
```

## simpleType **peopleNumberUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tPeopleNumberValue/unit** |
| facets | Kind / Value / Annotation |

| | | |
|---|---|---|
| | enumeration | NumberOfPeople |
| | enumeration | SquareFtPerPerson |
| | enumeration | SquareMPerPerson |

source

```
<simpleType name="peopleNumberUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="NumberOfPeople"/>
    <enumeration value="SquareFtPerPerson"/>
    <enumeration value="SquareMPerPerson"/>
  </restriction>
</simpleType>
```

simpleType **PMVUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | -3 | |
| enumeration | -2 | |
| enumeration | -1 | |
| enumeration | 0 | |
| enumeration | 1 | |
| enumeration | 2 | |
| enumeration | 3 | |

source

```
<simpleType name="PMVUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="-3"/>
    <enumeration value="-2"/>
    <enumeration value="-1"/>
    <enumeration value="0"/>
    <enumeration value="1"/>
    <enumeration value="2"/>
    <enumeration value="3"/>
  </restriction>
</simpleType>
```

simpleType **resourceTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | Electricity | |
| enumeration | NaturalGas | |
| enumeration | Propane | |
| enumeration | FuelOil1 | |
| enumeration | FuelOil2 | |
| enumeration | FuelOil4 | |
| enumeration | Water | |
| enumeration | ChilledWater | |
| enumeration | HotWater | |
| enumeration | Steam | |
| enumeration | Occupancy | |
| enumeration | ActiveEnergy | |
| enumeration | Voltage | |
| enumeration | Current | |
| enumeration | Temperature | |
| enumeration | RelativeHumidity | |
| enumeration | CO2 | |

source

```
<simpleType name="resourceTypeEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Electricity"/>
    <enumeration value="NaturalGas"/>
    <enumeration value="Propane"/>
    <enumeration value="FuelOil1"/>
    <enumeration value="FuelOil2"/>
    <enumeration value="FuelOil4"/>
    <enumeration value="Water"/>
    <enumeration value="ChilledWater"/>
    <enumeration value="HotWater"/>
    <enumeration value="Steam"/>
    <enumeration value="Occupancy"/>
    <enumeration value="ActiveEnergy"/>
    <enumeration value="Voltage"/>
    <enumeration value="Current"/>
    <enumeration value="Temperature"/>
    <enumeration value="RelativeHumidity"/>
    <enumeration value="CO2"/>
  </restriction>
</simpleType>
```

## simpleType **seasonUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tSeasonValue/season** |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | Summer | |
| enumeration | Autumn | |
| enumeration | Winter | |
| enumeration | Spring | |

source

```
<simpleType name="seasonUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Summer"/>
    <enumeration value="Autumn"/>
    <enumeration value="Winter"/>
    <enumeration value="Spring"/>
  </restriction>
</simpleType>
```

## simpleType **spaceTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | ActiveStorage | |
| enumeration | ActiveStorageHospitalOrHealthcare | |
| enumeration | AirOrTrainOrBusBaggageArea | |
| enumeration | AirportConcourse | |
| enumeration | AtriumEachAdditionalFloor | |
| enumeration | AtriumFirstThreeFloors | |
| enumeration | AudienceOrSeatingAreaPenitentiary | |
| enumeration | AudienceOrSeatingAreaExerciseCenter | |
| enumeration | AudienceOrSeatingAreaGymnasium | |
| enumeration | AudienceOrSeatingAreaSportsArena | |
| enumeration | AudienceOrSeatingAreaConventionCenter | |
| enumeration | AudienceOrSeatingAreaMotionPictureTheatre | |
| enumeration | AudienceOrSeatingAreaPerformingArtsTheatre | |

| | | |
|---|---|---|
| | enumeration | AudienceOrSeatingAreaReligious |
| | enumeration | AudienceOrSeatingAreaPoliceOrFireStations |
| | enumeration | AudienceOrSeatingAreaCourtHouse |
| | enumeration | AudienceOrSeatingAreaAuditorium |
| | enumeration | BankCustomerArea |
| | enumeration | BankingActivityAreaOffice |
| | enumeration | BarberAndBeautyParlor |
| | enumeration | CardFileAndCataloguingLibrary |
| | enumeration | ClassroomOrLectureOrTrainingPenitentiary |
| | enumeration | ClassroomOrLectureOrTraining |
| | enumeration | ConfinementCellsPenitentiary |
| | enumeration | ConfinementCellsCourtHouse |
| | enumeration | ConferenceMeetingOrMultipurpose |
| | enumeration | CorridorOrTransition |
| | enumeration | CorridorOrTransitionManufacturingFacility |
| | enumeration | CorridorsWithPatientWaitingExamHospitalOrHealthcare |
| | enumeration | CourtSportsAreaSportsArena |
| | enumeration | CourtroomCourtHouse |
| | enumeration | DepartmentStoreSalesAreaRetail |
| | enumeration | DetailedManufacturingFacility |
| | enumeration | DiningArea |
| | enumeration | DiningAreaHotel |
| | enumeration | DiningAreaFamilyDining |
| | enumeration | DiningAreaLoungeOrLeisureDining |
| | enumeration | DiningAreaMotel |
| | enumeration | DiningAreaTransportation |
| | enumeration | DiningAreaPenitentiary |
| | enumeration | DiningAreaCivilServices |
| | enumeration | DormitoryBedroom |
| | enumeration | DormitoryStudyHall |
| | enumeration | DressingOrLockerOrFittingRoomGymnasium |
| | enumeration | DressingOrLockerOrFittingRoomCourtHouse |
| | enumeration | DressingOrLockerOrFittingRoomPerformingArtsTheatre |
| | enumeration | DressingOrLockerOrFittingRoomAuditorium |
| | enumeration | DressingOrLockerOrFittingRoomExerciseCenter |
| | enumeration | ElectricalOrMechanical |
| | enumeration | ElevatorLobbies |
| | enumeration | EmergencyHospitalOrHealthcare |
| | enumeration | EquipmentRoomManufacturingFacility |
| | enumeration | ExamOrTreatmentHospitalOrHealthcare |
| | enumeration | ExcerciseAreaExerciseCenter |
| | enumeration | ExcerciseAreaGymnasium |
| | enumeration | ExhibitSpaceConventionCenter |
| | enumeration | FellowshipHallReligiousBuildings |
| | enumeration | FineMaterialWarehouse |
| | enumeration | FineMerchandiseSalesAreaRetail |
| | enumeration | FireStationEngineRoomPoliceOrFireStation |
| | enumeration | FoodPreparation |
| | enumeration | GarageServiceOrRepairAutomotiveFacility |
| | enumeration | GeneralHighBayManufacturingFacility |
| | enumeration | GeneralLowBayManufacturingFacility |

| | | |
|---|---|---|
| | enumeration | GeneralExhibitionMuseum |
| | enumeration | HospitalNurseryHospitalOrHealthcare |
| | enumeration | HospitalOrMedicalSuppliesHospitalOrHealthcare |
| | enumeration | HospitalOrRadiologyHospitalOrHealthcare |
| | enumeration | HotelOrConferenceCenterConferenceOrMeeting |
| | enumeration | InactiveStorage |
| | enumeration | JudgesChambersCourtHouse |
| | enumeration | LaboratoryOffice |
| | enumeration | LaundryIroningAndSorting |
| | enumeration | LaundryWashingHospitalOrHealthcare |
| | enumeration | LibraryAudioVisualLibraryAudioVisual |
| | enumeration | LivingQuartersDormitory |
| | enumeration | LivingQuartersMotel |
| | enumeration | LivingQuartersHotel |
| | enumeration | Lobby |
| | enumeration | LobbyReligiousBuildings |
| | enumeration | LobbyMotionPictureTheatre |
| | enumeration | LobbyAuditorium |
| | enumeration | LobbyPerformingArtsTheatre |
| | enumeration | LobbyPostOffice |
| | enumeration | LobbyHotel |
| | enumeration | LoungeOrRecreation |
| | enumeration | MallConcourseSalesAreaRetail |
| | enumeration | MassMerchandisingSalesAreaRetail |
| | enumeration | MediumOrBulkyMaterialWarehouse |
| | enumeration | MerchandisingSalesAreaRetail |
| | enumeration | MuseumAndGalleryStorage |
| | enumeration | NurseStationHospitalOrHealthcare |
| | enumeration | OfficeEnclosed |
| | enumeration | OfficeOpenPlan |
| | enumeration | OfficeCommonActivityAreasInactiveStorage |
| | enumeration | OperatingRoomHospitalOrHealthcare |
| | enumeration | OtherTelevisedPlayingAreaSportsArena |
| | enumeration | ParkingAreaAttendantOnlyParkingGarage |
| | enumeration | ParkingAreaPedestrianParkingGarage |
| | enumeration | PatientRoomHospitalOrHealthcare |
| | enumeration | PersonalServicesSalesAreaRetail |
| | enumeration | PharmacyHospitalOrHealthcare |
| | enumeration | PhysicalTherapyHospitalOrHealthcare |
| | enumeration | PlayingAreaGymnasium |
| | enumeration | Plenum |
| | enumeration | PoliceStationLaboratoryPoliceOrFireStations |
| | enumeration | PublicAndStaffLoungeHospitalOrHealthcare |
| | enumeration | ReadingAreaLibrary |
| | enumeration | ReceptionOrWaitingTransportation |
| | enumeration | ReceptionOrWaitingMotel |
| | enumeration | ReceptionOrWaitingHotel |
| | enumeration | RecoveryHospitalOrHealthcare |
| | enumeration | RestorationMuseum |
| | enumeration | Restrooms |
| | enumeration | RingSportsAreaSportsArena |

| | | |
|---|---|---|
| | enumeration | ServerRoom |
| | enumeration | SleepingQuartersPoliceOrFireStation |
| | enumeration | SortingAreaPostOffice |
| | enumeration | SpecialtyStoreSalesAreaRetail |
| | enumeration | StacksLibrary |
| | enumeration | StairsInactive |
| | enumeration | Stairway |
| | enumeration | SupermarketSalesAreaRetail |
| | enumeration | TerminalTicketCounterTransportation |
| | enumeration | WorkshopWorkshop |
| | enumeration | WorshipPulpitChoirReligious |

source


```
<simpleType name="spaceTypeEnum">
 <restriction base="NMTOKEN">
  <enumeration value="ActiveStorage"/>
  <enumeration value="ActiveStorageHospitalOrHealthcare"/>
  <enumeration value="AirOrTrainOrBusBaggageArea"/>
  <enumeration value="AirportConcourse"/>
  <enumeration value="AtriumEachAdditionalFloor"/>
  <enumeration value="AtriumFirstThreeFloors"/>
  <enumeration value="AudienceOrSeatingAreaPenitentiary"/>
  <enumeration value="AudienceOrSeatingAreaExerciseCenter"/>
  <enumeration value="AudienceOrSeatingAreaGymnasium"/>
  <enumeration value="AudienceOrSeatingAreaSportsArena"/>
  <enumeration value="AudienceOrSeatingAreaConventionCenter"/>
  <enumeration value="AudienceOrSeatingAreaMotionPictureTheatre"/>
  <enumeration value="AudienceOrSeatingAreaPerformingArtsTheatre"/>
  <enumeration value="AudienceOrSeatingAreaReligious"/>
  <enumeration value="AudienceOrSeatingAreaPoliceOrFireStations"/>
  <enumeration value="AudienceOrSeatingAreaCourtHouse"/>
  <enumeration value="AudienceOrSeatingAreaAuditorium"/>
  <enumeration value="BankCustomerArea"/>
  <enumeration value="BankingActivityAreaOffice"/>
  <enumeration value="BarberAndBeautyParlor"/>
  <enumeration value="CardFileAndCataloguingLibrary"/>
  <enumeration value="ClassroomOrLectureOrTrainingPenitentiary"/>
  <enumeration value="ClassroomOrLectureOrTraining"/>
  <enumeration value="ConfinementCellsPenitentiary"/>
  <enumeration value="ConfinementCellsCourtHouse"/>
  <enumeration value="ConferenceMeetingOrMultipurpose"/>
  <enumeration value="CorridorOrTransition"/>
  <enumeration value="CorridorOrTransitionManufacturingFacility"/>
  <enumeration
value="CorridorsWithPatientWaitingExamHospitalOrHealthcare"/>
  <enumeration value="CourtSportsAreaSportsArena"/>
  <enumeration value="CourtroomCourtHouse"/>
  <enumeration value="DepartmentStoreSalesAreaRetail"/>
  <enumeration value="DetailedManufacturingFacility"/>
  <enumeration value="DiningArea"/>
  <enumeration value="DiningAreaHotel"/>
  <enumeration value="DiningAreaFamilyDining"/>
  <enumeration value="DiningAreaLoungeOrLeisureDining"/>
  <enumeration value="DiningAreaMotel"/>
  <enumeration value="DiningAreaTransportation"/>
  <enumeration value="DiningAreaPenitentiary"/>
  <enumeration value="DiningAreaCivilServices"/>
  <enumeration value="DormitoryBedroom"/>
  <enumeration value="DormitoryStudyHall"/>
  <enumeration value="DressingOrLockerOrFittingRoomGymnasium"/>
  <enumeration value="DressingOrLockerOrFittingRoomCourtHouse"/>
  <enumeration
```

```
value="DressingOrLockerOrFittingRoomPerformingArtsTheatre"/>
    <enumeration value="DressingOrLockerOrFittingRoomAuditorium"/>
    <enumeration value="DressingOrLockerOrFittingRoomExerciseCenter"/>
    <enumeration value="ElectricalOrMechanical"/>
    <enumeration value="ElevatorLobbies"/>
    <enumeration value="EmergencyHospitalOrHealthcare"/>
    <enumeration value="EquipmentRoomManufacturingFacility"/>
    <enumeration value="ExamOrTreatmentHospitalOrHealthcare"/>
    <enumeration value="ExcerciseAreaExerciseCenter"/>
    <enumeration value="ExcerciseAreaGymnasium"/>
    <enumeration value="ExhibitSpaceConventionCenter"/>
    <enumeration value="FellowshipHallReligiousBuildings"/>
    <enumeration value="FineMaterialWarehouse"/>
    <enumeration value="FineMerchandiseSalesAreaRetail"/>
    <enumeration value="FireStationEngineRoomPoliceOrFireStation"/>
    <enumeration value="FoodPreparation"/>
    <enumeration value="GarageServiceOrRepairAutomotiveFacility"/>
    <enumeration value="GeneralHighBayManufacturingFacility"/>
    <enumeration value="GeneralLowBayManufacturingFacility"/>
    <enumeration value="GeneralExhibitionMuseum"/>
    <enumeration value="HospitalNurseryHospitalOrHealthcare"/>
    <enumeration value="HospitalOrMedicalSuppliesHospitalOrHealthcare"/>
    <enumeration value="HospitalOrRadiologyHospitalOrHealthcare"/>
    <enumeration value="HotelOrConferenceCenterConferenceOrMeeting"/>
    <enumeration value="InactiveStorage"/>
    <enumeration value="JudgesChambersCourtHouse"/>
    <enumeration value="LaboratoryOffice"/>
    <enumeration value="LaundryIroningAndSorting"/>
    <enumeration value="LaundryWashingHospitalOrHealthcare"/>
    <enumeration value="LibraryAudioVisualLibraryAudioVisual"/>
    <enumeration value="LivingQuartersDormitory"/>
    <enumeration value="LivingQuartersMotel"/>
    <enumeration value="LivingQuartersHotel"/>
    <enumeration value="Lobby"/>
    <enumeration value="LobbyReligiousBuildings"/>
    <enumeration value="LobbyMotionPictureTheatre"/>
    <enumeration value="LobbyAuditorium"/>
    <enumeration value="LobbyPerformingArtsTheatre"/>
    <enumeration value="LobbyPostOffice"/>
    <enumeration value="LobbyHotel"/>
    <enumeration value="LoungeOrRecreation"/>
    <enumeration value="MallConcourseSalesAreaRetail"/>
    <enumeration value="MassMerchandisingSalesAreaRetail"/>
    <enumeration value="MediumOrBulkyMaterialWarehouse"/>
    <enumeration value="MerchandisingSalesAreaRetail"/>
    <enumeration value="MuseumAndGalleryStorage"/>
    <enumeration value="NurseStationHospitalOrHealthcare"/>
    <enumeration value="OfficeEnclosed"/>
    <enumeration value="OfficeOpenPlan"/>
    <enumeration value="OfficeCommonActivityAreasInactiveStorage"/>
    <enumeration value="OperatingRoomHospitalOrHealthcare"/>
    <enumeration value="OtherTelevisedPlayingAreaSportsArena"/>
    <enumeration value="ParkingAreaAttendantOnlyParkingGarage"/>
    <enumeration value="ParkingAreaPedestrianParkingGarage"/>
    <enumeration value="PatientRoomHospitalOrHealthcare"/>
    <enumeration value="PersonalServicesSalesAreaRetail"/>
    <enumeration value="PharmacyHospitalOrHealthcare"/>
    <enumeration value="PhysicalTherapyHospitalOrHealthcare"/>
    <enumeration value="PlayingAreaGymnasium"/>
    <enumeration value="Plenum"/>
    <enumeration value="PoliceStationLaboratoryPoliceOrFireStations"/>
    <enumeration value="PublicAndStaffLoungeHospitalOrHealthcare"/>
```

| | |
|---|---|
| | ```<enumeration value="ReadingAreaLibrary"/>``` |

```
    <enumeration value="ReadingAreaLibrary"/>
    <enumeration value="ReceptionOrWaitingTransportation"/>
    <enumeration value="ReceptionOrWaitingMotel"/>
    <enumeration value="ReceptionOrWaitingHotel"/>
    <enumeration value="RecoveryHospitalOrHealthcare"/>
    <enumeration value="RestorationMuseum"/>
    <enumeration value="Restrooms"/>
    <enumeration value="RingSportsAreaSportsArena"/>
    <enumeration value="ServerRoom"/>
    <enumeration value="SleepingQuartersPoliceOrFireStation"/>
    <enumeration value="SortingAreaPostOffice"/>
    <enumeration value="SpecialtyStoreSalesAreaRetail"/>
    <enumeration value="StacksLibrary"/>
    <enumeration value="StairsInactive"/>
    <enumeration value="Stairway"/>
    <enumeration value="SupermarketSalesAreaRetail"/>
    <enumeration value="TerminalTicketCounterTransportation"/>
    <enumeration value="WorkshopWorkshop"/>
    <enumeration value="WorshipPulpitChoirReligious"/>
  </restriction>
</simpleType>
```

simpleType **stateTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| facets | Kind: enumeration, Value: presence; Kind: enumeration, Value: absence |

| Kind | Value | Annotation |
|---|---|---|
| enumeration | presence | |
| enumeration | absence | |

source

```
<simpleType name="stateTypeEnum">
  <restriction base="NMTOKEN">
    <enumeration value="presence"/>
    <enumeration value="absence"/>
  </restriction>
</simpleType>
```

simpleType **stationIdTypeEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tStationIdValue/unit** |

facets

| Kind | Value | Annotation |
|---|---|---|
| enumeration | WMO | documentation<br>A five-digit station identifier assigned by the World Meteorological Organization (WMO), used for international weather data exchange and station documentation. |
| enumeration | WBAN | documentation<br>Weather-Bureau-Army-Navy station id. A five-digit station identifier used at National Climatic Data Center (NCDC) for digital data storage and general station identification purposes. |
| enumeration | ICAO | documentation<br>A 4-letter indicator for geographical locations throughout the world, managed by the International Civil Aviation Organization (ICAO). |
| enumeration | NWSLI | documentation<br>A 3- to 5-character alpha-numeric identifier assigned by the US National Weather Service (NWS), used for site identification in the NWS communications systems (e.g. AFOS, ROSA, SHEF, etc.). |
| enumeration | FAACallSign | documentation<br>An alpha-numeric identifier of up to 4-characters managed by the USDT Federal Aviation Administration (FAA), used for site identification of airports, weather stations and other sites vital to navigation. |
| enumeration | COOP | documentation<br>A 6-digit number identifying a station in the NWS Cooperative Network. |

source

```
<simpleType name="stationIdTypeEnum">
  <restriction base="NMTOKEN">
```

```xml
    <enumeration value="WMO">
      <annotation>
        <documentation>A five-digit station identifier assigned by the World Meteorological Organization (WMO), used for international weather data exchange and station documentation.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="WBAN">
      <annotation>
        <documentation>Weather-Bureau-Army-Navy station id. A five-digit station identifier used at National Climatic Data Center (NCDC) for digital data storage and general station identification purposes.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="ICAO">
      <annotation>
        <documentation>A 4-letter indicator for geographical locations throughout the world, managed by the International Civil Aviation Organization (ICAO).</documentation>
      </annotation>
    </enumeration>
    <enumeration value="NWSLI">
      <annotation>
        <documentation>A 3- to 5-character alpha-numeric identifier assigned by the US National Weather Service (NWS), used for site identification in the NWS communications systems (e.g. AFOS, ROSA, SHEF, etc.).</documentation>
      </annotation>
    </enumeration>
    <enumeration value="FAACallSign">
      <annotation>
        <documentation>An alpha-numeric identifier of up to 4-characters managed by the USDT Federal Aviation Administration (FAA), used for site identification of airports, weather stations and other sites vital to navigation.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="COOP">
      <annotation>
        <documentation>A 6-digit number identifying a station in the NWS Cooperative Network.</documentation>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>
```

simpleType **temperatureUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tTemperatureValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration F<br>enumeration C<br>enumeration K<br>enumeration R |
| source | see below |

```xml
<simpleType name="temperatureUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="F"/>
    <enumeration value="C"/>
    <enumeration value="K"/>
    <enumeration value="R"/>
  </restriction>
```

```xml
</simpleType>
```

simpleType **timeUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tDurationValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration MilliSeconds<br>enumeration Seconds<br>enumeration Minutes<br>enumeration Hours<br>enumeration Days |

source

```xml
<simpleType name="timeUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="MilliSeconds"/>
    <enumeration value="Seconds"/>
    <enumeration value="Minutes"/>
    <enumeration value="Hours"/>
    <enumeration value="Days"/>
  </restriction>
</simpleType>
```

simpleType **typesOfEventUserProfilingUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| facets | Kind / Value / Annotation<br>enumeration Environmental<br>enumeration ControlActions<br>enumeration Occupancy |

source

```xml
<simpleType name="typesOfEventUserProfilingUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Environmental"/>
    <enumeration value="ControlActions"/>
    <enumeration value="Occupancy"/>
  </restriction>
</simpleType>
```

simpleType **unitlessUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | elements **tRelativeHumidityValue/unit** **tUnitlessValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration Percent<br>enumeration Fraction |

source

```xml
<simpleType name="unitlessUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Percent"/>
    <enumeration value="Fraction"/>
  </restriction>
</simpleType>
```

simpleType **velocityUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tVelocityValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration MetersPerSecond<br>enumeration MilesPerHour |

| | |
|---|---|
| | enumeration Knots |
| source | `<simpleType name="velocityUnitEnum"> <restriction base="NMTOKEN"> <enumeration value="MetersPerSecond"/> <enumeration value="MilesPerHour"/> <enumeration value="Knots"/> </restriction> </simpleType>` |

## simpleType **VoltageUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tVoltageValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration Volts<br>enumeration KiloVolts |
| source | see below |

```
<simpleType name="VoltageUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Volts"/>
    <enumeration value="KiloVolts"/>
  </restriction>
</simpleType>
```

## simpleType **volumeUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tVolumeValue/unit** |
| facets | Kind / Value / Annotation<br>enumeration CubicKilometers<br>enumeration CubicMeters<br>enumeration CubicCentimeters<br>enumeration CubicMillimeters<br>enumeration CubicMiles<br>enumeration CubicYards<br>enumeration CubicFeet<br>enumeration CubicInches |
| source | see below |

```
<simpleType name="volumeUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="CubicKilometers"/>
    <enumeration value="CubicMeters"/>
    <enumeration value="CubicCentimeters"/>
    <enumeration value="CubicMillimeters"/>
    <enumeration value="CubicMiles"/>
    <enumeration value="CubicYards"/>
    <enumeration value="CubicFeet"/>
    <enumeration value="CubicInches"/>
  </restriction>
</simpleType>
```

## simpleType **weekDayUnitEnum**

| | |
|---|---|
| namespace | http://www.inertia.eu/2013/schema/units/ |
| type | restriction of **NMTOKEN** |
| properties | base NMTOKEN |
| used by | element **tDayType/dayInWeek** |
| facets | Kind / Value / Annotation<br>enumeration Monday<br>enumeration Tuesday<br>enumeration Wednesday |

| | | |
|---|---|---|
| | enumeration | Thursday |
| | enumeration | Friday |
| | enumeration | Saturday |
| | enumeration | Sunday |

source

```
<simpleType name="weekDayUnitEnum">
  <restriction base="NMTOKEN">
    <enumeration value="Monday"/>
    <enumeration value="Tuesday"/>
    <enumeration value="Wednesday"/>
    <enumeration value="Thursday"/>
    <enumeration value="Friday"/>
    <enumeration value="Saturday"/>
    <enumeration value="Sunday"/>
  </restriction>
</simpleType>
```