SATISFACTORY

Project Acronym:	**SatisFactory**
Project Full Title:	**A collaborative and augmented-enabled ecosystem for increasing satisfaction and working experience in smart factory environments**
Grant Agreement:	**636302**
Project Duration:	**36 months (01/01/2015 - 31/12/2017)**

DELIVERABLE D1.3

SatisFactory Common Information Data Exchange Model

Deliverable Status:	**Final**
File Name:	**SatisFactory-D1.3-v3.pdf**
Due Date:	**April 2017 (M28)**
Submission Date:	**April 2017 (M28)**
Task Leader:	**CERTH**

Dissemination level	
Public	X
Confidential, only for members of the Consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020
Research and innovation programme under Grant Agreement n°636302

The SatisFactory project consortium is composed of:		
CERTH[1]	Centre for Research and Technology Hellas	Greece
SIGMA[2]	Sigma Orionis SA	France
FIT	Fraunhofer-Gesellschaft zur Foerderung der Angewandten Forschung E.V	Germany
COMAU	Comau SPA	Italy
EPFL	École Polytechnique Fédérale de Lausanne	Switzerland
ISMB	Istituto Superiore Mario Boella sulle tecnologie dell'informazione e delle telecomunicazioni	Italy
ABE	Atlantis Engineering AE	Greece
REGOLA	Regola srl	Italy
SUNLIGHT	Systems Sunlight Industrial & Commercial Company of Defensive, Energy, Electronic and Telecommunication Systems S.A.	Greece
GlassUP	GlassUp srl	Italy
QPLAN	Q-PLAN International Advisors LTD	Greece

Disclaimer

This document reflects only the author's views and the European Union is not liable for any use that may be made of the information contained therein.

[1] Project Coordinator

[2] Terminated beneficiary since June 2016 and replaced by QPLAN

Authors List

Leading Author (Editor)			
Surname	**First Name**	**Beneficiary**	**Contact email**
Krinidis	Stelios	CERTH	krinidis@iti.gr

Co-authors (in alphabetic order)				
#	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1	Georgopoulos	George	ATLANTIS	georgopoulos@abe.gr
2	Kanidis	Stefanos	SUNLIGHT	s.kanidis@sunlight.gr
3	Parcharidis	Symeon	SUNLIGHT	s.parcharidis@sunlight.gr
4	Triantafullou	Dimitra	CERTH	dtriant@iti.gr
5	Tropios	Pantelis	CERTH	ptropios@iti.gr
6	Tsolakis	Apostolos	CERTH	tsolakis@iti.gr
7	Zikos	Stylianos	CERTH	czikos@iti.gr

REVISION CONTROL

Version	Author	Date	Status
0.1	CERTH	May 2015	ToC
0.2	CERTH	June 2015	Initial Draft
0.4	CERTH	July 2015	Draft
0.7	CERTH	July 2015	Quality Check
0.9	CERTH	July 2015	Final Draft reviewed
1.0	**CERTH**	**July 2015**	**Ready for submission to the EC**
1.1	CERTH	April 2016	Draft second version
1.5	CERTH	April 2016	Peer review
1.9	CERTH	April 2016	Quality check
2.0	**CERTH**	**April 2016**	**Ready for submission to the EC**
2.1	CERTH	March 2017	Draft final version
2.6	CERTH	April 2017	Peer review
2.9	CERTH	April 2017	Quality check
3.0	**CERTH**	**April 2017**	**Ready for submission to the EC**

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

LIST OF DEFINITIONS & ABBREVIATIONS

Abbreviation	Definition
3D	Three Dimensional
ADL	Advanced Distributed Learning
API	Application Programming Interface
AR	Augmented Reality
B2MML	Business To Manufacturing Markup Language
CIDEM	Common Information Data Exchange Model
CIM	Common Information Model
COTS	Commercial Off The Shelf
DG RTD	Directorate-General for Research and Innovation
DoW	Description of Work
DSS	Decision Support System
EC	European Commission
ERP	Enterprise Resource Planning
EU	European Union
gbXML	Green Building XML
HMI	Human-Machine Interface
HTML	HyperText Markup Language
I/O	Input/Output
KML	Keyhole Markup Language
LMS	Learning Management System
MESA	Manufacturing Enterprise Solutions Association
OGI	Oil and Gas Interoperability
O&M	Operations and Maintenance

POI	Point of Interest
SCORM	Sharable Content Object Reference Model
WP	Workpackage
XML	EXtensible Markup Language
XSD	XML Schema Definition

EXECUTIVE SUMMARY

In order for the SatisFactory project to fulfil its mission as a collaborative and augmented-enabled ecosystem, 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 SatisFactory components that need it to support their operation.

The presented deliverable represents results of Task T1.4. More specifically, it defines the Common Information Data Exchange Model (CIDEM). The aim of CIDEM 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. knowledge models to support human resources optimization in T2.2). The CIDEM definition is considered as a shared vocabulary that enables to address the information needs for the SatisFactory framework components.

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 standards which could be relevant for SatisFactory concept. The information models from these standards were analysed as a possible basis for CIDEM. The second phase (composed from six steps) reflected the evolution of the SatisFactory architecture. The requirements from SatisFactory components on storage services were the basis for the definition of CIDEM. Overall, the employed methodology followed a component-centric approach and this approach is reflected by this deliverable as well.

In order to reflect evolving SatisFactory architecture, several architecture parts/components’ requirements were specified (in form of interface specifications) and analysed. All components of the architecture have been considered:

- Smart Sensor Network;
- Middleware;
- Integrated DSS;
- Ontology Manager;
- Context-Aware Manager;
- AR In-Factory Platform;
- Operational Platform with Augmented Intelligence;
- Collaborative Tools;
- Gamification Framework;
- Re-Adaptation Toolkit;
- Training & Educational Platform;
- Multi-Modal & Augmented HMIs and AR Devices.

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 CIDEM (represented by the CIDEM component within the architecture).

Subsequently, based on the analysis, elements of CIDEM have been defined. They were defined for all those components that intend to interact with the CIDEM API. The definition of CIDEM has been produced in the form of CIDEM interfaces and XSD schemas (both informal graphical visualisation as well as extensive formal definitions of elements and complex types are provided).

The presented deliverable reflects the current state of SatisFactory CIDEM as it is, which is considered almost final. Although the task devoted to the development of CIDEM has finished, this actual form of CIDEM is not guaranteed to be final, since there are other tasks running which may impact the CIDEM and induce its modifications in next project period. However, since most of the SatisFactory components have been finalized and tested at one of the pilots (CERTH/CPERI) the potential modifications are expected to be limited.

1. INTRODUCTION

The SatisFactory Common Information Data Exchange Model (CIDEM) defines the high level domain model comprising the basic elements (events, relations, interfaces etc.) underlying the SatisFactory collaborative and augmented-enabled ecosystem [1].

CIDEM 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]. CIDEM specification consists of architecture and concepts of CIDEM, language (by which the CIDEM schema is defined), and a method for mapping CIDEM to other information models. The CIDEM architecture is usually object-oriented. The CIDEM elements are typically represented as classes and any relationships between them are represented as CIDEM associations. Inheritance allows specialization of common base elements into more specific derived elements. The CIDEM schema is a conceptual schema which defines specific sets of objects and relationships between them, representing in overall a common base for the managed elements in an IT environment.

The SatisFactory CIDEM is specified by the inputs/outputs interfaces between SatisFactory components and the Common Information Data Exchange Model (CIDEM – serving as repository). This specification is formalized by signatures of CIDEM services (APIs) and XSD schemas defining data types used in services needed by SatisFactory components.

This last version of this deliverable contains the updates of the CIDEM, after the first and second approaches of the implementation of the SatisFactory tools. A large number of interfaces, as well as XSD files and structures have been changed, including changes to the ReAdaptation Events, a NewJob Event has been added, the use of a Logging Schema for AR events, the removal of the Topics list and changes to the R3D and Gamification schemas.

1.1 SCOPE OF THE REPORT

The CIDEM specification is formalized as semantic model (conceptual schema) including all information which is needed by SatisFactory 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.). SatisFactory CIDEM serves as a vocabulary in communication between any components of the SatisFactory framework and the Common Information Data Exchange Model (CIDEM) API. The scope of SatisFactory CIDEM is to provide information and semantic model for the domain objects used be the SatisFactory components. Based on the description of these components from the SatisFactory report D2.1 here is the outline of these objects:

- Shop-Floor information (static data)
    - Building information (geometry, etc.)
    - Configuration files
    - Equipment information
    - Sensors information
    - Assets information

    - Actors information
    - Procedures information
- Events and RAW data (dynamic data)
    - Measurements/ RAW data
    - Alert events
    - Maintenance events
    - Re-Adaptation events
    - AR events
    - Training events
    - Recording events
    - Gesture events
    - Presence events
    - Gear events
    - Logging events
- Social communication data (dynamic data)
    - Social communication events
- Gamification data (dynamic data)
    - Gamification events

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 an assembly line in a given building in a given time). The SatisFactory CIDEM enables to combine such dynamic data and static data. CIDEM process requests from the SatisFactory framework components (the requests are based on CIDEM specification) and store data in the hybrid repository. The response to these requests are serialised (based on CIDEM) and send over the CIDEM exporting services (APIs) to the requesting SatisFactory component.

1.2 STRUCTURE OF THE DELIVERABLE

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

- **Section 2** presents a methodological approach employed to organise activities aiming at the specification of SatisFactory CIDEM. A two-phase methodology based on component-centric approach is introduced.
- **Section 3** identifies analyses relevant industrial standards, focusing on standards related to Common Information Models in factories.
- **Section 4** provides analysis of CIDEM requirements.
- **Section 5** specifies elements of CIDEM. The definitions have the form of CIDEM interfaces and formal XSD schemas.
- **Annex I** presents detailed extensive definitions of XSD elements and complex types. The definitions have a formal character.

2. Methodology

The SatisFactory 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 CIDEM was to:

- enable to translate information in heterogeneous formats into a format understandable to all SatisFactory components;
- describe information sources using the vocabulary that is used by all the SatisFactory components;
- 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 CIDEM and ensuring that the above mentioned points are properly reflected at the design of CIDEM. At first, activities were divided into two phases:

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

  In the first phase of T1.4, 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. It aims to identify 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.

- **Phase 2**: SatisFactory requirements analysis

  In the second phase of T1.4 the activities were aimed at the evolving SatisFactory 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 CIDEM – the components that need to interact with the central data repository represented by the CIDEM component. The employed methodology is therefore based on component centric approach (in opposition to sometimes used data centric approach).

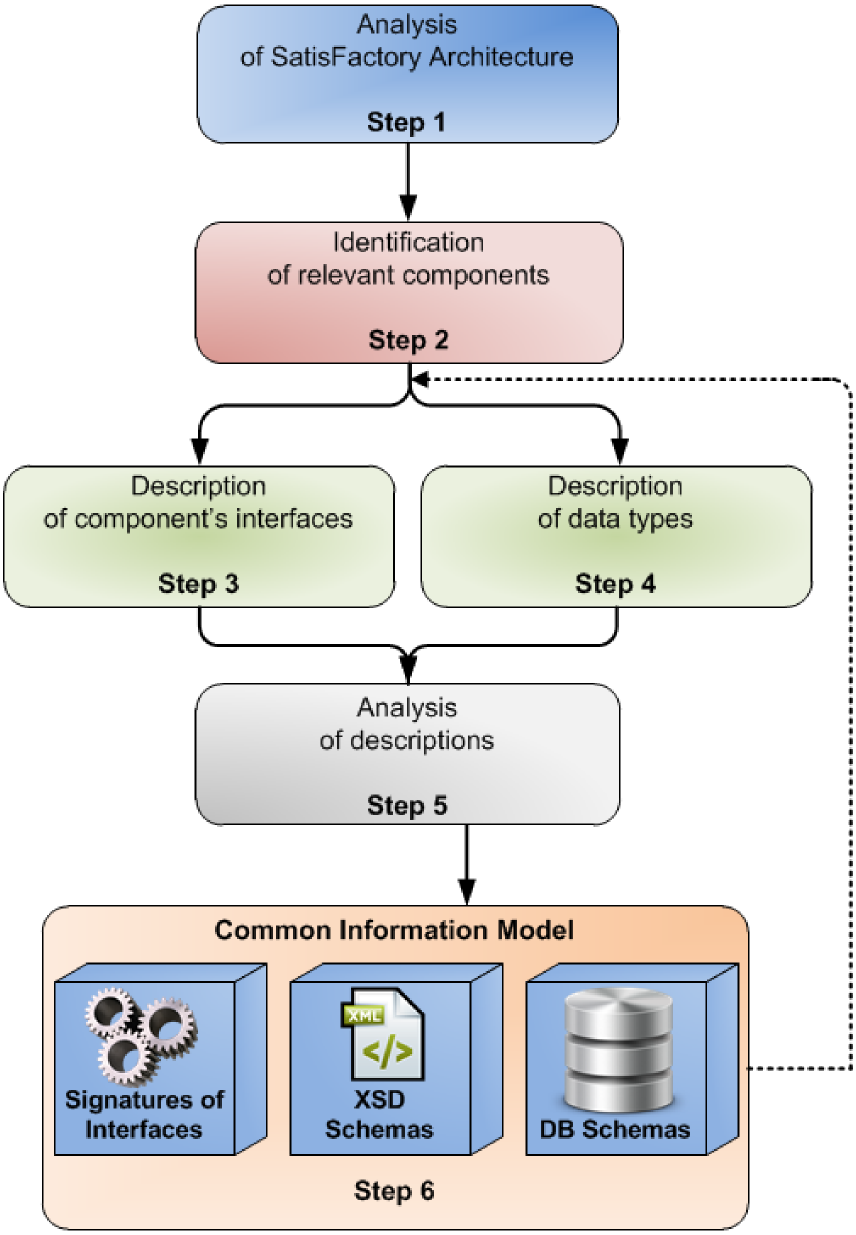


Figure 1: Steps of the methodology adopted during SatisFactory requirement analysis (Phase 2)

The methodology adopted during Phase 2 (Figure 1), is divided into six steps, which are descripted below:

- **Step 1**: Analysis of SatisFactory architecture

  The main goal of these activities was to clearly define the role of the SatisFactory CIDEM in the architecture as a common shared vocabulary. The vocabulary enables to access at data (both in read and write directions) in a unified way. The data access service is based on the vocabulary – the service itself is provided by the Common Information Data Exchange Model (CIDEM) API.

- **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 SatisFactory components that could benefit from using the CIDEM API to store in and retrieve data from. The data flow was investigated according to SatisFactory architecture. The description of SatisFactory components contains part about dependencies with other components and brief description of component I/O interfaces. 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 CIDEM, 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 CIDEM. 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 SatisFactory 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 4.

Table 2: Specification of SatisFactory 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 CIDEM API, 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 CIDEM, 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 4.

- **Step 6**: Proposal of Common Information Model
  - **Step 6.1**: Signatures of the interfaces

    By processing all the component interfaces of the CIDEM component, their names and signature are defined. These specifications of the CIDEM interfaces enable to read/store data for those SatisFactory 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 5.

  - **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 SatisFactory components. Since all 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 SatisFactory data types. The XSD specifications can be found in Section 5.

    The signatures of the interfaces with the known XSD specification of the data types used in them enable to formally Access any SatisFactory 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 CIDEM.

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

Note, since the SatisFactory activities are not closed yet, it is not the aim of this methodology to cover complete dependencies between stored information in the CIDEM and requested information from CIDEM.

3. Existing Standards Analysis

This section presents some industry standards. These standards were analysed before the requirements from SatisFactory components were specified (i.e. during phase 1 according to adopted methodology). Various standards were chosen in cooperation with the project's consortium, analysed and presented in the previous iterations of this deliverable, with minor modifications in the selected ones during this period.

3.1 Business To Manufacturing Markup Language (B2MML)

B2MML or *Business To Manufacturing Markup Language* is an XML implementation of the ANSI/ISA-95, Enterprise-Control System Integration, family of standards (ISA-95), known internationally as IEC/ISO 62264. B2MML consists of a set of XML schemas written using the World Wide Web Consortium's XML Schema language (XSD) that implement the data models in the ISA-95 standard

B2MML is meant to be a common data definition to link ERP and supply chain management systems with manufacturing systems such as Industrial Control Systems and Manufacturing Execution Systems. B2MML is a complete implementation of ISA-95 and is published by the Manufacturing Enterprise Solutions Association (MESA).

B2MML covers the core package of a CIM. Data types from the domain of manufacturing, namely Assets, Equipment, Actors, Procedures, Measurements, Alerts and Re-Adaptation activities have been used and adapted to the SatisFactory CIDEM.

3.2 MIMOSA

MIMOSA is a non-profit 501(c) industry association, focused on enabling industry solutions leveraging supplier neutral, open standards, to establish an interoperable industrial ecosystem for Commercial Off The Shelf (COTS) solutions components provided by major industry suppliers. In order to accomplish this goal, MIMOSA (working in cooperation with other like-minded groups) has facilitated the development of the Oil and Gas Interoperability (OGI) Solutions Process, which includes the OGI Pilot, the OGI Solutions Architecture and the ISO OGI Technical Specification. Collectively, these elements establish the basis for the OGI Ecosystem, which is a true supplier neutral solutions environment enabling a major paradigm shift towards a solutions process providing lower cost, faster implementations and improved quality.

The OGI Solutions Process is driven by high value added industry use cases, developed, validated and managed by MIMOSA and industry partners. Current use cases span the full life-cycle of major classes of physical assets (plants, platforms and facilities) including true life-cycle management for the "digital asset" which must accurately reflect the physical assets being modeled, monitored and managed. The OGI Solutions Process leverages a portfolio of published international and industry standards and specifications, which are incorporated by reference into the various applicable use cases. Key standards in the portfolio include those associated with the OpenO&M Initiative (ISA 88/95, MIMOSA CCOM, OPC UA, OAGi BOD architecture and OpenO&M ws-ISBM/CIR), as well as ISO 15926. The OGI Solutions Process seeks to avoid “re-inventing wheels” by leveraging a portfolio of existing standards purpose built for specific functions, with a focus on solving the business problems defined by the use cases, rather than on developing new standards. While a core team of owner/operators from the Oil and Gas industry were the initial stakeholders in this process, many of the use cases, standards, specifications and methods are applicable to a much broader cross section of industry sectors sometimes referred to as critical infrastructure. This is reflected in the breadth of industries represented by those contributing to or observing the OGI Pilot.

The OGI Pilot provides an industrial scale environment for use case development and improvement as well as establishing the proving grounds for interoperability within the OGI Ecosystem, which it defines based on the OGI Solutions Architecture. The OGI Pilot uses engineering data sets developed and managed by established industry EPC firms to be representative of the data sets required for real capital projects. These data sets are used as the basis for a “Continuous Handover”, where topological, schematic and parametric data sets are managed through simulated capital projects, then shared, exchanged and handed over at appropriate times (defined by the use cases) in machine readable, supplier neutral formats based on the portfolio of included standards. This “Digital Asset” is then used to directly provision the major classes of Operations and Maintenance (O&M) systems in a synchronized fashion, establishing the basis for the O&M systems to participate in defined O&M use cases. Collectively, the set of use cases and the portfolio of standards and specifications which they leverage, defines the basis for an “Industry Foundation Architecture”, which we now define as the OGI Solutions Architecture, upon which owner/operator specific business processes can be established through standardized methods for orchestration and governance.

In general, enterprises that are critically dependent upon complex physical assets have historically focused integration efforts on two major horizontal layers; Real-Time Control and Business Information Systems. Experts within these two areas seldom work directly with each other and do not focus on integration between the layers, which has resulted in a significant vertical information gap. This gap is compounded when O&M processes, systems and people are not efficiently integrated with each other, resulting in a corresponding horizontal information gap. Together, these gaps create an empty space in the very center of enterprise process and information integration.

In the past, operational inefficiencies coming from the lack of integration have been overlooked or minimized due to a general lack of interdisciplinary understanding. Overall optimization requires proper integration of O&M processes, systems and people. MIMOSA is working on effective solutions to eliminate these impediments to efficiency.

Historically, the O&M community has also lacked tight alignment with the Life-cycle Engineering community. This has led to a series of poorly connected activities with highly suboptimal results including data quality problems and the loss of configuration control for complex physical assets including plants, platforms and facilities. The effect has been a loss of integrity management for the digital asset which makes integrity management for the physical asset much more difficult. Working in close cooperation with groups such as POSC Caesar Association and Fiatech, MIMOSA is helping to establish the basis for a more integrated approach to Critical Infrastructure Management, holistically combining full life-cycle engineering with O&M activities. This supports more sustainable approaches to both integrity management and risk management.

MIMOSA covers the maintenance package of a CIM. However, B2MML has been selected to cover this package. Both, standards have a lot of similarities, and B2MML has a more flexible structure, making it more attractive and suitable to SatisFactory needs.

3.3 SCORM

Sharable Content Object Reference Model (SCORM) is a collection of standards and specifications for web-based electronic educational technology. It defines communications between client side content and a host system (called "the run-time environment"), which is commonly supported by a training management system. SCORM also defines how content may be packaged into a transferable format.

SCORM is a specification of the Advanced Distributed Learning (ADL) Initiative from the Office of the United States Secretary of Defense.

SCORM 2004 introduced a complex idea called sequencing, which is a set of rules that specifies the order in which a learner may experience content objects. In simple terms, they constrain a learner to a fixed set of paths through the training material, permit the learner to "bookmark" their progress when taking breaks, and assure the acceptability of test scores achieved by the learner. The standard uses XML, and it is based on the results of work done by AICC, IMS Global, IEEE, and Ariadne.

SCORM is the de facto industry standard for e-learning interoperability. Specifically, SCORM governs how online learning content and Learning Management Systems (LMSs) communicate with each other. SCORM does not speak to instructional design or any other pedagogical concern, it is purely a technical standard.

SCORM covers the package of the model which is related to the training activities within SatisFactory project. Data types from the domain of training have been used and adapted to the SatisFactory CIDEM.

3.4 GBXML

The *Green Building XML (gbXML)* schema, referred to as "gbXML", was developed to facilitate the transfer of building information stored in CAD building information models, enabling integrated interoperability between building design models and a wide variety of engineering analysis tools and models available today. Today, gbXML has the industry support and wide adoption by the leading CAD vendors, Autodesk, Graphisoft, and Bentley. With the development of export and import capabilities in several major engineering modeling tools, gbXML has become a defacto industry standard schema. Its use dramatically streamlines the transfer of building information to and from engineering models, eliminating the need for time consuming plan take-offs. This removes a significant cost barrier to designing resource efficient buildings and specifying associated equipment. It enables building design teams to truly collaborate and realized the potential benefits of Building Information Modeling.

In June of 2000, the gbXML schema was submitted for inclusion in aecXML(TM), the industry-led initiative, launched by Bentley Systems with much excitement in the summer of 1999. Shortly thereafter, gbXML became the draft schema for the Building Performance & Analysis Working Group.

XML, extensible markup language, is a type of computer language that allows software programs to communicate information with little to no human interaction. This approach allows building designers to focus on what they want to do most - design beautiful, environmentally responsible buildings that use intelligent technologies to meet their client's needs at the lowest cost possible. Helping realize the promise of Building Information Modeling, gbXML allows intelligent solutions for the design, certification, operation, maintenance, and recycling of buildings. The possibilities are limited only by the collective imagination of the building design community.

gbXML covers the package of a CIM which is related to the building (pilot area) description within SatisFactory project. Data types from the domain of building have been used and adapted to the SatisFactory CIDEM.

3.5 OPENSOCIAL

OpenSocial is a public specification that defines a set of APIs for social applications that run on the web. OpenSocial's goal is to make more apps available to more users, by providing a common API that can be used in many different contexts. Developers can create applications, using standard JavaScript and HTML, that run on social websites that have implemented the OpenSocial APIs. These websites, known as OpenSocial containers, allow developers to access their social information; in return they receive a large suite of applications for their users.

The OpenSocial APIs expose methods for accessing information about people, their friends, and their data, within the context of a container. This means that when running an application on Orkut, the user is interacting with Orkut friends, while running the same application on MySpace lets user interact with user’s MySpace friends.

OpenSocial covers the package of the model which is related to the social communication activities within SatisFactory project. Data types from the domain of social platforms have been used and adapted to the SatisFactory CIDEM.

4. Analysis of CIDEM Requirements

Analysis of CIDEM requirements is tightly connected with the requirements to the Common Information Data Exchange Model (CIDEM) API, since CIDEM API provides the information backend for the interface. As it is described in D2.1 [3], all SatisFactory Components communicate among each other through the CIDEM API when it comes to retrieving historical data. Other modules that request information from the CIDEM or need to store information in the CIDEM will interact with this module. The CIDEM API will provide functionality for data access and management (import, export, search, access, etc.).

In following sections we try to identify possible data structures flowing between these components and the CIDEM. It was decided that the data structure specification in CIDEM will be based on XML schemas. One of the reasons for it is that XML schema is a (industry) standard for data structure specification in Middleware. The second reason is that the semantics of elements from these XML schemas (structural level of data) can be then quite straightforward defined in the ontologies (semantic level of data).

The design of CIDEM was mainly determined by the information required by the SatisFactory components that can be generated only as a combination (and/or modification) of particular information models.

The SatisFactory components requesting information from the CIDEM using the web services of the CIDEM API to retrieve or to save information to the CIDEM storage. All the web services will use a composition of different CIDEM elements for information exchange. The scenarios that incorporate CIDEM are described in Section 5.

Note that CIDEM will contain also original information in raw form that can be accessed through the references in CIDEM elements to them. Therefore these data could be also accessible by SatisFactory components. The elements described below are considered as shared vocabulary within the SatisFactory system.

4.1 Information Model

The shop floor information model has been imported into the CIDEM at the beginning of the pilots. The full shop-floor information representation of the factory is also directly stored in XML, so it can be used for visualisation or exporting.

The SatisFactory information model is comprised by a number of different Shop floor components. Every Shop floor contains the following components:

- Building Information Model
- Configuration Files
- Equipment List
- Sensor/Devices List
- Assets List
- Actors List
- Procedures List

Each of these elements of the SatisFactory Information Model is analysed below.

4.1.1 ShopFloor

4.1.1.1 ShopFloor Information Model

ShopFloor information model has to be imported into the CIDEM at the beginning of each pilot. It contains all the static information about the geometry of the building (walls, windows, spaces, etc.), which will be used as a common base for all SatisFactory components. The location of all information will be in accordance to this ShopFloor model. In order to cope with SatisFactory needs, a well-known open schema has been adopted, the *Green Building XML (gbXML)*. gbXML format is supported by almost all design tools, so it could be easily portable.

The basic concept for modeling the shop-floor in SatisFactory is a Space representing any chosen place in the building, usually one enclosed space surrounded by walls. A space contains equipment, assets, sensors etc.

The Shop-Floor model has been implemented in a manner that can be extended in building level, so as to be able to interconnected with other systems such as Energy Efficience and Resource Management, where the multi-purpose aspect of Building Information Model (BIM) is going to very useful for the overall operation of the factory as a factory and as a building as well.

Finally, the ShopFloor Information Model includes information regarding positioning of Forbidden Areas and Smart Assembly Stations within the pilot's geometry.

4.1.1.2 Configuration Files

Certain SatisFactory components are using CIDEM to access common configuration files that can be altered remotely through the CIDEM API without direct access to the physical machine. Configuration files can be either stored as single strings, and be retrieved as such, or in cases where required specific schemas were created so as to be able to store the information needed as XML, making it possible to retrieve partial information based on the available services provided by the CIDEM API.

4.1.1.3 Equipment List

Equipment list is very important information concerning the pilots and the factory operation, which could be inserted to the CIDEM at the beginning of each pilot. It contains the information regarding the equipment that is located in the factory, and more specifically to the spaces/ areas where the use cases are going to be deployed. This information could be used by most of the SatisFactory components (such as maintenance tools and other subcomponents of the integrated Decision Support System (DSS), etc.). In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modeling the equipment in shop-floor is the availability of all the information about the equipment in the shop-floor in general and more specifically in each space.

4.1.1.4 Sensors List

Sensor list contains information regarding the sensors that are already installed or will be installed within SatisFactory project in the shop-floor and provide information about the dynamic behaviour of the shop-floor (e.g. assembly lines, production lines, etc.). This information could be used by most of the SatisFactory components (such as maintenance tools and other subcomponents of the integrated Decision Support System (DSS), Augmented Reality tools, etc.). Although "Equipment List" described above could be used to this end, we preferred a more dedicated solution, which is inherited by Adapt4EE [4] and INERTIA [5] projects. This solution has been applied and efficiently tested in these two projects in real-life environments in their pilot sites.

The basic concept for modeling the sensors in a shop-floor is the availability of the overall information about the sensor network in the shop-floor in general and more specifically in each space.

4.1.1.5 Assets List

Assets list is very important information concerning the pilots and the factory operation, which could be inserted to the CIDEM at the beginning of each pilot. It contains the information regarding the assets of the factory. In this category, information about assets that could not be categorized in "Equipment List" or belong to the general category of assets. This information could be used by most of the SatisFactory components (such as maintenance tools and other subcomponents of the integrated Decision Support System (DSS), Augmented Reality tools, etc.). In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modeling the asset in shop-floor is the availability of useful information about the shop-floor and its assets.

4.1.1.6 Actors List

Actors list contains information regarding the actors that are involved in the use cases within SatisFactory project in the shop-floor. This information could be used by most of the SatisFactory components (such as maintenance tools and other subcomponents of the integrated Decision Support System (DSS), Augmented Reality tools, etc.) in conjunction with the Procedures that are described below. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modeling the actors in a shop-floor is the availability of the information about the actors that work in each space/ location and in general in the shop-floor, and of course the kind of the job that each actor performs. This information will be available in conjunction with the Procedures/ Activities running in each shop-floor.

4.1.1.7 Procedures List

Procedures list contains information regarding the procedures/ activities that are occurring during the use cases within SatisFactory project in the shop-floor. This information could be used by most of the SatisFactory components, especially those that are related to the assembly and training activities (e.g. Augmented Reality tools, etc.), in conjunction with the Actors that are described above. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modeling the procedures in a shop-floor is the availability of the information about the procedures/ activities that performed in each space/ location and in general in the shop-floor, and of course the kind of the job that each actor performs. This information will be available in conjunction with the Actors employed in each shop-floor.

4.1.2 Events

All the events and measurements extracted in a shop-floor have to be imported into the CIDEM dynamically during the pilot tests through the middleware component. They contain all the information about the dynamic behaviour and status of the shop-floor. This information is mainly collected by the sensors and systems installed in the shop-floor (including the already installed sensors and systems). Furthermore, information extracted by the SatisFactory components (e.g. maintenance events, incident events, etc.) will be stored to the CIDEM as well. This information is going to be directly stored in in the CIDEM in XML format through the CIDEM API, so it could be later used either for visualization or exporting.

The SatisFactory events could be categorized in a number of classes, which are listed below:

- Measurements;
- Alerts;
- Maintenance events;
- Re-adaptation events;
- Augmented reality events;
- Training events;
- Recording events;
- Gesture events;
- Presence events;
- Gear events;
- AR Logging events;

Each of these event categories of the SatisFactory is analysed below.

4.1.2.1 Measurements

Measurements have to be imported into the CIDEM during the pilot deployment and execution. They contain all the dynamic information regarding the shop-floor and its operation. The installed multi-sensorial network will feed the CIDEM through Middleware with all this information, which is vital for the correct operation of the SatisFactory components and will be utilized by all of them. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modelling the measurements of shop-floor in SatisFactory is that measurements represents the dynamic behaviour of the shop-floor.

4.1.2.2 Alerts

Alterts contain the dynamic information of a shop-floor related to incidents and other abnormal situations, captured by the SatisFactory components. The SatisFactory components will feed the CIDEM with this information, which is vital for the correct operation of the overall shop-floor. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modelling the alerts of shop-floor in SatisFactory is that they represent/ indicate abnormal situations in the workplace, whose timely solutions could eliminate accidents and improve the working environment.

4.1.2.3 Maintenance Events

Maintenance events contain the dynamic information of a shop-floor related to the maintenance issues of the equipment and the assets in the shop-floor. These issues are related with the scheduled maintenance, as well as with the maintenances that should occur immediately after an event (e.g. an incident, etc.). The SatisFactory components will feed the CIDEM with this information, which is vital for the correct operation of the overall shop-floor, since the proper maintenance of the equipment is related to a number of very important issues, such as employees safety, production, etc. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modelling the maintenance events of shop-floor in SatisFactory is that they are vital for the correct operation of the overall shop-floor improving employees' safety, increasing production, etc. Furthermore, proper scheduled maintenance could protect from potential incidents.

4.1.2.4 Re-adaptation Events

Re-adaptation events contain the dynamic information of a shop-floor related to the issues of production lines and reallocation of human resources at them. Information related to the production lines and the human resources allocation will be dynamically analysed by the SatisFactory components, providing a better reallocation of the human resources so as to have the optimum performance and employees satisfaction. The SatisFactory components will feed the CIDEM with this information. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Business To Manufacturing Markup Language (B2MML)*, which is used by many factories and integrated systems in manufacturing environments, as well as fits to SatisFactory needs.

The basic concept for modelling the re-adaptation events of shop-floor in SatisFactory is that the dynamic environment of a shop-floor needs potential re-adaptation of the human resources so as to achieve the optimum employees' satisfaction and production performance.

4.1.2.5 Augmented Reality Events

Augmented Reality (AR) events contain the dynamic information that is related to the AR activities by the SatisFactory components in the shop-floor. All these events should be stored to the CIDEM. In order to cope with SatisFactory needs, a definition developed by REGOLA was adopted, the *R3D*, which is used mainly by REGOLA's augmented reality tools and applications.

The basic concept for modelling the AR events occurred by the SatisFactory components in the shop-floor is the necessity of keeping historic records and the re-usage of AR models and their properties.

4.1.2.6 Training Events

Training events contain the dynamic information of a shop-floor related to the training activities performed by the SatisFactory components at the shop-floor. The information related to these issues will be dynamically stored to the CIDEM. In order to cope with SatisFactory needs, a well-known standard was adopted, the *Sharable Content Object Reference Model (SCORM)*, which is used by a large number of e-learning software products.

The basic concept for modelling the training events of shop-floor in SatisFactory is the dynamic form of the training events, which is depended on the trainer, the time, etc. All this information could provide useful information to the employees and the trainers increasing their productivity and their comprehension at the workflow.

4.1.2.7 Recording Events

Recording events contain the dynamic information of a shop-floor related to the recordings made by the cameras (e.g. depth and thermal cameras) of the Smart Sensor Network at the shop-floor. Information related to these issues will be dynamically stored to the CIDEM. In order to cope with SatisFactory needs, a dedicated customized solution has been selected to describe this information.

The basic concept for modelling the recording events of shop-floor in SatisFactory is the dynamic form of the events related to the cameras (e.g. depth and thermal cameras) installed in the shop-floor. This information could provide useful information about the camera to the modules that are related and use data from these cameras.

4.1.2.8 Gesture Events

Gesture events contain the dynamic information of a shop-floor related to the gestures detected by the respective SatisFactory components at the shop-floor. The information related to these issues will be dynamically stored to the CIDEM. In order to cope with

SatisFactory needs, a dedicated customized solution has been selected to describe this information.

The basic concept for modelling the gesture events in SatisFactory is the dynamic form of the gestures of the humans/employees performing activities that are strongly related to the SatisFactory use cases. All this information could provide useful information to the SatisFactory components that needs to recognize activities, gestures, etc.

4.1.2.9 Presence Events

Presence events contain the dynamic information of a shop-floor related to the presence of humans/employees in the shop-floor. The information related to these issues will be dynamically stored to the CIDEM. In order to cope with SatisFactory needs, a dedicated solution has been selected to describe this information.

The basic concept for modelling the presence events of shop-floor in SatisFactory is the dynamic form of the human/employees presence in specific areas of the shop-floor where SatisFactory use cases will be deployed.

4.1.2.10 Gear Events

Gear events contain the dynamic information related to the items utilized by the employees during the use cases operation. The information related to these issues will be dynamically stored to the CIDEM. In order to cope with SatisFactory needs, a dedicated solution has been selected to describe this information.

The basic concept for modelling the Gear events in SatisFactory is the dynamic form of the status of the items utilized by the employees during each step of the SatisFactory use cases. This information is mandatory to the components related to the use cases (e.g. training components need to know the status of each item utilized within training process).

4.1.2.11 AR Logging Events

AR Logging events contain the logs from the AR procedures and training sessions. These logs contain information on the training sessions performed by each trainee, pertaining to the procedures followed, the time spent on each step, the cumulative time spent on the training session etc. This information can then be used by the training supervisor to perform analysis over time with regard to training performance for each trainee, and to evaluate the training material and scenarios against the results of different categories of trainees.

4.1.3 Social Communication

All the events and information related to the social communication that is going to be extracted and collected in the shop-floor by the corresponding SatisFactory components have to be imported into the CIDEM dynamically during the pilot tests through the middleware component in XML format through CIDEM API. They will contain important information about the communication among employees through the SatisFactory communication platform. In order to cope with SatisFactory needs, a well-known standard was adopted, the *OpenSocial standard*, which is used by a large number of social web working groups and products.

The basic concept for modelling the social communication events and information of the corresponding components of SatisFactory in the shop-floor is the historical records and the issue that all social communication platforms needs to keep records of every event and information is related to the communication activities within the platform.

4.1.4 Gamification

All the events and information related to the gamification activities within SatisFactory projects should be collected and stored dynamically in the CIDEM during the pilot tests through middleware component. The information about the gamification activities during training process as well as during the normal activities of employees, which are going to be enriched with gamification techniques in order to increase the attractiveness of the specific processes and activities, is very important to be stored. In order to cope with SatisFactory needs, a well-known standard was adopted, the *SportsML-G2 standard*, which is used by a large number of products and applications the models sports and gamification processes.

The basic concept for modelling the gamification events and activities of the corresponding components of SatisFactory in the shop-floor is the historical records and the issue that all gamification tools needs to keep records of every event and information is related to the them.

4.2 ARCHITECTURE – CIDEM MAPPING

In this subsection, a brief mapping among CIDEM components and SatisFactory architecture components is presented.

Table 3: Architecture – CIDEM mapping

CIDEM component	Smart Sensor Network	Middleware	Ontology Manager	Context-Aware Manager	AR In-Factory Platform	Integrated DSS	Collaboration Tools	Gamification Framework	Operational Platform with Augmented Intelligence	HR Re-Adaptation Toolkit	Training & Educational Platform	Multi-Modal & Augmented HMIs and AR Devices
ShopFloor Information Model	X	X	X	X	X	X			X	X	X	X
Topic Model		X		X	X	X	X	X	X	X	X	X
Equipment Model		X	X	X		X			X		X	X
Sensor Model	X	X	X	X	X	X		X	X	X	X	X

CIDEM component	Smart Sensor Network	Middleware	Ontology Manager	Context-Aware Manager	AR In-Factory Platform	Integrated DSS	Collaboration Tools	Gamification Framework	Operational Platform with Augmented Intelligence	HR Re-Adaptation Toolkit	Training & Educational Platform	Multi-Modal & Augmented HMIs and AR Devices
Asset Model		X	X	X	X	X		X	X			
Actor Model		X	X	X	X	X	X	X	X	X	X	X
Procedure Model		X	X	X	X	X	X	X	X	X	X	X
Measurements	X	X	X	X	X	X	X	X	X	X	X	X
Alerts		X	X	X	X	X			X	X		X
Maintenance Events	X	X	X	X	X	X			X	X	X	X
Re-Adaptation Events		X	X	X	X	X			X	X		X
Augmented Reality Events		X	X		X			X	X		X	X
Training Events		X	X	X	X		X		X		X	X
Recording Events	X	X		X								X
Gesture Events	X	X		X							X	X
Presence Events	X	X		X								
Gear Events	X	X		X								
AR Logging Events		X			X				X		X	X
Social Communication Model		X	X	X			X	X	X			
Gamification Model		X	X	X				X	X			X

5. CIDEM SPECIFICATIONS

Based on the analysis from Section 4 the elements of CIDEM are defined here. These elements are defined for all SatisFactory components that are interacting with CIDEM through its APIs. The definition of CIDEM for each of such component consists of CIDEM interface name and XSD schemas defining data types used in these interfaces. Thus the structure of CIDEM description for each SatisFactory component looks as following:

- CIDEM interfaces
- XSD schemas

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

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

A high level diagram of the CIDEM structure is illustrated at Figure 2. Each element of the schema is analysed below.

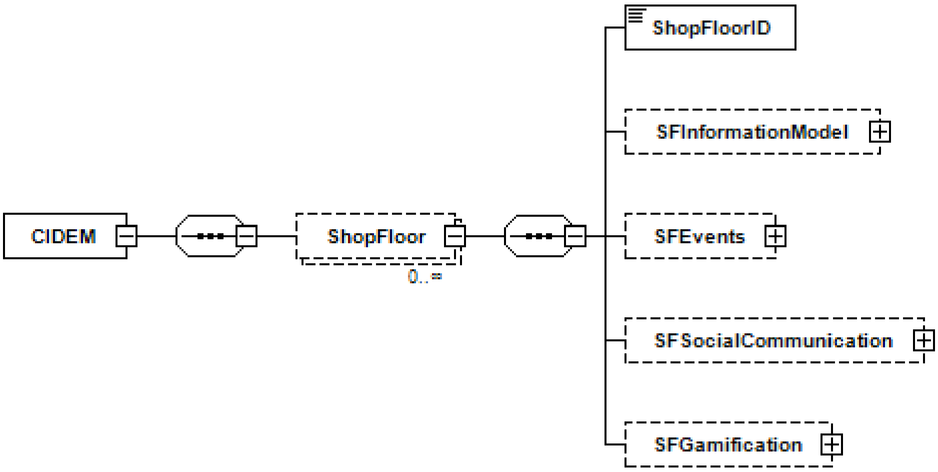


Figure 2: High level structure of the CIDEM

5.1 INFORMATION MODEL

The information model is comprised by the shop-floor static information. A high level schema is depicted at Figure 3. More details about its elements are in the following subsections.

5.1.1 Interfaces

The interfaces that are supported by the CIDEM for the shop-floor are listed below:

Exporting Interfaces

- boolean getShopFloor (string shopfloorID, string gbXML, boolean zip)

  It retrieves the overall information stored in the CIDEM related to the shopfloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. This saves a lot of time. If it is false, then the returned information will be XML file in text form.
- string getAllShopfloorIds()
  It retrieves the names of all shop-floors that are stored in the CIDEM.

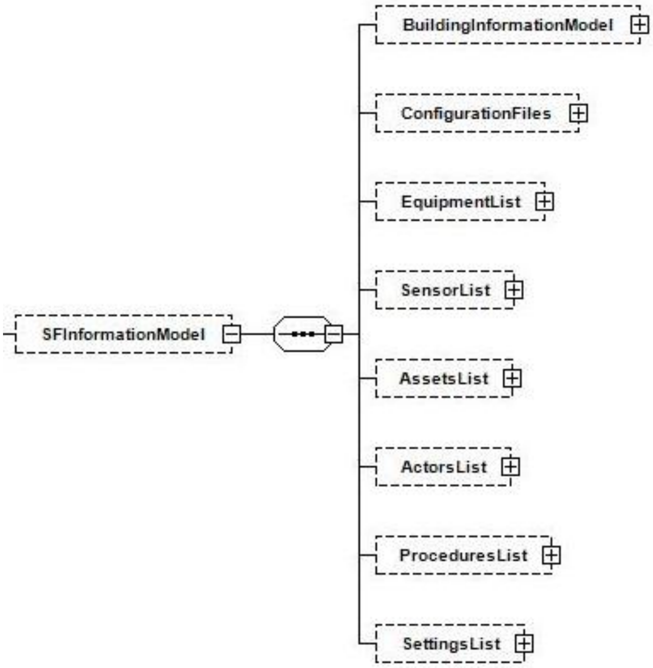


Figure 3: High level structure of the Information Model

5.1.2 BuildingInformationModel

5.1.2.1 Interfaces

The interfaces that are supported by the CIDEM for the building information model are listed below:

Importing Interfaces

- boolean setBuildingInformationModel (string XML)
  It sends for storing an XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor id inside the XML file, and the corresponding building information is filtered and stored in the respective position in the CIDEM.

Exporting Interfaces

- string getBuildingInformationModel (string shopFloorID, boolean zip)
  It retrieves the building information stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.2.2 XSD Schemas

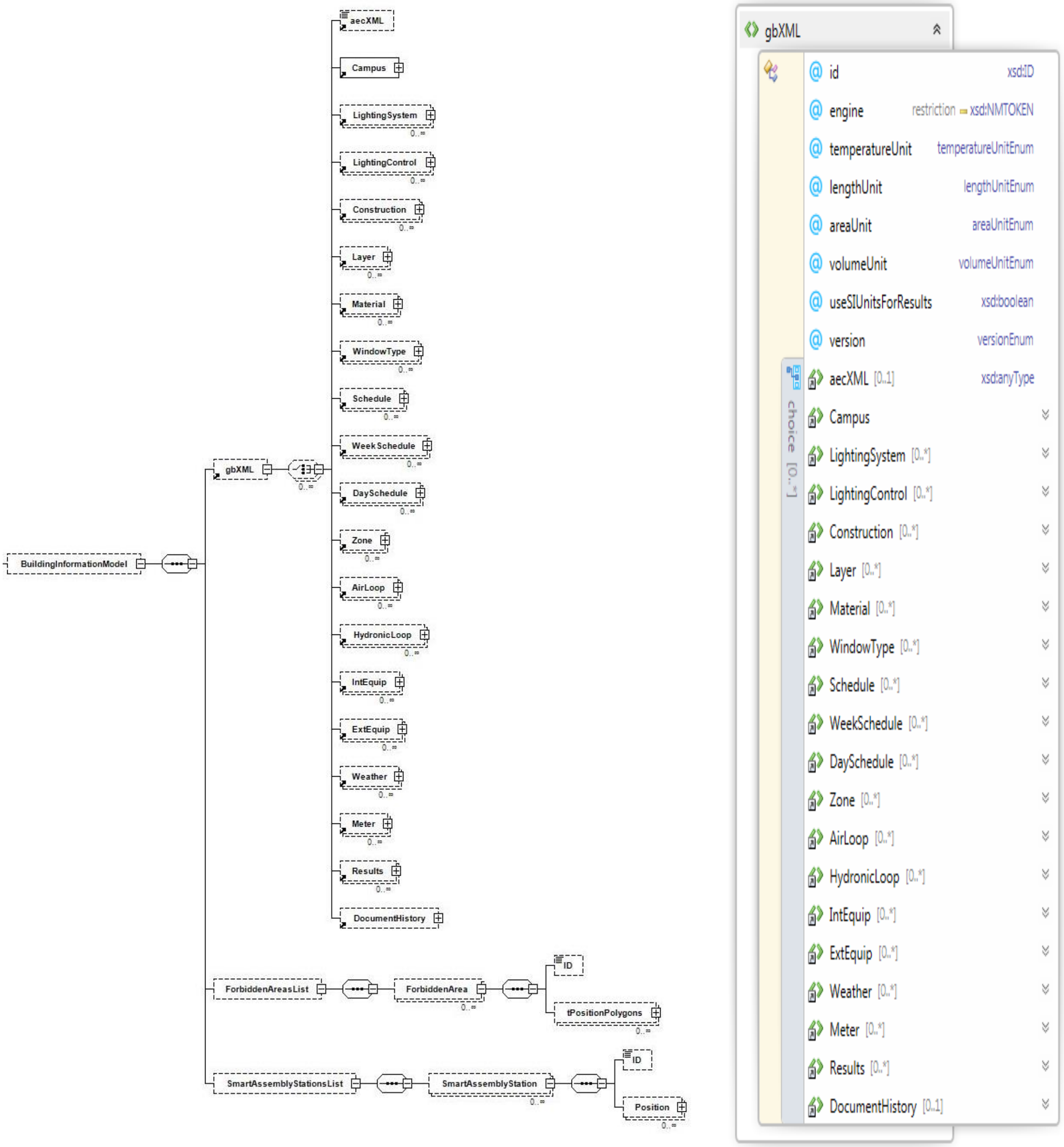


Figure 4: Building Information Model schema

5.1.2.3 gbXML

Interfaces

The interfaces that are supported by the CIDEM for the gbXML (architectural map of the shop-floor) are listed below:

Importing Interfaces

- boolean setgbXML(string XML)
  It sends for storing a XML file (compatible with the gbXML structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding architectural map (gbXML) is stored in the corresponding position in the CIDEM.

Exporting Interfaces

- string getgbXMLByID(string shopFloorID,boolean zip)
  It retrieves the architectural map (in gbXML form) stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.2.4 ForbidenAreasList

The interfaces that are supported by the CIDEM for the forbidden areas in the shop-floor are listed below:

Interfaces

Importing Interfaces

- boolean setForbiddenAreasList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding restricted areas are stored in the corresponding position in the CIDEM.
- boolean setForbiddenArea(string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding restricted area is stored in the corresponding position in the CIDEM.

Exporting Interfaces

- string getForbiddenAreasListByID (string shopFloorID,boolean zip)
  It retrieves the forbidden area list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getForbiddenAreaByID (string shopFloorID, string ForbiddenAreaID, boolean zip)
  It retrieves a specific forbidden area based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.2.5 Smart Assembly Stations List

The interfaces that are supported by the CIDEM for the smart assembly stations in the shop-floor are listed below:

Interfaces

Importing Interfaces

- boolean setSmartAssemblyStationsList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding stations are stored in the corresponding position in the CIDEM.
- boolean setSmartAssemblyStations (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding station is stored in the corresponding position in the CIDEM.

Exporting Interfaces

- string getSmartAssemblyStationsListByID (string shopFloorID, boolean zip)
  It retrieves the smart assembly stations list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getSmartAssemblyStationByID (string shopFloorID, string SmartAssemblyStationID, boolean zip)
  It retrieves the smart assembly station stored in the CIDEM related to the shopFloorID with a specific station ID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.3 Configuration Files List

The interfaces that are supported by the CIDEM for the configuration files (middleware topics) in the shop-floor are listed below:

5.1.3.1 Interfaces

Importing Interfaces

- boolean setConfigurationFile (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding topic is stored in the corresponding position in the CIDEM.

Exporting Interfaces

- string getConfigurationFileByID (string shopFloorID, string ConfigurationFileID, boolean zip)
  It retrieves a specific configuration file based on its ID list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.3.2 XSD Schemas

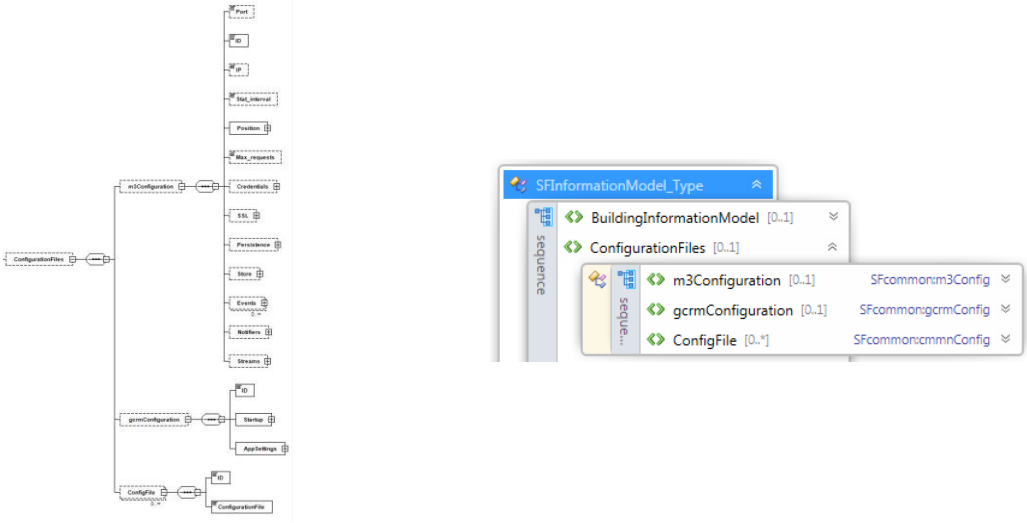


Figure 5: Topic schema

5.1.4 Equipment List

The interfaces that are supported by the CIDEM for the equipment in the shop-floor are listed below:

5.1.4.1 Interfaces

<u>Importing Interfaces</u>

- boolean setEquipment (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding equipment is stored in the corresponding position in the CIDEM.
- boolean setEquipmentList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding equipment list is stored in the corresponding position in the CIDEM replacing the already existing one (if exist).

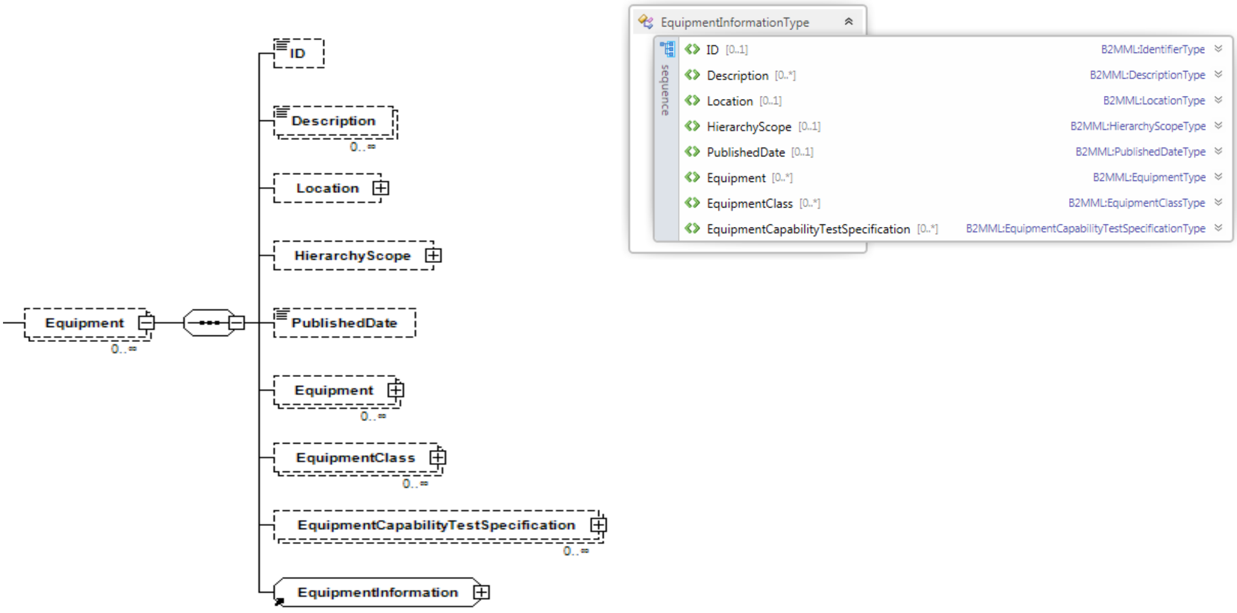
<u>Exporting Interfaces</u>

- string getEquipmentList (string shopFloorID,boolean zip)
  It retrieves the equipment list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getEquipmentByID (string shopFloorID, string equipmentID, boolean zip)
  It retrieves the information of a specific equipment based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned

XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

- string getEquipmentByLocation (string shopFloorID, string spaceID, boolean zip)
It retrieves the information of specific equipment based on the shopFloorID and the spaceID that it is located in. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.4.2 XSD Schemas



ID	[0..1]	B2MML:IdentifierType
Description	[0..*]	B2MML:DescriptionType
Location	[0..1]	B2MML:LocationType
HierarchyScope	[0..1]	B2MML:HierarchyScopeType
PublishedDate	[0..1]	B2MML:PublishedDateType
Equipment	[0..*]	B2MML:EquipmentType
EquipmentClass	[0..*]	B2MML:EquipmentClassType
EquipmentCapabilityTestSpecification	[0..*]	B2MML:EquipmentCapabilityTestSpecificationType

Figure 6: Equipment schema

5.1.5 Sensors List

The interfaces that are supported by the CIDEM for the sensors located in the shop-floor are listed below:

5.1.5.1 Interfaces

Importing Interfaces

- boolean setSensor (string XML)
It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding sensor is stored in the CIDEM.
- boolean setSensorList (string XML)

It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding sensor list is stored in the corresponding position in the CIDEM replacing the already existing one (if exist).

Exporting Interfaces

- string getSensorList (string shopFloorID, boolean zip)
  It retrieves the sensor list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getSensorByID (string shopFloorID, string sensorID, boolean zip)
  It retrieves the information of specific equipment based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.5.2 XSD Schemas

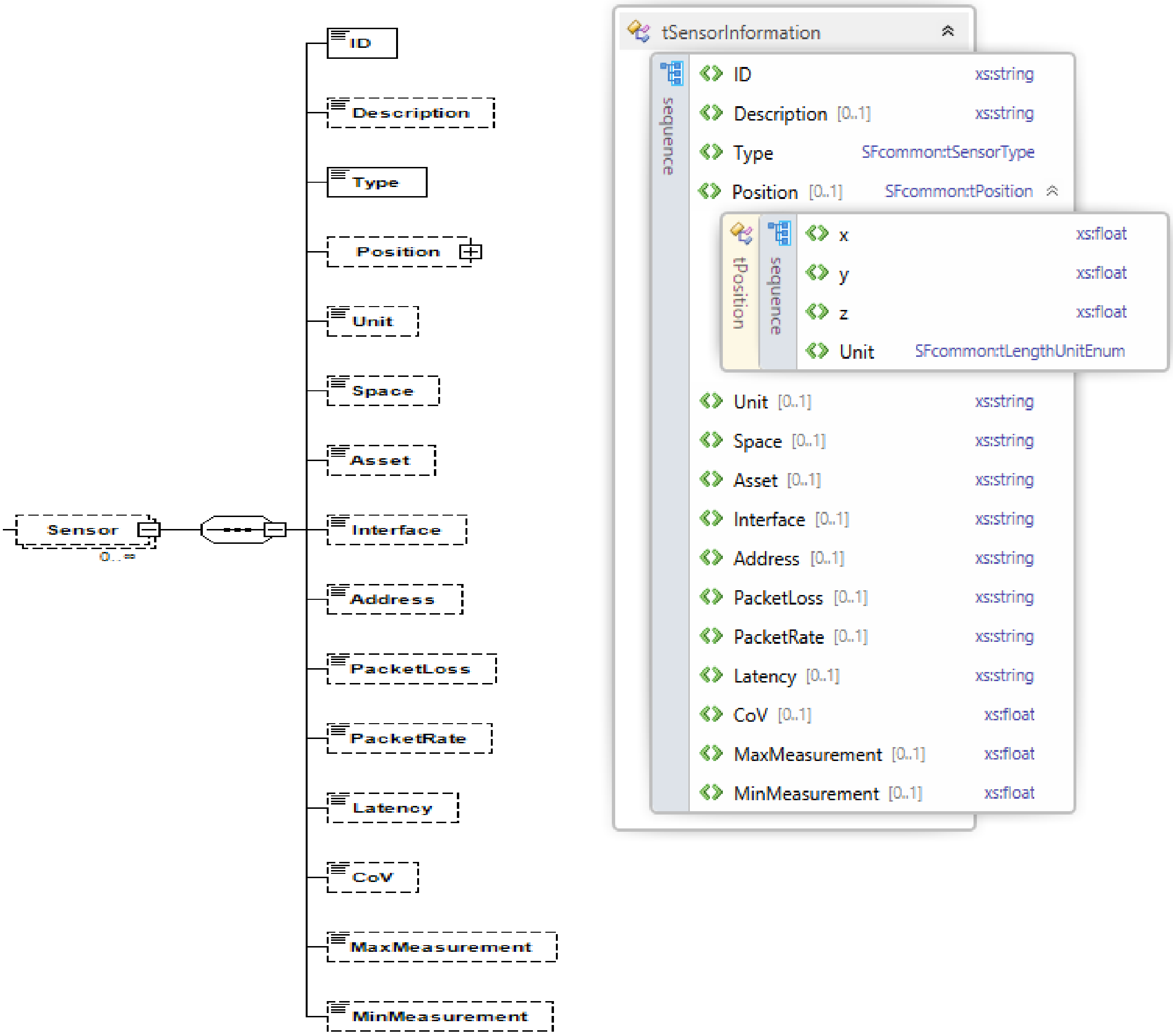


Figure 7: Sensor schema

5.1.6 Assets List

The interfaces that are supported by the CIDEM for the assets located in the shop-floor are listed below:

5.1.6.1 Interfaces

Importing Interfaces

- boolean setAsset (string XML)

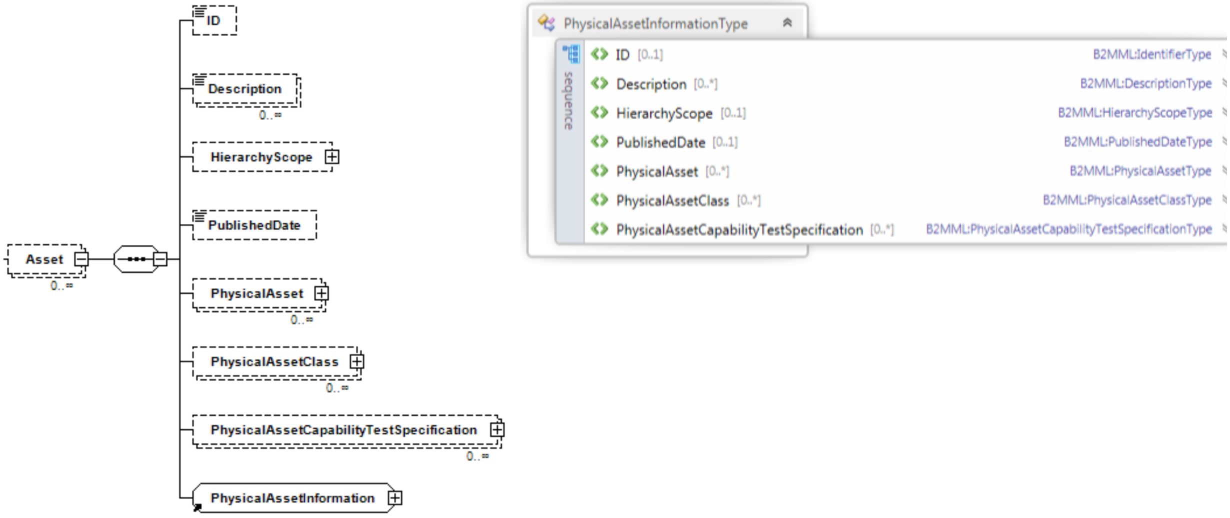
It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding asset is stored in the corresponding position in the CIDEM.

- boolean setAssetList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding asset list is stored in the CIDEM replacing the already existing one (if exist).

Exporting Interfaces

- string getAssetList (string shopFloorID, boolean zip)
  It retrieves the asset list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getAssetByID (string shopFloorID, string assetID, boolean zip)
  It retrieves the information of specific asset based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.6.2 XSD Schemas



ID	[0..1]	B2MML:IdentifierType
Description	[0..*]	B2MML:DescriptionType
HierarchyScope	[0..1]	B2MML:HierarchyScopeType
PublishedDate	[0..1]	B2MML:PublishedDateType
PhysicalAsset	[0..*]	B2MML:PhysicalAssetType
PhysicalAssetClass	[0..*]	B2MML:PhysicalAssetClassType
PhysicalAssetCapabilityTestSpecification	[0..*]	B2MML:PhysicalAssetCapabilityTestSpecificationType

Figure 8: Asset schema

5.1.7 Actors List

The interfaces that are supported by the CIDEM for the actors related to the SatisFactory ise cases are listed below:

5.1.7.1 Interfaces

Importing Interfaces

- boolean setActor (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the information of the actor is stored in the corresponding position in the CIDEM.
- boolean setActorList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding actor list is stored in the CIDEM replacing the already existing one (if exist).
- boolean setActorWithRegistration (MultivaluedMap parameters)
  The method allows the insertion of a new actor via the collaboration platform. The attributes of the new actor are passed by using the POST method. The new actor is stored to the database and its ID number is returned as a response.
- boolean setTagIDtoActorID (string XML, string actorID)
  It assigns to an already stored actor a tag (compatible with the CIDEM structure) to the CIDEM that is used to the collaboration platform. CIDEM reads the shop-floor and actor IDs inside the XML file, and the corresponding tag is stored in the CIDEM enriching or replacing the actor information (if exist).

Exporting Interfaces

- string getActorList (string shopFloorID, boolean zip)
  It retrieves the actor list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getActorByID (string shopFloorID, string actorID, boolean zip)
  It retrieves the information of specific actor based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getActor_by_ID_and_TagID (string shopFloorID, string actorID)
  It retrieves the information of specific actor based on its ID and its Tag ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getPersonnelClasses (string shopFloorID, boolean zip)
  It retrieves the personnel classes of the stored information in the actors list in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getPerson (string shopFloorID)

It retrieves the actor groups from the stored information in the actors list in the CIDEM related to the shopFloorID in JSON format.

- string getQualificationTestSpecification (string shopFloorID)
  It retrieves the experience levels from the stored information in the actors list in the CIDEM related to the shopFloorID in JSON format.

5.1.7.2 XSD Schemas

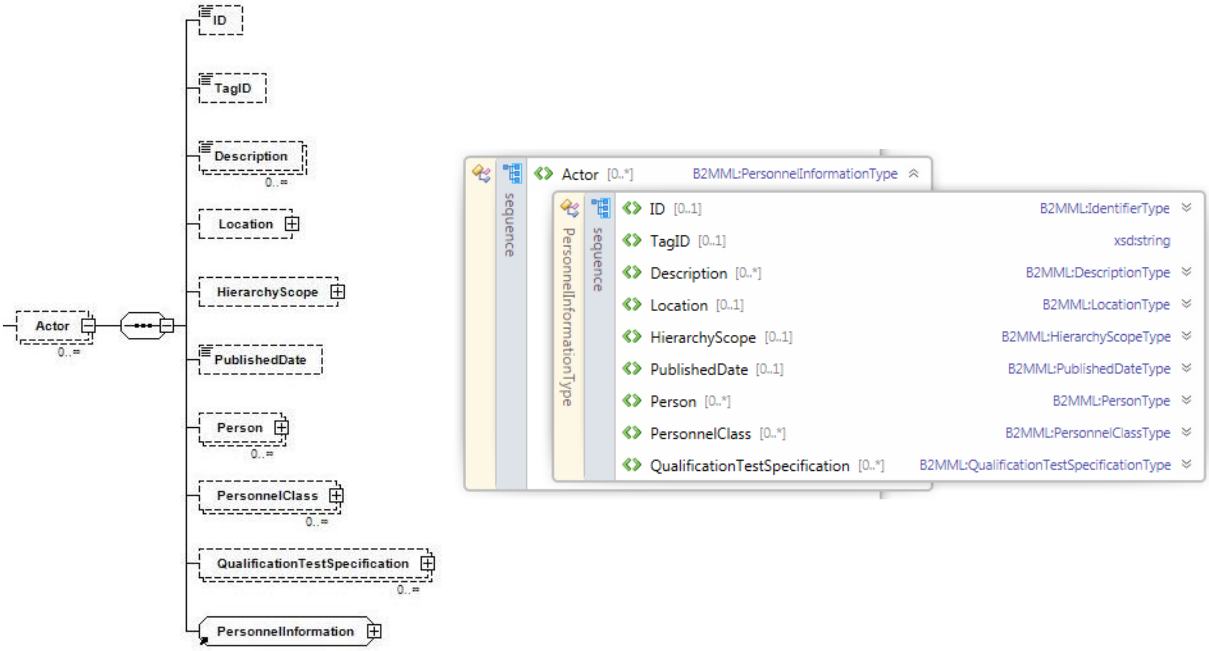

ID
TagID
Description
0..∞
Location
HierarchyScope
Actor
0..∞
PublishedDate
Person
0..∞
PersonnelClass
0..∞
QualificationTestSpecification
0..∞
PersonnelInformation
sequence
Actor [0..*] B2MML:PersonnelInformationType
PersonnelInformationType
sequence
ID [0..1] B2MML:IdentifierType
TagID [0..1] xsd:string
Description [0..*] B2MML:DescriptionType
Location [0..1] B2MML:LocationType
HierarchyScope [0..1] B2MML:HierarchyScopeType
PublishedDate [0..1] B2MML:PublishedDateType
Person [0..*] B2MML:PersonType
PersonnelClass [0..*] B2MML:PersonnelClassType
QualificationTestSpecification [0..*] B2MML:QualificationTestSpecificationType

Figure 9: Actor schema

5.1.8 Procedures List

The interfaces that are supported by the CIDEM for the procedures related to the SatisFactory use cases are listed below:

5.1.8.1 Interfaces

Importing Interfaces

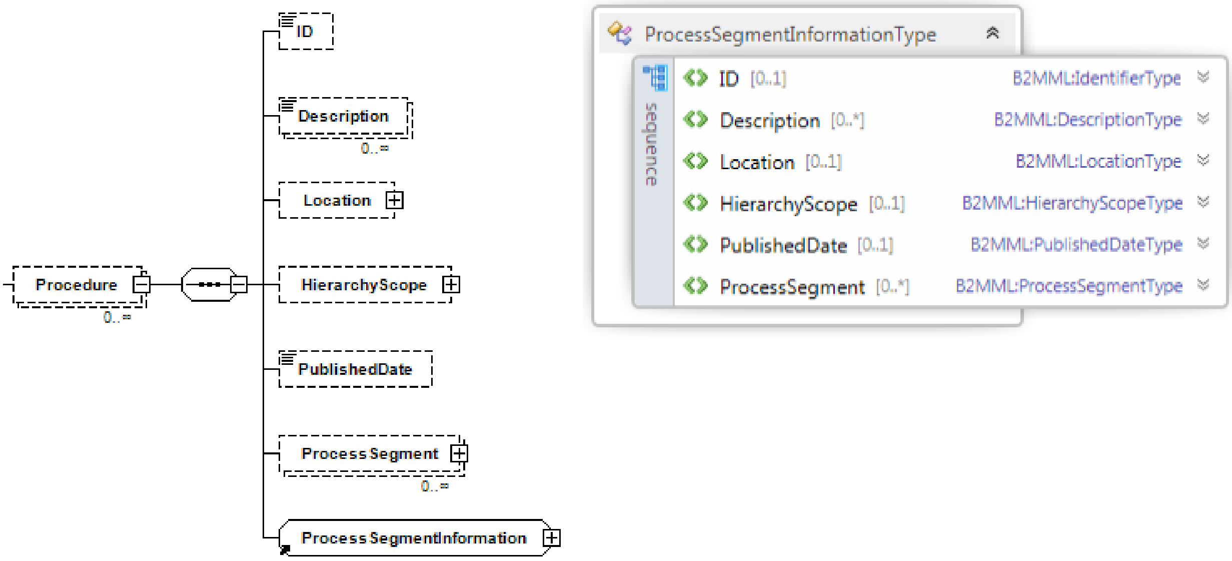
- boolean setProcedure (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the information of the procedure is stored in the corresponding position in the CIDEM.
- boolean setProcedureList (string XML)

It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding procedure list is stored in the CIDEM replacing the already existing one (if exist).

Exporting Interfaces

- string getProcedureList (string shopFloorID, boolean zip)
  It retrieves the procedure list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getProcedureByID (string shopFloorID, string procedureID, boolean zip)
  It retrieves the information of specific procedure based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.1.8.2 XSD Schemas



ID	[0..1]	B2MML:IdentifierType
Description	[0..*]	B2MML:DescriptionType
Location	[0..1]	B2MML:LocationType
HierarchyScope	[0..1]	B2MML:HierarchyScopeType
PublishedDate	[0..1]	B2MML:PublishedDateType
ProcessSegment	[0..*]	B2MML:ProcessSegmentType

Figure 10: Procedure schema

5.2 EVENTS

The events model is comprised by the shop-floor dynamic information. A high level schema is depicted at Figure 11. More details about its elements are in the following subsections.

5.2.1 General

The interfaces that are supported by the CIDEM for the events produced in the shop-floor are listed below:

5.2.1.1 Interfaces

Importing Interfaces

- boolean setEventsList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding list of events is stored in the CIDEM replacing the existing one (if exist).
- boolean setEvent (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding information of the event is stored in the CIDEM.

Exporting Interfaces

- string getEventsList (string shopFloorID, boolean zip)
  It retrieves the events list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getEventByID (string shopFloorID, string EventID, boolean zip)
  It retrieves the information of specific event based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getEventsList_byTYPE (string shopFloorID, string EventType, boolean zip)
  It retrieves the events list stored in the CIDEM related to the shopFloorID and referred to a specific type (e.g. measurements, maintenance, etc.). If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getEventsList_byTYPE_and_Date (string shopFloorID, string EventType, string startdate, string enddate, string type, boolean zip)
  It retrieves the events list stored in the CIDEM related to the shopFloorID and referred to a specific type (e.g. measurements, maintenance, etc.) and occurred within a specific period of time (between startdate and enddate). An addition has been made to support the Ontology Manager; if the type is set as “RDF” the information retrieved is translated through XSL transformation into RDF/XML triplets, in order to be used by the Ontology Manager. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.2.1.2 XSD Schemas

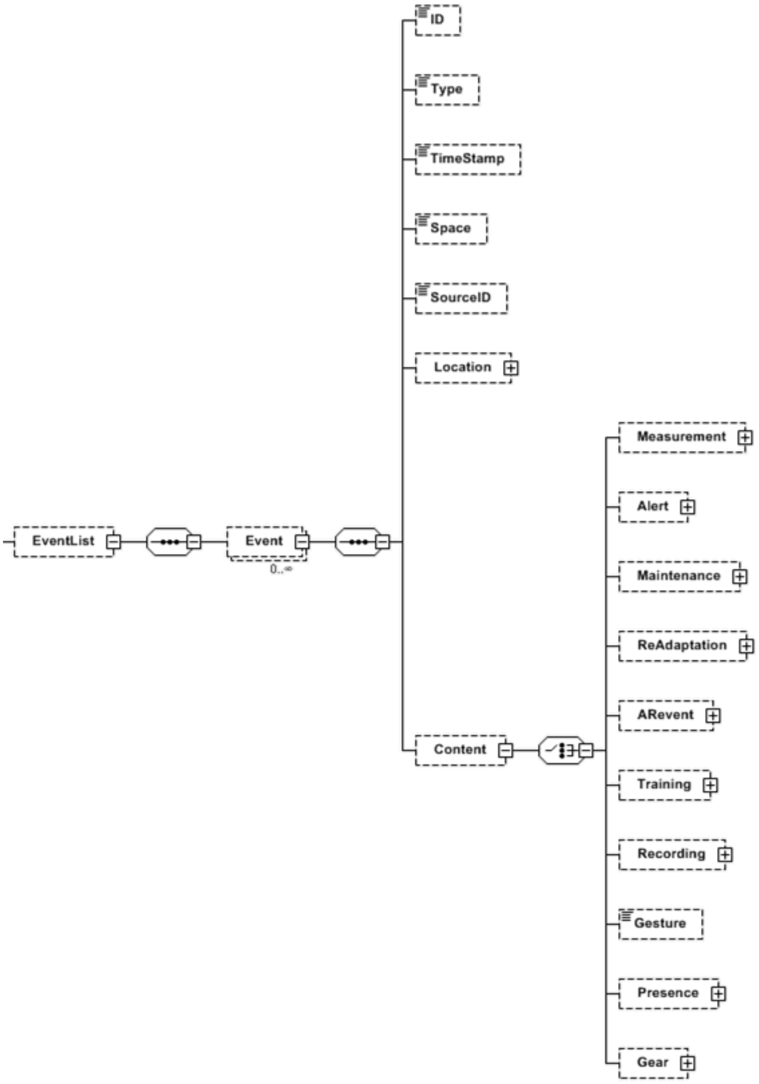
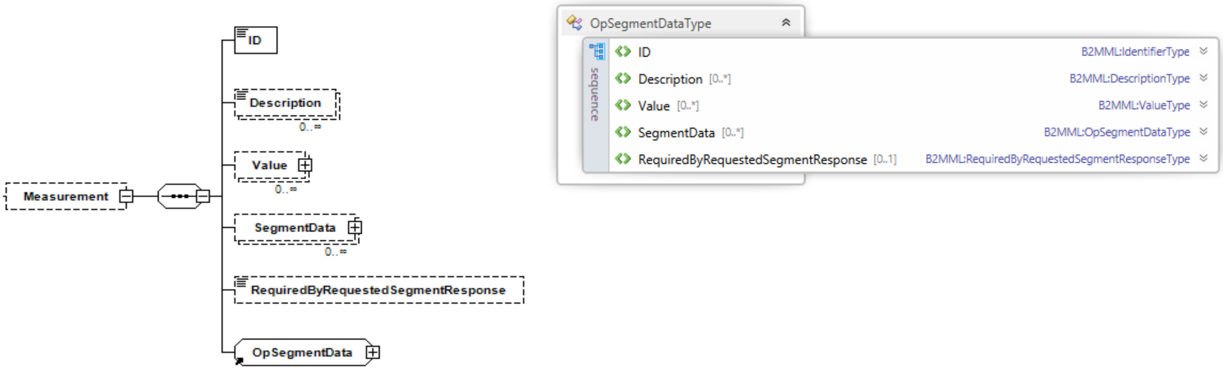


Figure 11: High level structure of the Events model

5.2.2 Measurements

5.2.2.1 XSD Schemas

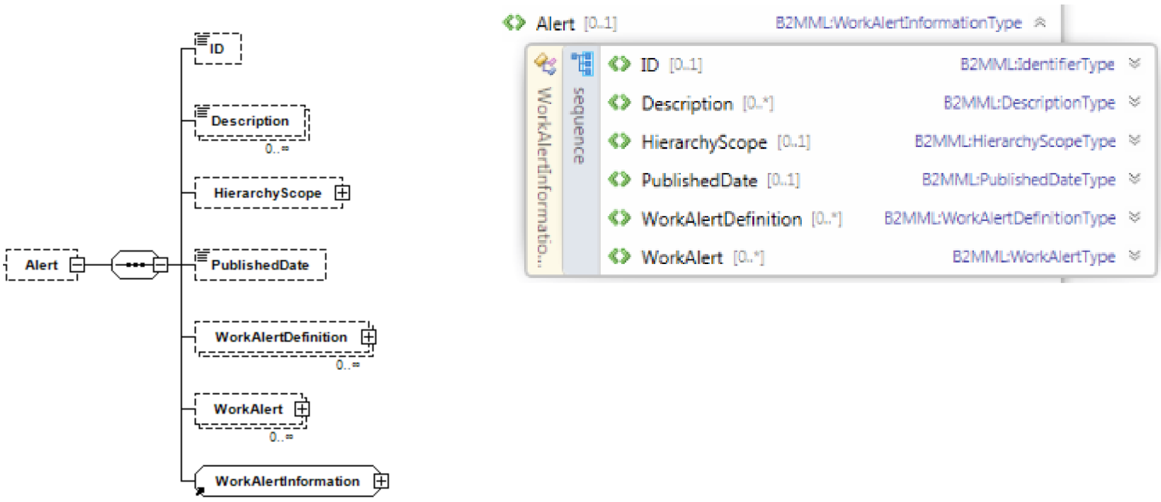


ID		B2MML:IdentifierType
Description	[0..*]	B2MML:DescriptionType
Value	[0..*]	B2MML:ValueType
SegmentData	[0..*]	B2MML:OpSegmentDataType
RequiredByRequestedSegmentResponse	[0..1]	B2MML:RequiredByRequestedSegmentResponseType

Figure 12: Measurement schema

5.2.3 Alerts

5.2.3.1 XSD Schemas



ID	[0..1]	B2MML:IdentifierType
Description	[0..*]	B2MML:DescriptionType
HierarchyScope	[0..1]	B2MML:HierarchyScopeType
PublishedDate	[0..1]	B2MML:PublishedDateType
WorkAlertDefinition	[0..*]	B2MML:WorkAlertDefinitionType
WorkAlert	[0..*]	B2MML:WorkAlertType

Figure 13: Alerts schema

5.2.4 Maintenance Events

5.2.4.1 XSD Schemas

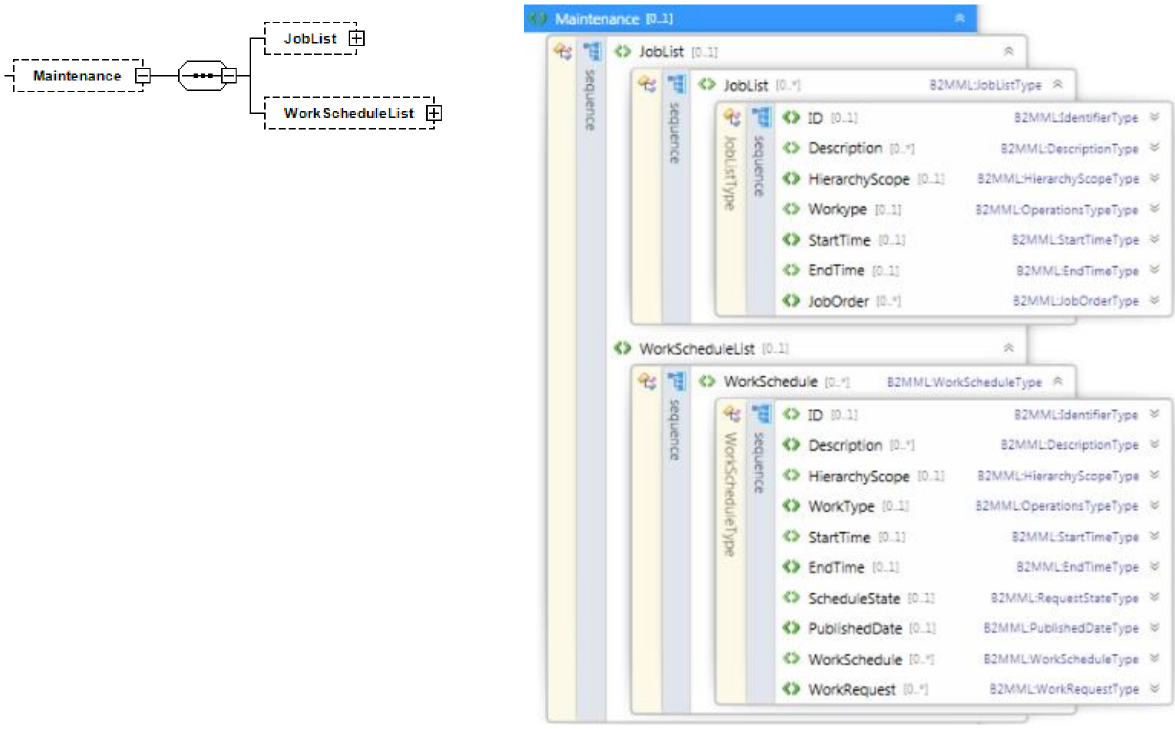
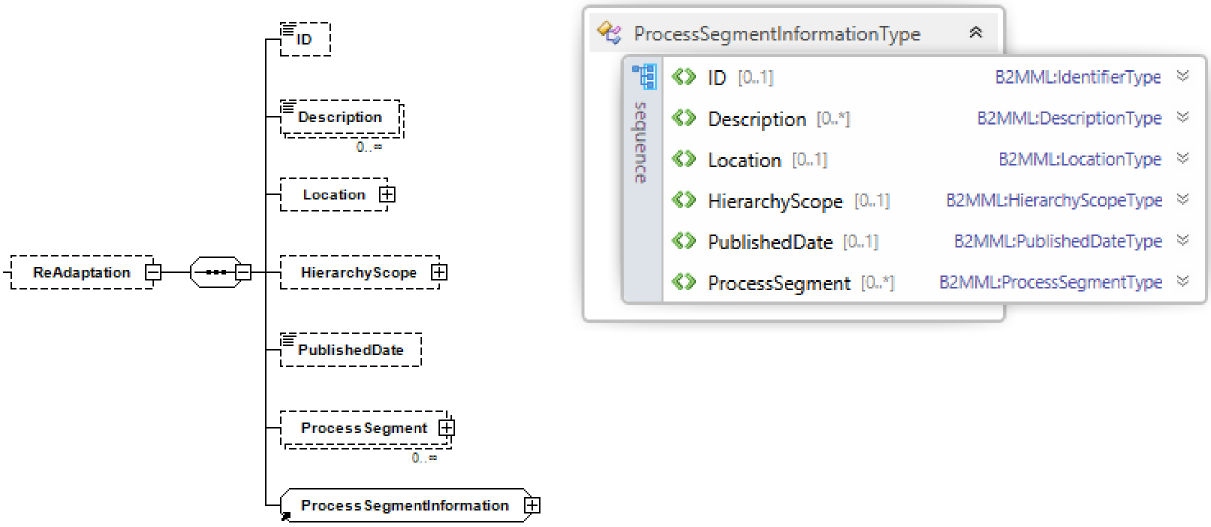

JobList
Maintenance
WorkScheduleList
Maintenance [0..1]
sequence
JobList [0..1]
JobList [0..*] B2MML:JobListType
JobListType
ID [0..1] B2MML:IdentifierType
Description [0..*] B2MML:DescriptionType
HierarchyScope [0..1] B2MML:HierarchyScopeType
Workype [0..1] B2MML:OperationsTypeType
StartTime [0..1] B2MML:StartTimeType
EndTime [0..1] B2MML:EndTimeType
JobOrder [0..*] B2MML:JobOrderType
WorkScheduleList [0..1]
WorkSchedule [0..*] B2MML:WorkScheduleType
WorkScheduleType
ID [0..1] B2MML:IdentifierType
Description [0..*] B2MML:DescriptionType
HierarchyScope [0..1] B2MML:HierarchyScopeType
WorkType [0..1] B2MML:OperationsTypeType
StartTime [0..1] B2MML:StartTimeType
EndTime [0..1] B2MML:EndTimeType
ScheduleState [0..1] B2MML:RequestStateType
PublishedDate [0..1] B2MML:PublishedDateType
WorkSchedule [0..*] B2MML:WorkScheduleType
WorkRequest [0..*] B2MML:WorkRequestType

Figure 14: Maintenance schema

5.2.5 Re-Adaptation Events

5.2.5.1 XSD Schemas

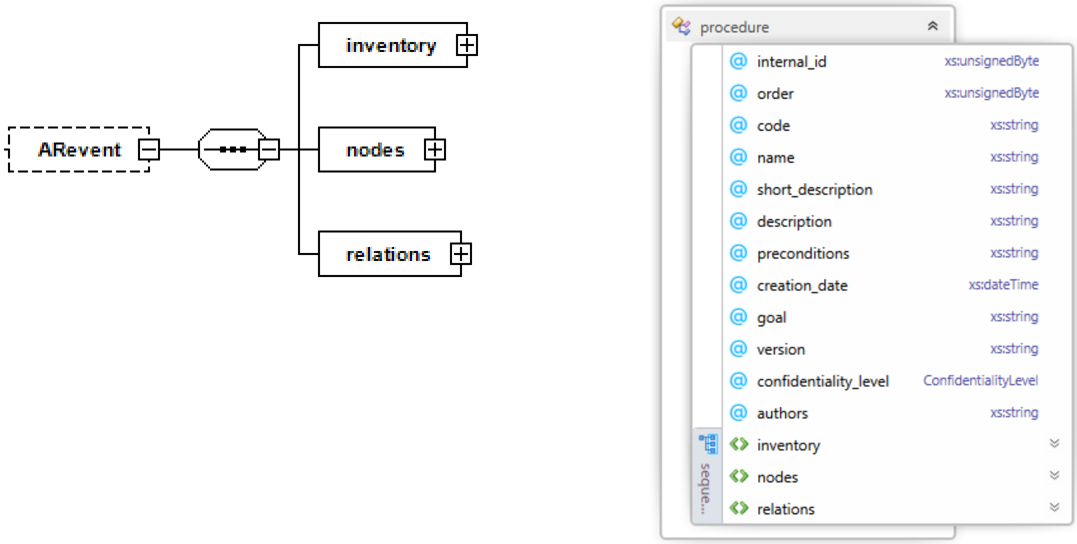


ID	[0..1]	B2MML:IdentifierType
Description	[0..*]	B2MML:DescriptionType
Location	[0..1]	B2MML:LocationType
HierarchyScope	[0..1]	B2MML:HierarchyScopeType
PublishedDate	[0..1]	B2MML:PublishedDateType
ProcessSegment	[0..*]	B2MML:ProcessSegmentType

Figure 15: Re-Adaptation schema

5.2.6 Augmented Reality Events

5.2.6.1 XSD Schemas

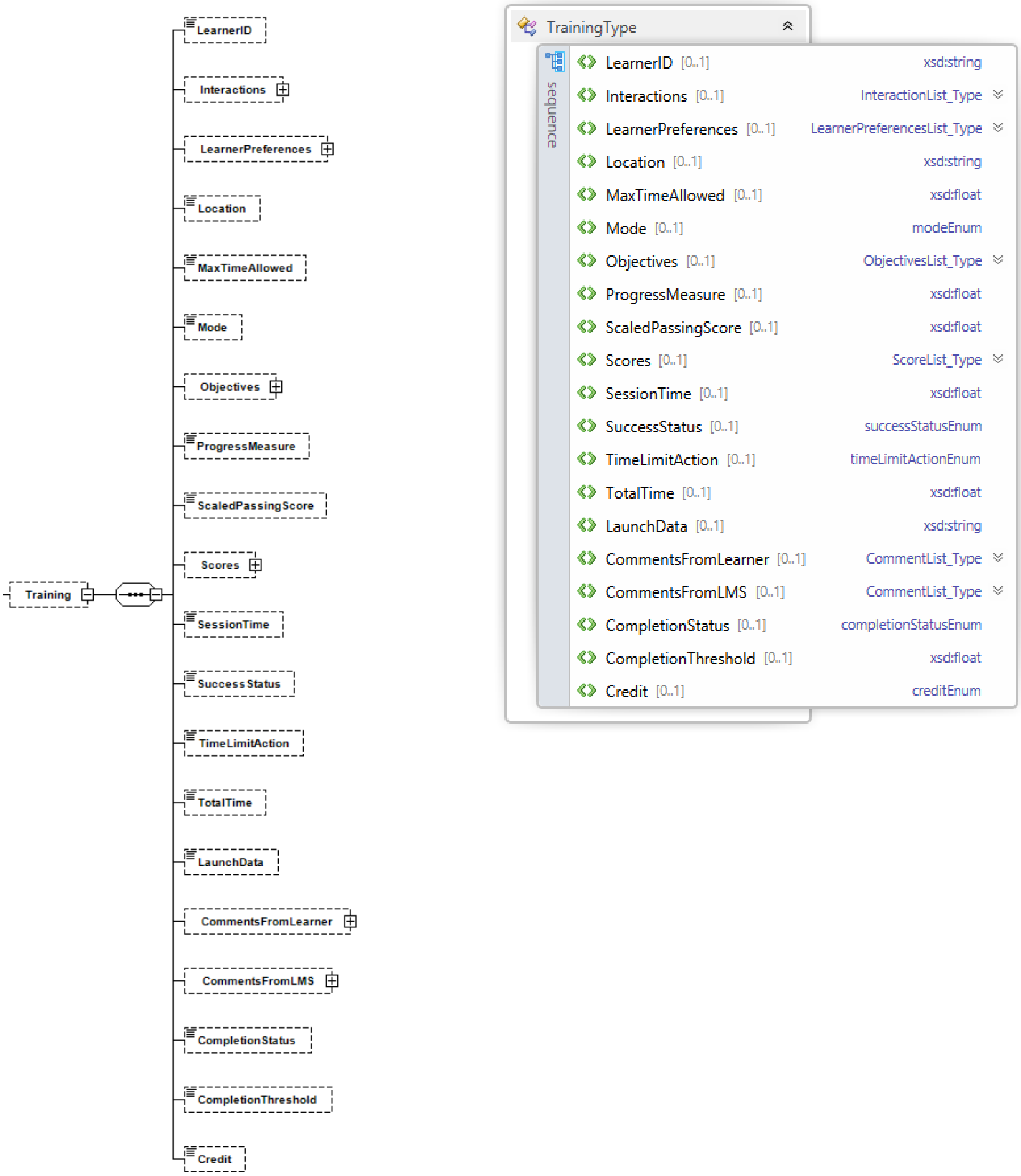


internal_id	xs:unsignedByte
order	xs:unsignedByte
code	xs:string
name	xs:string
short_description	xs:string
description	xs:string
preconditions	xs:string
creation_date	xs:dateTime
goal	xs:string
version	xs:string
confidentiality_level	ConfidentialityLevel
authors	xs:string
inventory	
nodes	
relations	

Figure 16: Augmented Reality Events schema

5.2.7 Training Events

5.2.7.1 XSD Schemas



LearnerID	[0..1]	xsd:string
Interactions	[0..1]	InteractionList_Type
LearnerPreferences	[0..1]	LearnerPreferencesList_Type
Location	[0..1]	xsd:string
MaxTimeAllowed	[0..1]	xsd:float
Mode	[0..1]	modeEnum
Objectives	[0..1]	ObjectivesList_Type
ProgressMeasure	[0..1]	xsd:float
ScaledPassingScore	[0..1]	xsd:float
Scores	[0..1]	ScoreList_Type
SessionTime	[0..1]	xsd:float
SuccessStatus	[0..1]	successStatusEnum
TimeLimitAction	[0..1]	timeLimitActionEnum
TotalTime	[0..1]	xsd:float
LaunchData	[0..1]	xsd:string
CommentsFromLearner	[0..1]	CommentList_Type
CommentsFromLMS	[0..1]	CommentList_Type
CompletionStatus	[0..1]	completionStatusEnum
CompletionThreshold	[0..1]	xsd:float
Credit	[0..1]	creditEnum

Figure 17: Training Events schema

5.2.8 Recording Events

5.2.8.1 XSD Schemas

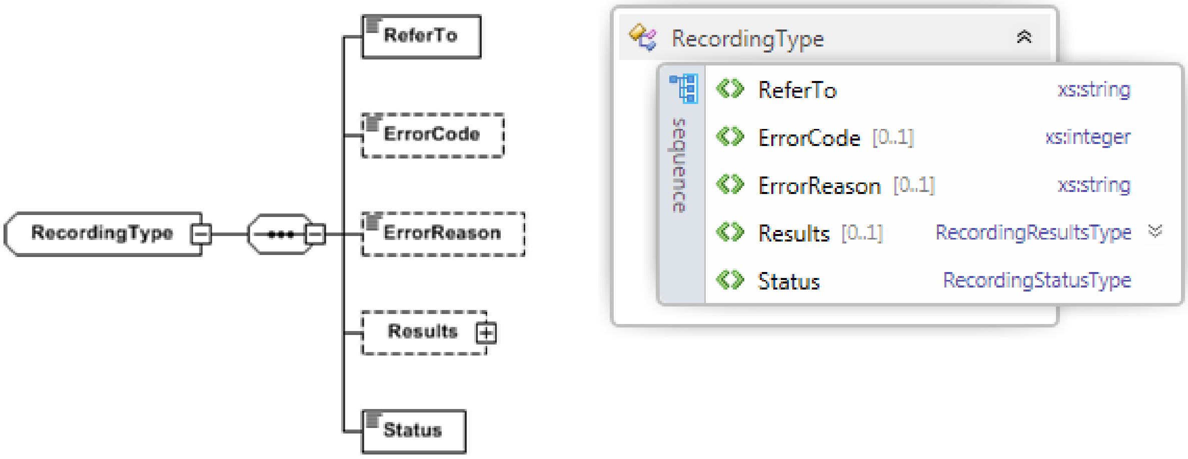

ReferTo
ErrorCode
RecordingType
ErrorReason
Results
Status
RecordingType
sequence
ReferTo xs:string
ErrorCode [0..1] xs:integer
ErrorReason [0..1] xs:string
Results [0..1] RecordingResultsType
Status RecordingStatusType

Figure 18: Recording Events schema

5.2.9 Gesture Events

5.2.9.1 XSD Schemas

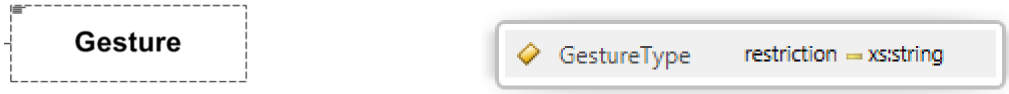


Figure 19: Gesture Events schema

5.2.10 Presence Events

5.2.10.1 XSD Schemas

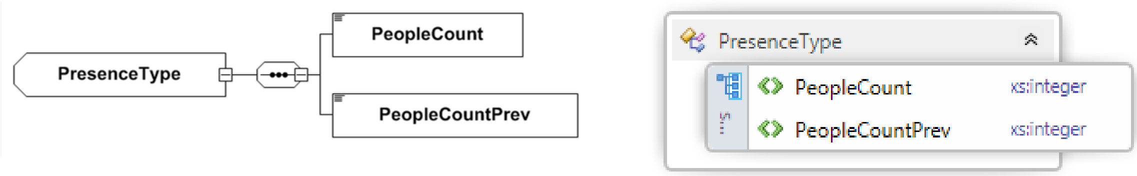


Figure 20: Presence Events schema

5.2.11 Gear Events

5.2.11.1 XSD Schemas

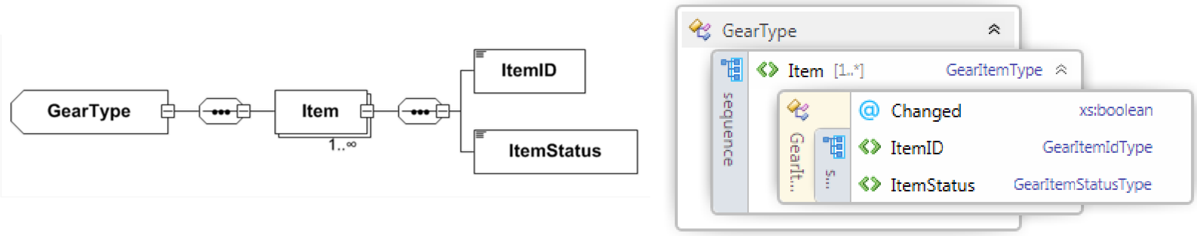


Figure 21: Gear Events schema

5.2.12 Augmented Reality Logging Events

5.2.12.1 XSD Schemas

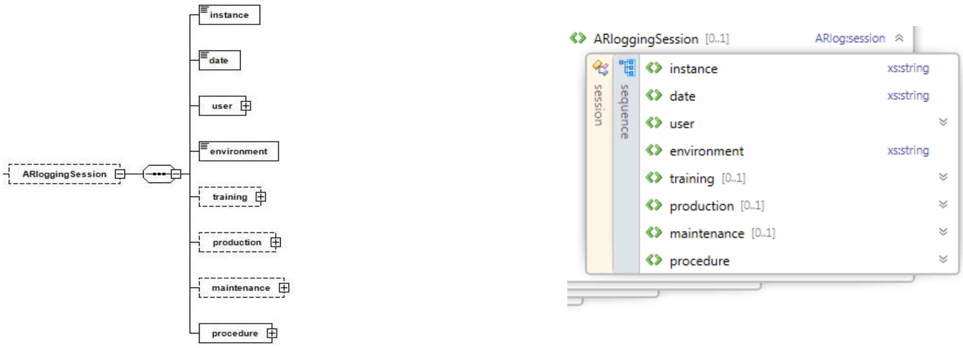


Figure 22: AR Logging Events schema

5.3 SOCIAL COMMUNICATION

The interfaces that are supported by the CIDEM for the events related to the social communication in the shop-floor are listed below:

5.3.1.1 Interfaces

Importing Interfaces

- boolean setCommunicationList (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding social communication event list is stored in the CIDEM replacing the already existing one (if exist).
- boolean setCommunicationEvent (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding social communication event is stored in the CIDEM.

Exporting Interfaces

- string getCommunicationList (string shopFloorID)
  It retrieves the communication events list stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getCommunicationEventByID (string shopFloorID, string EventID)
  It retrieves the information of specific communication event based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getCommunicationList_byTYPE (string shopFloorID, string EventType, boolean zip)
  It retrieves the communication events list stored in the CIDEM related to the shopFloorID and referred to a specific type. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.
- string getCommunicationList_byTYPE_and_Date (string shopFloorID, string EventType, string startdate,string enddate,boolean zip)
  It retrieves the communication events list stored in the CIDEM related to the shopFloorID and referred to a specific type and occurred within a specific period of time (between startdate and enddate). If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.3.2 XSD Schemas

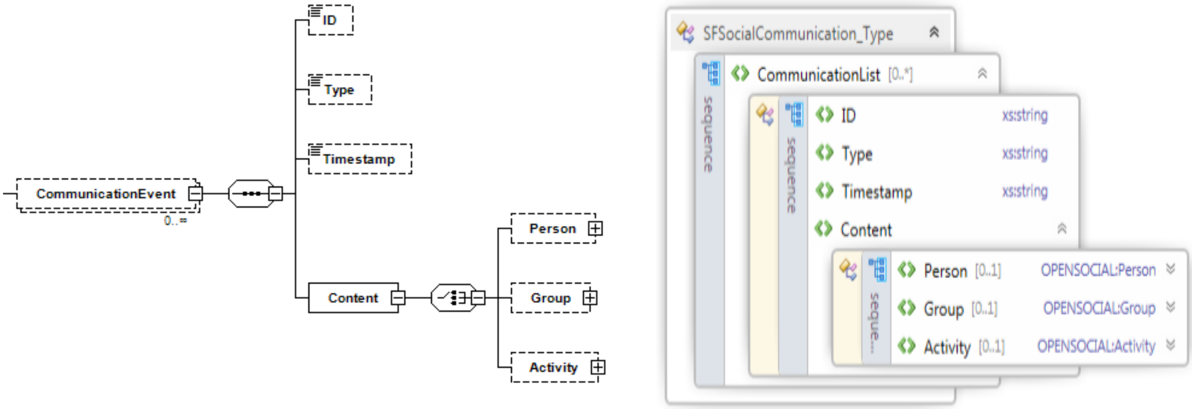


Figure 23: Social Communication Events schema

5.4 GAMIFICATION

The interfaces that are supported by the CIDEM for the events related to the gamification are listed below:

5.4.1 Interfaces

Importing Interfaces

- boolean setGamificationEvent (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM. CIDEM reads the shop-floor ID inside the XML file, and the corresponding information for the events related to the gamification is stored in the CIDEM.
- boolean updateGamificationEvent (string XML)
  It sends for storing a XML file (compatible with the CIDEM structure) to the CIDEM replacing an existing gamification element based on the ID. CIDEM reads the shop-floor ID inside the XML file, as well as the gamification element's id and the corresponding gamification element is stored in the CIDEM replacing the existing one (if exist).

Exporting Interfaces

- string getGamificationEventByID (string shopFloorID, string gamificationID)
  It retrieves the information of specific event based on its ID stored in the CIDEM related to the shopFloorID. If the boolean value zip is true, then the returned XML will be zipped and returned as a zipped file. If it is false, then the returned information will be a XML file in text form.

5.4.2 XSD Schemas

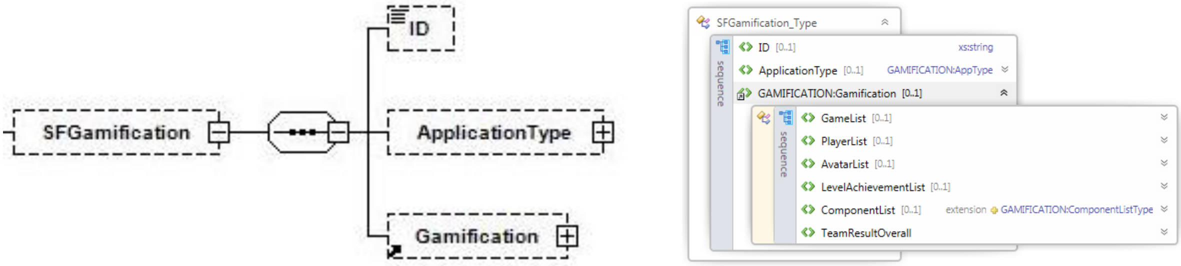

ID
SFGamification
ApplicationType
Gamification
SFGamification_Type
sequence
ID [0..1] xs:string
ApplicationType [0..1] GAMIFICATION:AppType
GAMIFICATION:Gamification [0..1]
sequence
GameList [0..1]
PlayerList [0..1]
AvatarList [0..1]
LevelAchievementList [0..1]
ComponentList [0..1] extension GAMIFICATION:ComponentListType
TeamResultOverall

Figure 24: Gamification Events schema

6. CIDEM Hardware/Software Requirements

The CIDEM consists of software components that are directly linked solely to other software sub-components. In order to provide fast access to the data stored in the repository the machine hosting this component needs to be a high performance machine, as indicated in the following table.

Hardware Requirements	Intel Core i7 3.5GHz
	16 GB RAM
	1 TB hard disk
	64-bit Windows 7 Professional operating system
	1 Gbps on-board LAN (300 Mbps dedicated)

Beside hardware restriction, in order to install CIDEM to the SatisFactory pilots, a variety of software components is also needed, as indicated in the following table.

Software Requirements	Java JDK/JRE 1.8 64bit
	MongoDB 3.2 (Database)
	BaseX 8.3 (Database)
	Glashfish 4.1.1 (Server)

More information can be found in D5.1 “SatisFactory Components Integration and Framework” [6].

7. Technologies Used For The CIDEM and CIDEM APIs

CIDEM is provided as a set of XSD schemas that define the structure of the model. For reusability and easier readability of the model, it is splitted into several parts. Every part of the CIDEM has its target namespace and a corresponding XSD source file:

http://www.satisfactory-project.eu/XMLschema/v1.0/CIDEM (SatisFactory.xsd)

- Main file that can be used to validate XMLs.

http://www.satisfactory-project.eu/XMLschema/v1.0/common (SatisFactory-Common.xsd)

- Common elements of the CIDEM.

http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events (Satisfactory_SasEvents.xsd)

- Common elements of the CIDEM.

http://www.satisfactory-project.eu/XMLschema/v1.0/gbXML (gbXML_v5.12.xsd)

- ShopFloor model elements.

http://www.satisfactory-project.eu/XMLschema/v1.0/B2MML (B2MML.xsd)

- Equipment, Assets, Actors, Procedures, Measurements, Maintenance events, Alerts, and Re-Adaptation elements.

http://www.satisfactory-project.eu/XMLschema/v1.0/R3D (R3D.xsd)

- Augmented reality elements.

http://www.satisfactory-project.eu/XMLschema/v1.0/SCORM (SCORM.xsd)

- Elements related to the training activities.

http://www.satisfactory-project.eu/XMLschema/v1.0/OpenSocial (OpenSocial.xsd)

- Elements related to the social communication activities in the shop-floor.

http://www.satisfactory-project.eu/XMLschema/v1.0/Gaming (Gaming.xsd)

- Elements related to the gamification activities in the shop-floor.

To simplify the instantiation, XSD schemas use the unqualified form of elements and attributes:

elementFormDefault="unqualified" attributeFormDefault="unqualified"

For the storage, CIDEM will still have to choose appropriate technology. NoSQL and XML databases are the probable choice, as it will make transformation from/to XML straightforward. However this still has to be decided during the upcoming project development phases.

CONCLUSION

This report presented the analysis of requirements to the SatisFactory Common Information Data Exchange Model (CIDEM). Existing information models from area of factories were analysed. CIDEM elements were described (using XSD schemas) and their usage was in the specification section. Technologically the proposed CIDEM is a set of XSD schemas defining several data elements needed for exchange of information between SatisFactory components. It was proposed that these components access CIDEM via CIDEM API.

The documentation of XSD schemas is provided in Annex I.

The CIDEM was designed to provide definition of SatisFactory shared vocabulary and meta-data. Proposed CIDEM contains the description of the information sources from particular modules to be able to use them within pilot execution. Therefore it can be the backbone for the system. Furthermore basic interfaces have been implemented for the communication of the CIDEM with the rest of the SatisFactory components.

In this state of the project it became clear that the current version of the CIDEM is not the final one. There are still ongoing discussions that are/will be reflected in CIDEM. Also the use of CIDEM after the first implementation and its update will for sure bring some new requirements to be adopted in it. Furthermore, the implementation of the technical parts of the project will provide valuable information, which will be evaluated and included in the next and final version of the deliverable.

This version of the CIDEM contains the updates after the first approach of the implementation of the SatisFactory tools. A large number of interfaces, as well as XSD files and structures have been changed, including the removal of MIMOSA standard and its replacement with the corresponding structures from the B2MML standard, the adoption of the R3D model as defined by REGOLA, which has a long experience in augmented reality applications, etc. Furthermore, the Event list has been enriched with four (4) new kinds of events (Recording, Gesture, Presence and Gear events). Finally, topics static information has been added to the shop-floor Information model, as well as the Forbidden areas list.

REFERENCES

[1] SatisFactory Grant Agreement Annex I – “Description of Work” (DoW)

[2] Common Information Model (computing) http://en.wikipedia.org/wiki/Common_Information_Model_%28computing%29 (11/06/2015)

[3] SatisFactory Deliverable D2.1, D2.1 “SatisFactory System Architecture”, August 2015

[4] Adapt4EE project, http://www.adapt4ee.eu/

[5] INERTIA project, http://www.inertia-project.eu/

[6] SatisFactory Deliverable D5.1, D5.1 “SatisFactory Components Integration and Framework”, December 2016.

ANNEX I: SATISFACTORY CIDEM XSD FILES DOCUMENTATION

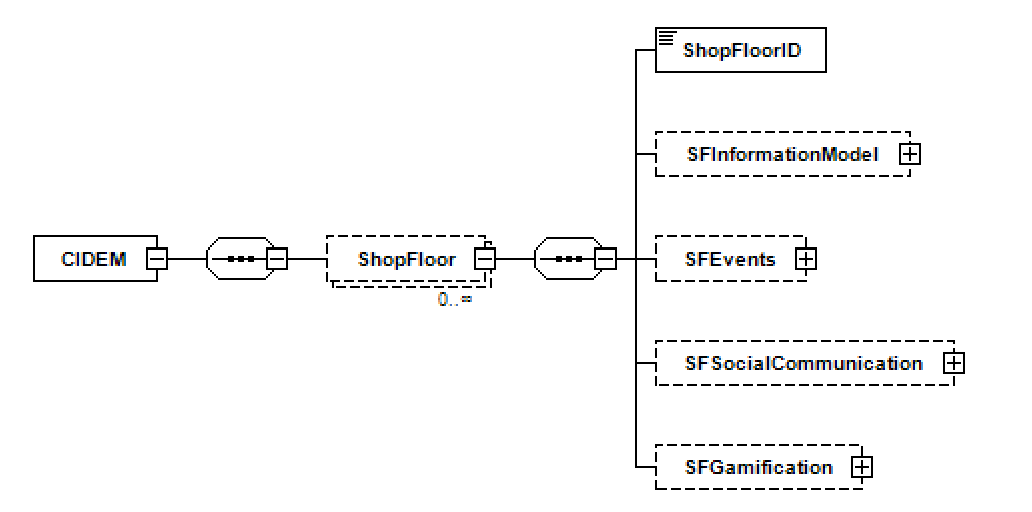
SCHEMA SATISFACTORY.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/**

Elements	**Complex Types**
CIDEM	SFInformationModel_Type SFEvents SFSocialCommunication_Type SFGamification_Type

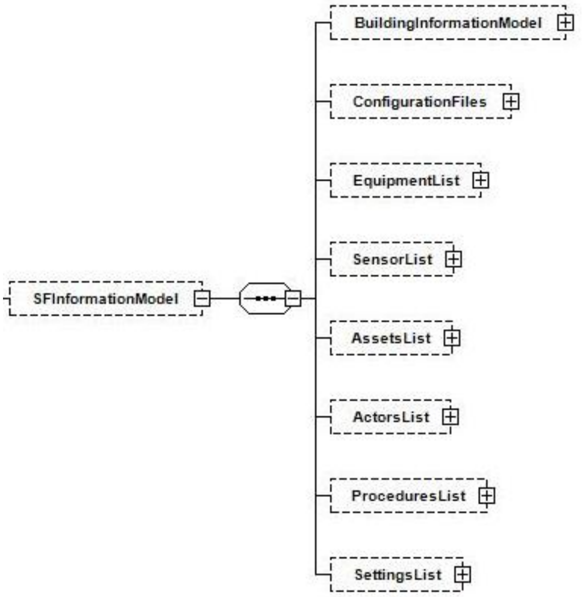
element **CIDEM**

diagram	 ShopFloorID SFInformationModel CIDEM ShopFloor 0..∞ SFEvents SFSocialCommunication SFGamification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/
children	ShopFloorID, SFInformationModel, SFEvents, SFSocialCommunication, SFGamification

source	<xs:element name="CIDEM">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="ShopFloor" minOccurs="0" maxOccurs="unbounded">
	        <xs:complexType>
	          <xs:sequence>
	            <xs:element name="ShopFloorID" type="xs:string" minOccurs="1" maxOccurs="1"/>
	            <xs:element name="SFInformationModel" type="SFInformationModel_Type"

	minOccurs="0" maxOccurs="1"/> <xs:element name="SFEvents" type="SFEvents_Type" minOccurs="0" maxOccurs="1"/> <xs:element name="SFSocialCommunication" type="SFSocialCommunication_Type" minOccurs="0" maxOccurs="1"/> <xs:element name="SFGamification" type="SFGamification_Type" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element>

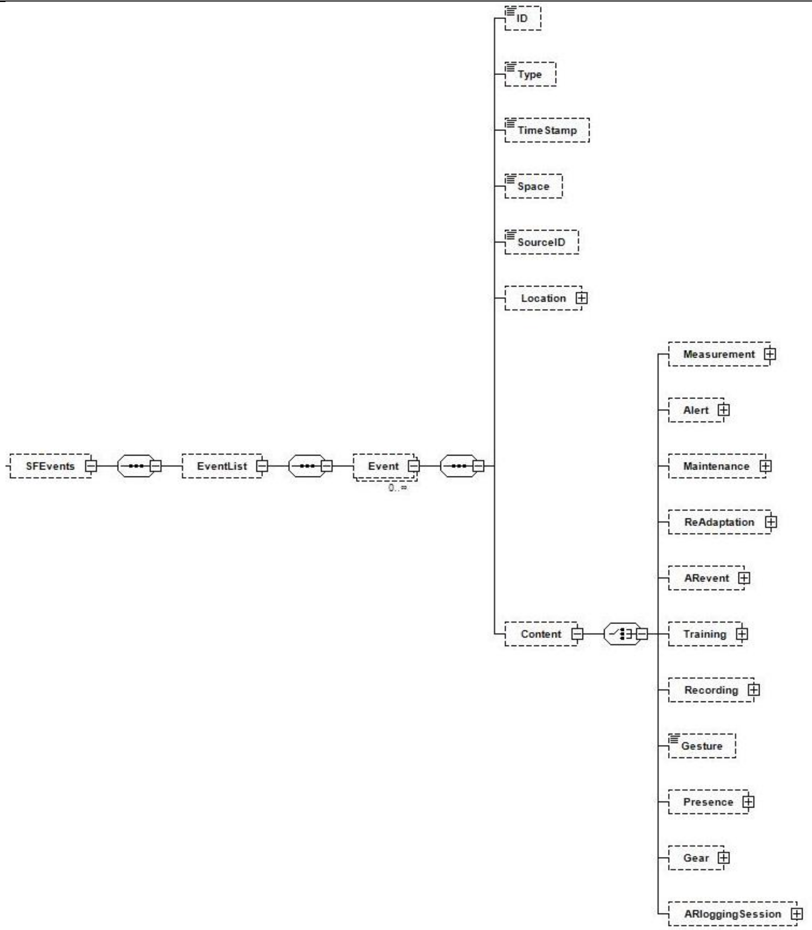
complexType **SFInformationModel_Type**

diagram	 SFInformationModel BuildingInformationModel ConfigurationFiles EquipmentList SensorList AssetsList ActorsList ProceduresList SettingsList
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/
children	BuildingInformationModel, ConfigurationFiles, EquipmentList, SensorList, AssetsList, ActorsList, ProceduresList, Settings List
source	<xs:complexType name="SFInformationModel_Type"> <xs:sequence> <xs:element name="BuildingInformationModel" minOccurs="0" maxOccurs="1" > <xs:complexType> <xs:sequence> <xs:element ref="gbxml:gbXML" minOccurs="0" maxOccurs="1" /> <xs:element name="ForbiddenAreasList" minOccurs="0" maxOccurs="1" > <xs:complexType> <xs:sequence> <xs:element name="ForbiddenArea"

```
minOccurs="0" maxOccurs="unbounded">
                                             <xs:complexType>
                                                   <xs:sequence>
                                                          <xs:element
type="xs:string" name="ID" minOccurs="0" maxOccurs="1"/>
                                                          <xs:element
type="SFcommon:tPosition" name="tPositionPolygons" minOccurs="0"
maxOccurs="unbounded" />
                                                   </xs:sequence>
                                             </xs:complexType>
                                       </xs:element>
                                 </xs:sequence>
                                </xs:complexType>
                          </xs:element>
                          <xs:element name="SmartAssemblyStationsList"
minOccurs="0" maxOccurs="1" >
                                <xs:complexType>
                                 <xs:sequence>
                                       <xs:element
name="SmartAssemblyStation" minOccurs="0" maxOccurs="unbounded">
                                             <xs:complexType>
                                                   <xs:sequence>
                                                          <xs:element
type="xs:string" name="ID" minOccurs="0" maxOccurs="1"/>
                                                          <xs:element
type="SFcommon:tPosition" name="Position" minOccurs="0"
maxOccurs="unbounded" />
                                                   </xs:sequence>
                                             </xs:complexType>
                                       </xs:element>
                                 </xs:sequence>
                                </xs:complexType>
                          </xs:element>
      </xs:sequence>
    </xs:complexType>
   </xs:element>
       <xs:element name="ConfigurationFiles" minOccurs="0" maxOccurs="1" >
    <xs:complexType>
     <xs:sequence>
                   <xs:element type="SFcommon:m3Config"
name="m3Configuration" minOccurs="0" />
                   <xs:element type="SFcommon:gcrmConfig"
name="gcrmConfiguration" minOccurs="0"/>
                   <xs:element type="SFcommon:cmmnConfig"
name="ConfigFile" minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="EquipmentList" minOccurs="0" maxOccurs="1" >
     <xs:complexType>
       <xs:sequence>
                                <xs:element type="B2MML:EquipmentInformationType"
name="Equipment" minOccurs="0" maxOccurs="unbounded" />
       </xs:sequence>
     </xs:complexType>
   </xs:element>
          <xs:element name="SensorList" minOccurs="0" maxOccurs="1" >
     <xs:complexType>
       <xs:sequence>
         <xs:element type="SFcommon:tSensorInformation" name="Sensor"
maxOccurs="unbounded" minOccurs="0"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
          <xs:element name="AssetsList" minOccurs="0" maxOccurs="1" >
     <xs:complexType>
       <xs:sequence>
                                <xs:element type="B2MML:PhysicalAssetInformationType"
name="Asset" minOccurs="0" maxOccurs="unbounded" />
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="ActorsList" minOccurs="0" maxOccurs="1" >
     <xs:complexType>
       <xs:sequence>
                                <xs:element type="B2MML:PersonnelInformationType"
name="Actor" minOccurs="0" maxOccurs="unbounded" />
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="ProceduresList" minOccurs="0" maxOccurs="1" >
     <xs:complexType>
       <xs:sequence>
                                <xs:element type="B2MML:ProcessSegmentInformationType"
name="Procedure" minOccurs="0" maxOccurs="unbounded" />
       </xs:sequence>
     </xs:complexType>
   </xs:element>
          <xs:element name="SettingsList" minOccurs="0" maxOccurs="1" >
     <xs:complexType>
       <xs:sequence>
                                <xs:element name="Setting" minOccurs="0"
maxOccurs="unbounded">
                                         <xs:complexType>
                                                  <xs:sequence>
                                                           <xs:element name="customerId"
```

```
minOccurs="1">
                                                    <xs:simpleType>
                                                         <xs:restriction
base="xs:string">
                                                              <xs:minLength
value="1"/>
                                                         </xs:restriction>
                                                    </xs:simpleType>
                                               </xs:element>
                                               <xs:element name="s_key"
minOccurs="1">
                                                    <xs:simpleType>
                                                         <xs:restriction
base="xs:string">
                                                              <xs:minLength
value="1"/>
                                                         </xs:restriction>
                                                    </xs:simpleType>
                                               </xs:element>
                                               <xs:element name="s_value"
minOccurs="1">
                                                    <xs:simpleType>
                                                         <xs:restriction
base="xs:string">
                                                              <xs:minLength
value="1"/>
                                                         </xs:restriction>
                                                    </xs:simpleType>
                                               </xs:element>
                                          </xs:sequence>
                                     </xs:complexType>
                                </xs:element>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
```

complexType **SFEvents_Type**

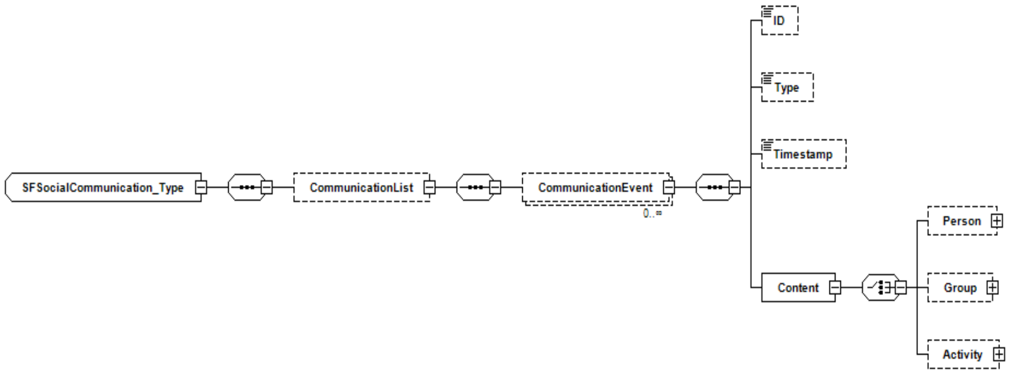
diagram	 SFEvents EventList Event 0..∞ ID Type TimeStamp Space SourceID Location Content Measurement Alert Maintenance ReAdaptation ARevent Training Recording Gesture Presence Gear ARloggingSession
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/
children	Measurement, Alert, Maintenance, ReAdaptation, ARevent, Training, Recording, Gesture, Presence, Gear, ARloggingSession

source	<xs:complexType name="SFEvents_Type">
	  <xs:sequence>
	    <xs:element name="EventList" maxOccurs="1" minOccurs="0">
	      <xs:complexType>
	        <xs:sequence>
	          <xs:element name="Event" maxOccurs="unbounded" minOccurs="0">
	            <xs:complexType>
	            <xs:sequence>
	              <xs:element type="xs:string" name="ID" minOccurs="0" maxOccurs="1"/>
	              <xs:element type="xs:string" name="Type" minOccurs="0" maxOccurs="1"/>
	              <xs:element type="xs:dateTime" name="TimeStamp" minOccurs="0" maxOccurs="1"/>
	              <xs:element type="xs:string" name="Space" minOccurs="0" maxOccurs="1"/>
	              <xs:element type="xs:string" name="SourceID" minOccurs="0" maxOccurs="1"/>
	            <xs:element type="SFcommon:tPosition" name="Location" minOccurs="0"

```
maxOccurs="1"/>
      <xs:element name="Content" minOccurs="0" maxOccurs="1">
       <xs:complexType>
        <xs:choice minOccurs="1" maxOccurs="1">
         <xs:element type="B2MML:OpSegmentDataType" name="Measurement"
minOccurs="0" maxOccurs="1"/>
         <xs:element type="B2MML:WorkAlertInformationType" name="Alert"
minOccurs="0" maxOccurs="1"/>
         <xs:element name="Maintenance" minOccurs="0" maxOccurs="1">
          <xs:complexType>
             <xs:sequence>
              <xs:element name="JobsList" minOccurs="0" maxOccurs="1" >
               <xs:complexType>
                <xs:sequence>
                 <xs:element type="B2MML:JobListType" name="Job"
minOccurs="0" maxOccurs="unbounded" />
                </xs:sequence>
               </xs:complexType>
              </xs:element>
              <xs:element type="B2MML:WorkScheduleType"
name="WorkScheduleList" minOccurs="0" maxOccurs="1" />
             </xs:sequence>
            </xs:complexType>
           </xs:element>
           <xs:element name="ReAdaptation" minOccurs="0" maxOccurs="1">
             <xs:complexType>
              <xs:sequence>
               <xs:element name="JobsList" minOccurs="0" maxOccurs="1" >
                <xs:complexType>
                   <xs:sequence>
                    <xs:element type="B2MML:JobListType" name="Job"
minOccurs="0" maxOccurs="unbounded" />
                    <xs:element name="NewJob"
type="B2MML:WorkRequestType" minOccurs="0" maxOccurs="1"/>
                   </xs:sequence>
                 </xs:complexType>
                </xs:element>
              </xs:sequence>
             </xs:complexType>
            </xs:element>
            <xs:element type="R3D:root" name="ARevent" minOccurs="0"
maxOccurs="1"/>
            <xs:element type="SCORM:TrainingType" name="Training"
minOccurs="0" maxOccurs="1"/>
            <xs:element type="SAS:RecordingType" name="Recording"
minOccurs="0" maxOccurs="1"/>
            <xs:element type="SAS:GestureType" name="Gesture" minOccurs="0"
maxOccurs="1"/>
```

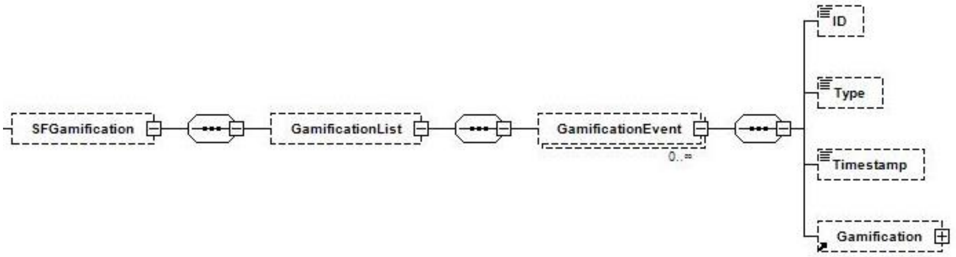
	<xs:element type="SAS:PresenceType" name="Presence" minOccurs="0" maxOccurs="1"/> <xs:element type="SAS:GearType" name="Gear" minOccurs="0" maxOccurs="1"/> <xs:element type="ARlog:session" name="ARloggingSession" minOccurs="0" maxOccurs="1"/> </xs:choice> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>

complexType **SFSocialCommunication_Type**

diagram	 SFSocialCommunication_Type CommunicationList CommunicationEvent 0..∞ ID Type Timestamp Content Person Group Activity
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/
children	Person, Group, Activity
source	<xs:complexType name="SFSocialCommunication_Type"> <xs:sequence> <xs:element name="CommunicationList" maxOccurs="1" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element name="CommunicationEvent" maxOccurs="unbounded" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element type="xs:string" name="ID" minOccurs="0" maxOccurs="1"/> <xs:element type="xs:string" name="Type" minOccurs="0" maxOccurs="1"/>

	<xs:element type="xs:dateTime" name="Timestamp" minOccurs="0" maxOccurs="1"/> <xs:element name="Content"> <xs:complexType> <xs:choice minOccurs="1" maxOccurs="1"> <xs:element type="OPENSOCIAL:Person" name="Person" minOccurs="0"/> <xs:element type="OPENSOCIAL:Group" name="Group" minOccurs="0"/> <xs:element type="OPENSOCIAL:Activity" name="Activity" minOccurs="0"/> </xs:choice> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>

complexType **SFGamification_Type**

diagram	 SFGamification GamificationList GamificationEvent 0..∞ ID Type Timestamp Gamification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/
children	ID, Type, Timstamp, Gamification
source	<xs:complexType name="SFGamification_Type"> <xs:sequence> <xs:element name="GamificationList" minOccurs="0" maxOccurs="1"> <xs:complexType> <xs:sequence> <xs:element name="GamificationEvent" maxOccurs="unbounded" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element type="xs:string" name="ID" minOccurs="0" maxOccurs="1"/> <xs:element type="xs:string" name="Type"

	minOccurs="0" maxOccurs="1"/> <xs:element type="xs:dateTime" name="Timestamp" minOccurs="0" maxOccurs="1"/> <xs:element ref="GAMIFICATION:Gamification" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>

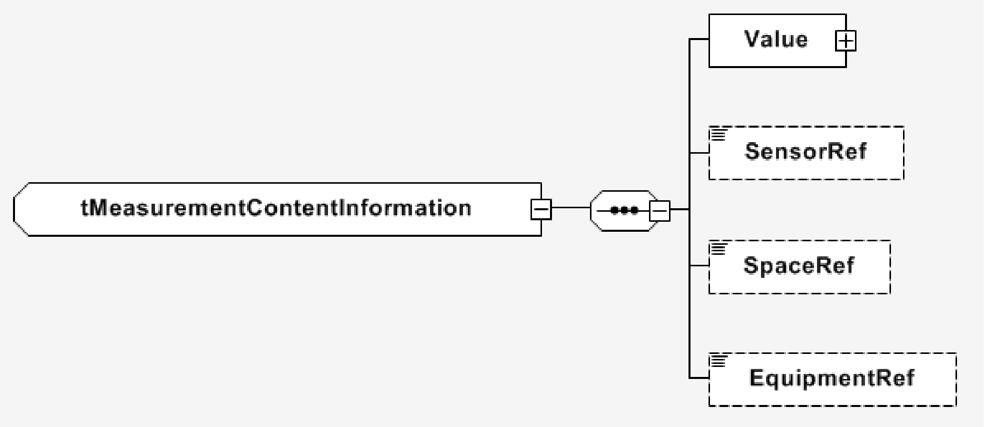
SCHEMA SATISFACTORY-COMMON.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/common**

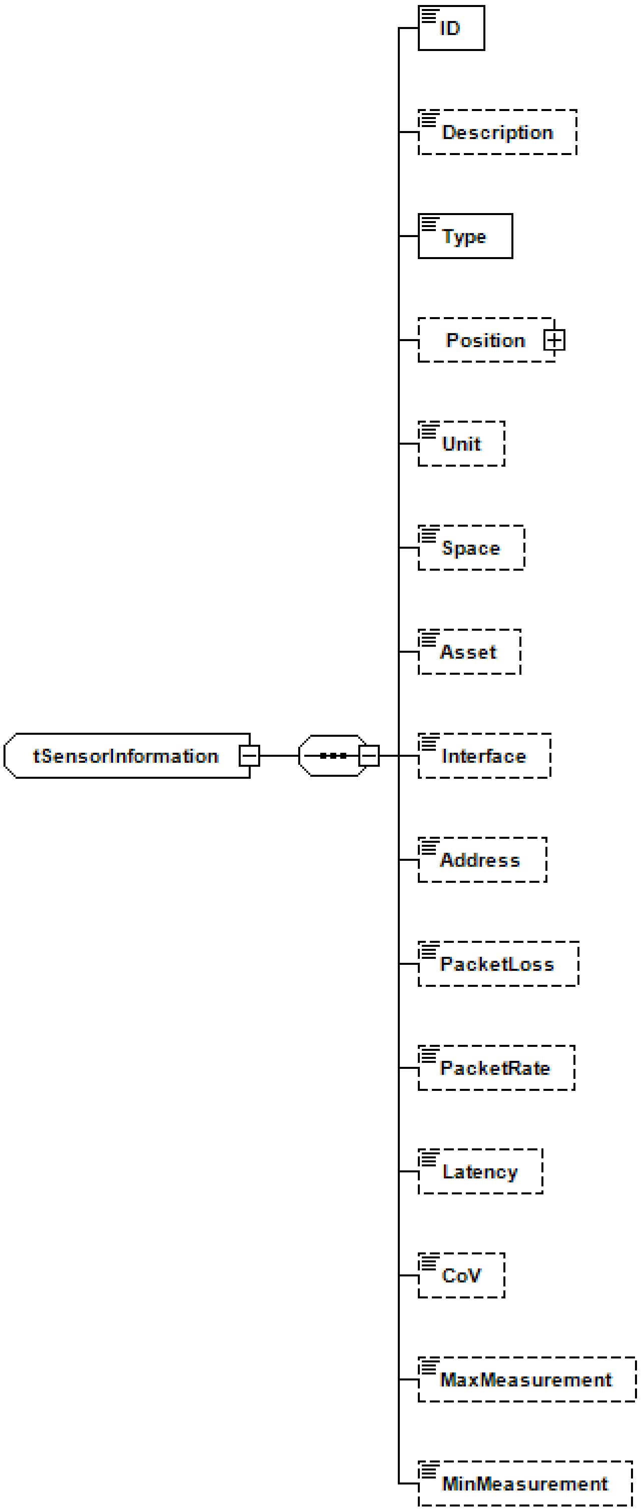
Elements	Complex Types	Simple Types
	TopicInformation	
	RelatedTopics	
	m3Config	
	gvrmConfig	
	cmmnConfig	
	tMeasurementContentInformation	
	tSensorInformation	tLengthUnitEnum
	tPosition	tValueUnit
	tValue	tInputFormat
	tCredentials	
	tSSL	
	tPersistence	
	tStore	
	tEvents	
	tUpload	
	tMode	
	tNotifier	
	tTopic	
	tStream	
	tEncoder	
	tSupporterRunTime	
	tAdd	

complexType **tMeasurementContentInformation**

diagram	 Value SensorRef SpaceRef EquipmentRef tMeasurementContentInformation
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
children	Value, SensorRef, SpaceRef, EquipmentRef

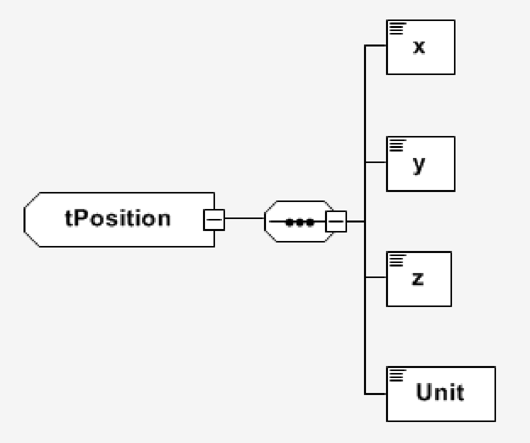
source	<xs:complexType name="tMeasurementContentInformation">
	    <xs:sequence>
	        <xs:element type="SFcommon:tValue" name="Value"/>
	        <xs:element type="xs:string" name="SensorRef" minOccurs="0"/>
	        <xs:element type="xs:string" name="SpaceRef" minOccurs="0"/>
	        <xs:element type="xs:string" name="EquipmentRef" minOccurs="0"/>
	    </xs:sequence>
	</xs:complexType>

complexType **tSensorInformation**

diagram	 tSensorInformation ID Description Type Position Unit Space Asset Interface Address PacketLoss PacketRate Latency CoV MaxMeasurement MinMeasurement
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

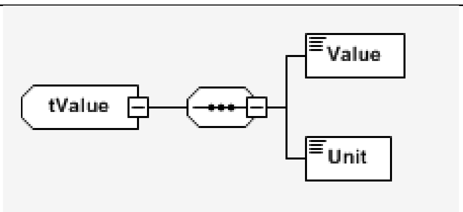
source	<xs:complexType name="tSensorInformation">
	  <xs:sequence>
	    <xs:element type="xs:string" name="ID"/>
	    <xs:element type="xs:string" name="Description" minOccurs="0"/>
	    <xs:element type="SFcommon:tSensorType" name="Type"/>
	    <xs:element type="SFcommon:tPosition" name="Position" minOccurs="0"/>
	    <xs:element type="xs:string" name="Unit" minOccurs="0"/>
	    <xs:element type="xs:string" name="Space" minOccurs="0"/>
	    <xs:element type="xs:string" name="Asset" minOccurs="0"/>
	    <xs:element type="xs:string" name="Interface" minOccurs="0"/>
	    <xs:element type="xs:string" name="Address" minOccurs="0"/>
	    <xs:element type="xs:string" name="PacketLoss" minOccurs="0"/>
	    <xs:element type="xs:string" name="PacketRate" minOccurs="0"/>
	    <xs:element type="xs:string" name="Latency" minOccurs="0"/>
	    <xs:element type="xs:float" name="CoV" minOccurs="0"/>
	    <xs:element type="xs:float" name="MaxMeasurement" minOccurs="0"/>
	    <xs:element type="xs:float" name="MinMeasurement" minOccurs="0"/>
	  </xs:sequence>
	</xs:complexType>

complexType **tPosition**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

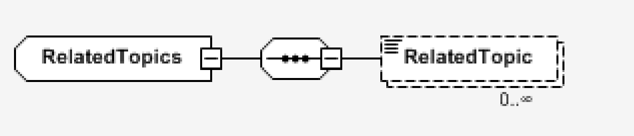
source	<xs:complexType name="tPosition">
	  <xs:sequence>
	    <xs:element type="xs:float" name="x"/>
	    <xs:element type="xs:float" name="y"/>
	    <xs:element type="xs:float" name="z"/>
	    <xs:element type="SFcommon:tLengthUnitEnum" name="Unit"/>
	  </xs:sequence>
	</xs:complexType>

complexType **tValue**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

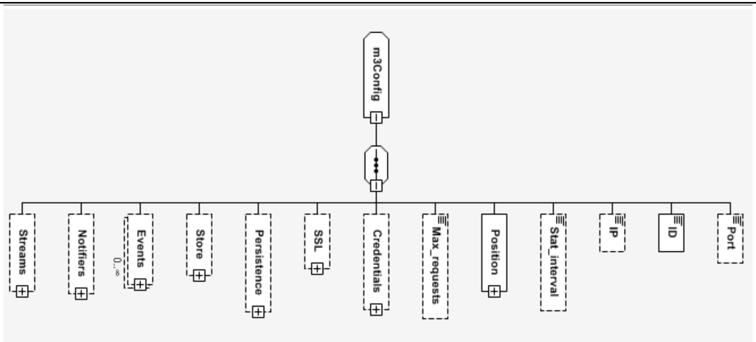
source	<xs:complexType name="tValue">
	    <xs:sequence>
	        <xs:element type="xs:float" name="Value"/>
	        <xs:element type="SFcommon:tValueUnit" name="Unit"/>
	    </xs:sequence>
	</xs:complexType>

complexType **RelatedTopics**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

source	<xs:complexType name="RelatedTopics">
	    <xs:sequence>
	        <xs:element type="xs:string" name="RelatedTopic" minOccurs="0" maxOccurs = "unbounded"/>
	    </xs:sequence>
	</xs:complexType>

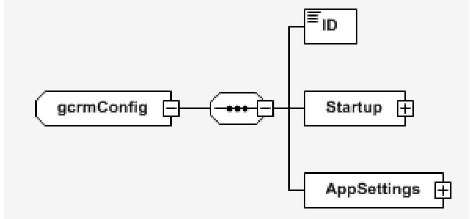
complexType **m3Config**

diagram	 m3Config Streams Notifiers 0..∞ Events Store Persistence SSL Credentials Max_requests Position Stat_interval IP ID Port
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

source	<xs:complexType name="m3Config">
	    <xs:sequence>
	        <xs:element type="xs:integer" name="Port" default="2345" minOccurs="0"/>
	        <xs:element type="xs:string" name="ID" minOccurs="1"/>
	        <xs:element type="xs:string" name="IP" minOccurs="0"/>
	        <xs:element type="xs:string" name="Stat_interval" minOccurs="0"/>
	        <xs:element type="SFcommon:tPosition" name="Position" minOccurs="1"/>

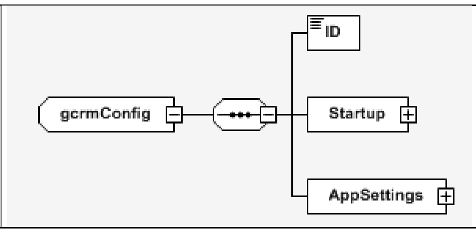
	<xs:element type="xs:integer" name="Max_requests" minOccurs="0"/> <xs:element type="SFcommon:tCredentials" name="Credentials" minOccurs="0"/> <xs:element type="SFcommon:tSSL" name="SSL" minOccurs="0"/> <xs:element type="SFcommon:tPersistence" name="Persistence" minOccurs="0"/> <xs:element type="SFcommon:tStore" name="Store" minOccurs="0"/> <xs:element type="SFcommon:tEvents" name="Events" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="Notifiers" minOccurs="0" maxOccurs="1" > <xs:complexType> <xs:sequence> <xs:element type="SFcommon:tNotifier" name="Notifier" minOccurs="1" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="Streams" minOccurs="0" maxOccurs="1" > <xs:complexType> <xs:sequence> <xs:element type="SFcommon:tStream" name="Stream" minOccurs="1" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>

complexType **gcrmConfig**

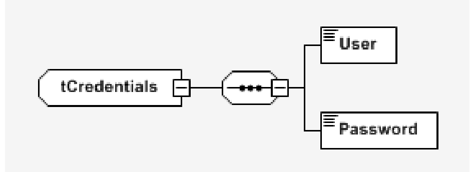
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="gcrmConfig"> <xs:sequence> <xs:element type="xs:string" name="ID" minOccurs="1"/> <xs:element name="Startup" minOccurs="1"> <xs:complexType> <xs:sequence> <xs:element type="SFcommon:tSupportedRuntime" name="SupportedRuntime" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element>

	<xs:element name="AppSettings" minOccurs="1"> <xs:complexType> <xs:sequence> <xs:element type="SFcommon:tAdd" name="Add" minOccurs="0" maxOccurs="unbounded" nillable="true"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>

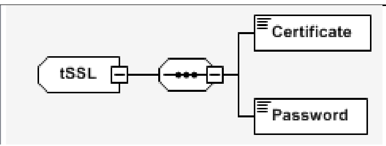
complexType **cmmnConfig**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="cmmnConfig"> <xs:sequence> <xs:element type="xs:string" name="ID" minOccurs="1"/> <xs:element type="xs:string" name="ConfigurationFile" minOccurs="1"/> </xs:sequence> </xs:complexType>

complexType **tCredentials**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tCredentials"> <xs:sequence> <xs:element type="xs:string" name="User" minOccurs="1"/> <xs:element type="xs:string" name="Password" minOccurs="1"/> </xs:sequence> </xs:complexType>

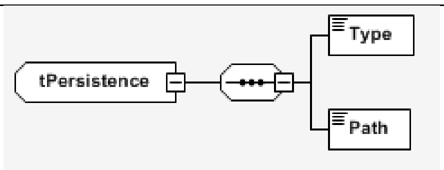
complexType **tSSL**

diagram	

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

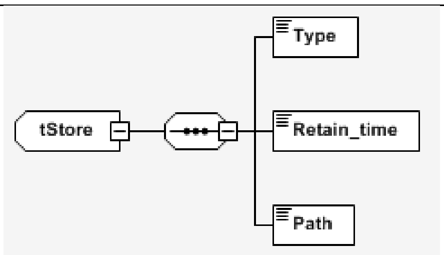
source	<xs:complexType name="tSSL">
	  <xs:sequence>
	    <xs:element type="xs:string" name="Certificate" minOccurs="1"/>
	    <xs:element type="xs:string" name="Password" minOccurs="1"/>
	  </xs:sequence>
	</xs:complexType>

complexType **tPersistence**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

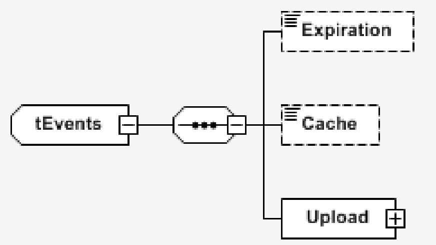
source	<xs:complexType name="tPersistence">
	  <xs:sequence>
	    <xs:element type="xs:string" name="Type" default="SQLite" minOccurs="1"/>
	    <xs:element type="xs:string" name="Path" minOccurs="1"/>
	  </xs:sequence>
	</xs:complexType>

complexType **tStore**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

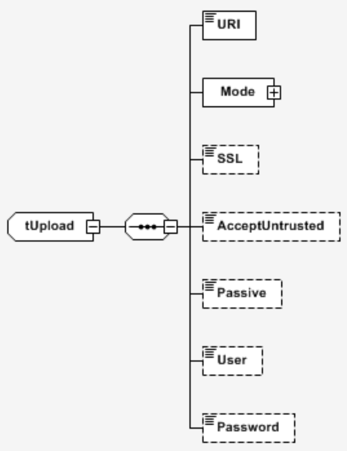
source	<xs:complexType name="tStore">
	  <xs:sequence>
	    <xs:element type="xs:string" name="Type" minOccurs="1"/>
	    <xs:element type="xs:string" name="Retain_time" minOccurs="1"/>
	    <xs:element type="xs:string" name="Path" minOccurs="1"/>
	  </xs:sequence>
	  <xs:attribute name="enabled" type="xs:boolean"/>
	</xs:complexType>

complexType **tEvents**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

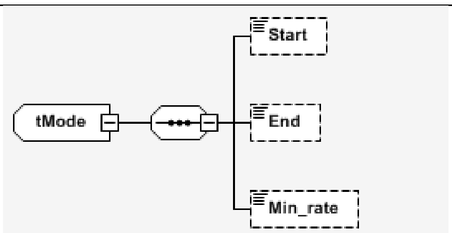
source	<xs:complexType name="tEvents">
	    <xs:sequence>
	        <xs:element type="xs:string" name="Expiration" default="12" minOccurs="0"/>
	        <xs:element type="xs:string" name="Cache" default="2" minOccurs="0"/>
	        <xs:element type="SFcommon:tUpload" name="Upload" minOccurs="1"/>
	    </xs:sequence>
	    <xs:attribute name="enabled" type="xs:boolean"/>
	</xs:complexType>

complexType **tUpload**

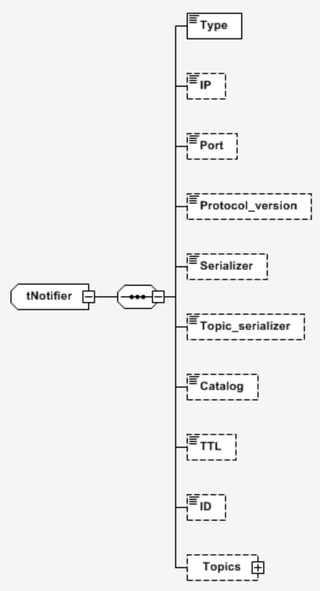
diagram	 tUpload URI Mode SSL AcceptUntrusted Passive User Password
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common

source	<xs:complexType name="tUpload">
	    <xs:sequence>
	        <xs:element type="xs:string" name="URI" minOccurs="1"/>
	        <xs:element type="SFcommon:tMode" name="Mode" minOccurs="1"/>
	        <xs:element type="xs:boolean" name="SSL" default="false" minOccurs="0"/>
	        <xs:element type="xs:boolean" name="AcceptUntrusted" default="false" minOccurs="0"/>
	        <xs:element type="xs:boolean" name="Passive" default="true" minOccurs="0"/>
	        <xs:element type="xs:string" name="User" default="" minOccurs="0"/>
	        <xs:element type="xs:string" name="Password" default="" minOccurs="0"/>
	    </xs:sequence>
	    <xs:attribute name="enabled" type="xs:boolean"/>
	</xs:complexType>

complexType **tMode**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tMode"> <xs:sequence> <xs:element type="xs:string" name="Start" minOccurs="0"/> <xs:element type="xs:string" name="End" minOccurs="0"/> <xs:element type="xs:string" name="Min_rate" default="0" minOccurs="0"/> </xs:sequence> </xs:complexType>

complexType **tNotifier**

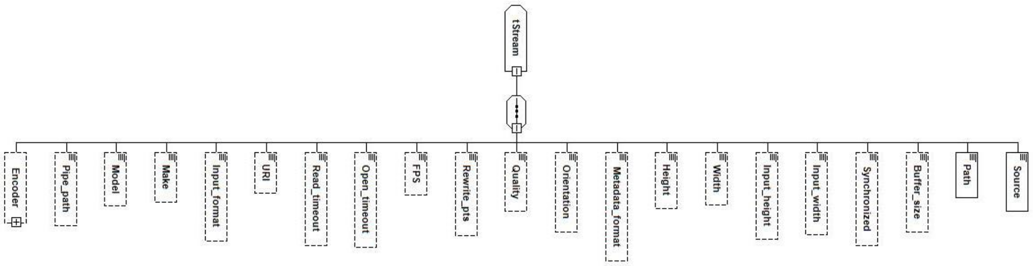
diagram	 tNotifier Type IP Port Protocol_version Serializer Topic_serializer Catalog TTL ID Topics
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tNotifier"> <xs:sequence> <xs:element type="xs:string" name="Type" minOccurs="1"/> <xs:element type="xs:string" name="IP" minOccurs="0"/> <xs:element type="xs:string" name="Port" minOccurs="0"/> <xs:element type="xs:string" name="Protocol_version" default="3.1.1" minOccurs="0"/> <xs:element type="xs:string" name="Serializer" minOccurs="0"/> <xs:element type="xs:string" name="Topic_serializer" minOccurs="0"/> <xs:element type="xs:string" name="Catalog" minOccurs="0"/> <xs:element type="xs:string" name="TTL" minOccurs="0"/>

	<xs:element type="xs:string" name="ID" minOccurs="0"/> <xs:element name="Topics" minOccurs="0" maxOccurs="1" > <xs:complexType> <xs:sequence> <xs:element type="SFcommon:tTopic" name="Topic" minOccurs="1" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> <xs:attribute name="enabled" type="xs:boolean"/> </xs:complexType>

complexType **tTopic**

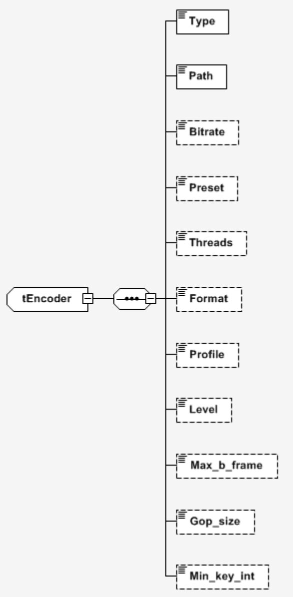
diagram	-
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tTopic"> <xs:attribute type="xs:string" name="type"/> <xs:attribute type="xs:string" name="mode"/> <xs:attribute type="xs:string" name="retain"/> <xs:attribute type="xs:string" name="qos"/> <xs:attribute type="xs:string" name="path"/> <xs:attribute type="xs:string" name="catalog"/> <xs:attribute type="xs:string" name="resourceId"/> </xs:complexType>

complexType **tStream**

diagram	 tStream Encoder Pipe_path Model Make Input_format URI Read_timeout Open_timeout FPS Rewrite_pts Quality Orientation Metadata_format Height Width Input_height Input_width Synchronized Buffer_size Path Source
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tStream"> <xs:sequence> <xs:element type="xs:string" name="Source" minOccurs="1"/> <xs:element type="xs:string" name="Path" minOccurs="1"/> <xs:element type="xs:string" name="Buffer_size" default="0" minOccurs="0"/> <xs:element type="xs:boolean" name="Synchronized" default="false" minOccurs="0"/> <xs:element type="xs:string" name="Input_width" minOccurs="0"/> <xs:element type="xs:string" name="Input_height" minOccurs="0"/> <xs:element type="xs:string" name="Width" minOccurs="0"/>

	<xs:element type="xs:string" name="Height" minOccurs="0"/> <xs:element type="xs:string" name="Metadata_format" default="none" minOccurs="0"/> <xs:element type="xs:string" name="Orientation" default="1" minOccurs="0"/> <xs:element type="xs:string" name="Quality" default="100" minOccurs="0"/> <xs:element type="xs:string" name="Rewrite_pts" minOccurs="0"/> <xs:element type="xs:string" name="FPS" minOccurs="0"/> <xs:element type="xs:string" name="Open_timeout" default="20000" minOccurs="0"/> <xs:element type="xs:string" name="Read_timeout" default="1000" minOccurs="0"/> <xs:element type="xs:string" name="URI" minOccurs="0"/> <xs:element type="SFcommon:tInputFormat" name="Input_format" minOccurs='0'/> <xs:element type="xs:string" name="Make" minOccurs="0"/> <xs:element type="xs:string" name="Model" minOccurs="0"/> <xs:element type="xs:string" name="Pipe_path" minOccurs="0"/> <xs:element type="SFcommon:tEncoder" name="Encoder" minOccurs="0"/> </xs:sequence> <xs:attribute name="enabled" type="xs:boolean"/> </xs:complexType>

complexType **tEncoder**

diagram	 tEncoder Type Path Bitrate Preset Threads Format Profile Level Max_b_frame Gop_size Min_key_int
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tEncoder"> <xs:sequence> <xs:element type="xs:string" name="Type" minOccurs="1"/> <xs:element type="xs:string" name="Path" minOccurs="1"/> <xs:element type="xs:string" name="Bitrate" default="2000000" minOccurs="0"/>

	<xs:element type="xs:string" name="Preset" default="fast" minOccurs="0"/> <xs:element type="xs:string" name="Threads" default="1" minOccurs="0"/> <xs:element type="xs:string" name="Format" default="avi" minOccurs="0"/> <xs:element type="xs:string" name="Profile" default="high" minOccurs="0"/> <xs:element type="xs:string" name="Level" default="3.1" minOccurs="0"/> <xs:element type="xs:string" name="Max_b_frame" default="2" minOccurs="0"/> <xs:element type="xs:string" name="Gop_size" default="50" minOccurs="0"/> <xs:element type="xs:string" name="Min_key_int" default="25" minOccurs="0"/> </xs:sequence> <xs:attribute name="enabled" type="xs:boolean"/> </xs:complexType>

complexType **tSupportedRunTime**

diagram	-
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tSupportedRuntime"> <xs:attribute name="version" type="xs:string"/> <xs:attribute name="sku" type="xs:string"/> </xs:complexType>

complexType **tAdd**

diagram	-
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
source	<xs:complexType name="tAdd"> <xs:attribute name="key" type="xs:string"/> <xs:attribute name="value" type="xs:string"/> </xs:complexType>

simpleType **tSensorType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string
source	<xs:simpleType name="tSensorType" final="restriction"> <xs:restriction base="xs:string"> <xs:enumeration value="DepthCamera"/> <xs:enumeration value="ThermalCamera"/> <xs:enumeration value="Accelerometer"/> <xs:enumeration value="Gyroscope"/> <xs:enumeration value="CardioSensor"/> <xs:enumeration value="TempSensor"/> <xs:enumeration value="Thermocouple"/> <xs:enumeration value="PWM"/> <xs:enumeration value="Controller SetPoint"/> <xs:enumeration value="Counter"/> <xs:enumeration value="Totalizer"/> <xs:enumeration value="Pressure"/>

	<xs:enumeration value="Calculation"/> <xs:enumeration value="Controller SetPointOutput"/> <xs:enumeration value="RFID"/> <xs:enumeration value="Unknown"/> </xs:restriction> </xs:simpleType>

simpleType **tLengthUnitEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string
source	<xs:simpleType name="tLengthUnitEnum"> <xs:restriction base="xs:string"> <xs:enumeration value="Kilometres"/> <xs:enumeration value="Meters"/> <xs:enumeration value="Centimetres"/> <xs:enumeration value="Millimetres"/> </xs:restriction> </xs:simpleType>

simpleType **tValueUnit**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string
source	<xs:simpleType name="tValueUnit"> <xs:restriction base="xs:string"> <xs:enumeration value="Celsius"/> <xs:enumeration value="Percentage"/> </xs:restriction> </xs:simpleType>

simpleType **tInputFormat**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string
source	<xs:simpleType name="tInputFormat"> <xs:restriction base="xs:string"> <xs:enumeration value="RGB24"/> <xs:enumeration value="BGR24"/> <xs:enumeration value="RGB32"/> <xs:enumeration value="BGR32"/> <xs:enumeration value="ARGB"/> <xs:enumeration value="ABGR"/> <xs:enumeration value="RGBA"/> <xs:enumeration value="BGRA"/> <xs:enumeration value="GRAY8"/> <xs:enumeration value="GRAY16"/> </xs:restriction> </xs:simpleType>

SCHEMA SATISFACTORY_SASEVENTS.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events**

Elements

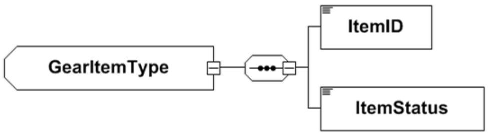
Complex Types

- GearItemType
- GearType
- PresenceType
- RecordingResultsType
- RecordingType

Simple Types

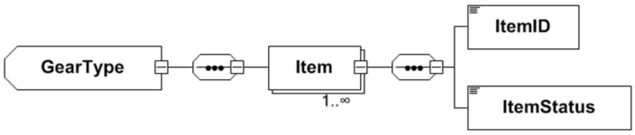
- GearItemIdType
- GearItemStatusType
- GestureType
- RecordingStatusType

complexType **GearItemType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events
children	ItemID, ItemStatus

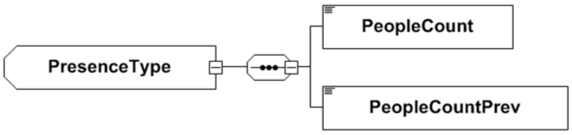
source	<xs:complexType name="GearItemType">
	    <xs:sequence>
	        <xs:element name="ItemID" type="GearItemIdType"/>
	        <xs:element name="ItemStatus" type="GearItemStatusType"/>
	    </xs:sequence>
	    <xs:attribute name="Changed" type="xs:boolean"/>
	</xs:complexType>

complexType **GearType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events
children	Item

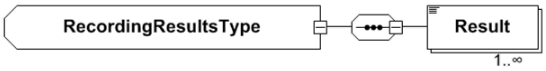
source	<xs:complexType name="GearType">
	    <xs:sequence>
	        <xs:element name="Item" type="GearItemType" maxOccurs="unbounded"/>
	    </xs:sequence>
	</xs:complexType>

complexType **PresenceType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events

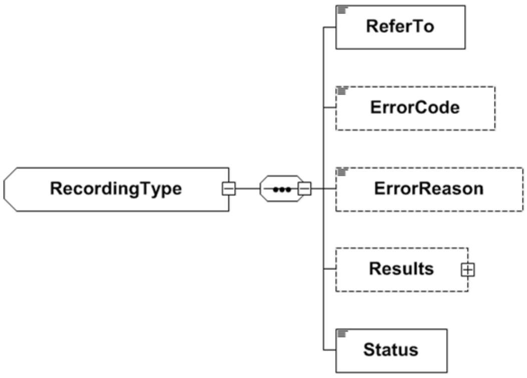
source	<xs:complexType name="PresenceType">
	    <xs:sequence>
	        <xs:element name="PeopleCount" type="xs:integer"/>
	        <xs:element name="PeopleCountPrev" type="xs:integer"/>
	    </xs:sequence>
	</xs:complexType>

complexType **RecordingResultsType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events

source	<xs:complexType name="RecordingResultsType">
	    <xs:sequence>
	        <xs:element name="Result" type="xs:string" maxOccurs="unbounded"/>
	    </xs:sequence>
	</xs:complexType>

complexType **RecordingType**

diagram	 RecordingType ReferTo ErrorCode ErrorReason Results Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/sas/events

source	<xs:complexType name="RecordingType">
	    <xs:sequence>
	        <xs:element name="ReferTo" type="xs:string"/>
	        <xs:element name="ErrorCode" type="xs:integer" minOccurs="0"/>
	        <xs:element name="ErrorReason" type="xs:string" minOccurs="0"/>
	        <xs:element name="Results" type="RecordingResultsType" minOccurs="0"/>
	        <xs:element name="Status" type="RecordingStatusType"/>
	    </xs:sequence>
	</xs:complexType>

simpleType **GearItemIdType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string

source	<xs:simpleType name="GearItemIdType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="Helmet"/>
	    <xs:enumeration value="Jacket"/>
	    <xs:enumeration value="LeftGlove"/>
	    <xs:enumeration value="RightGlove"/>
	  </xs:restriction>
	</xs:simpleType>

simpleType **GearItemStatusType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string

source	<xs:simpleType name="GearItemStatusType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="Present"/>
	    <xs:enumeration value="Missing"/>
	    <xs:enumeration value="Untracked"/>
	  </xs:restriction>
	</xs:simpleType>

simpleType **GestureType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string

source	<xs:simpleType name="GestureType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="LeftHandSwipeRight"/>
	    <xs:enumeration value="RightHandSwipeLeft"/>
	    <xs:enumeration value="LeftHandSwipeLeft"/>
	    <xs:enumeration value="RightHandSwipeRight"/>
	    <xs:enumeration value="BothHandsRaised"/>
	  </xs:restriction>
	</xs:simpleType>

simpleType **RecordingStatusType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/common
type	string

source	<xs:simpleType name="RecordingStatusType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="Error"/>
	    <xs:enumeration value="Ready"/>
	    <xs:enumeration value="Expired"/>
	  </xs:restriction>

	</xs:simpleType>

Schema gbXML_v5.12.xsd

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/gbXML**

It is a common XSD schema, which has been used without any intervension. Thus, the description of its XSD schema is omitted.

SCHEMA B2MML.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML**

Elements

EquipmentInformation
PersonnelInformation
PhysicalAssetInformation
ProcessSegmentInformation

Complex Types

AnyGenericValueType
AssemblyRelationship1Type
AssemblyRelationshipType
AssemblyType1Type
AssemblyTypeType
CodeType
DataType1Type
DataTypeType
DateTimeType
Dependency1Type
DependencyType
DescriptionType
EndTimeType
EquipmentAssetMappingType
EquipmentCapabilityTestSpecificationIDType
EquipmentCapabilityTestSpecificationType
EquipmentClassIDType
EquipmentClassPropertyType
EquipmentClassType
EquipmentElementLevel1Type
EquipmentElementLevelType
EquipmentIDType
EquipmentInformationType
EquipmentPropertyType
EquipmentSegmentSpecificationPropertyType
EquipmentSegmentSpecificationType
EquipmentType
EquipmentUseType
ExpirationTimeType
HierarchyScopeType
IdentifierType
JobOrderCommand1Type
JobOrderCommandRuleType
JobOrderCommandType
JobOrderDispatchStatusType
LocationType
MaterialClassIDType
MaterialDefinitionIDType
MaterialLotIDType
MaterialSegmentSpecificationPropertyType

Simple Types

DurationType

MaterialSegmentSpecificationType
MaterialSubLotIDType
MaterialUse1Type
MaterialUseType
NameType
NumericType
OpEquipmentRequirementPropertyType
OpEquipmentRequirementType
OpEquipmentSpecificationPropertyType
OpEquipmentSpecificationType
OperationsDefinitionIDType
OperationsType1Type
OperationsTypeType
OpMaterialRequirementPropertyType
OpMaterialRequirementType
OpMaterialSpecificationPropertyType
OpMaterialSpecificationType
OpPersonnelRequirementPropertyType
OpPersonnelRequirementType
OpPersonnelSpecificationPropertyType
OpPersonnelSpecificationType
OpPhysicalAssetRequirementPropertyType
OpPhysicalAssetRequirementType
OpPhysicalAssetSpecificationPropertyType
OpPhysicalAssetSpecificationType
OpSegmentDataType
ParameterType
PersonIDType
PersonNameType
PersonnelClassIDType
PersonnelClassPropertyType
PersonnelClassType
PersonnelInformationType
PersonnelSegmentSpecificationPropertyType
PersonnelSegmentSpecificationType
PersonnelUseType
PersonPropertyType
PersonType
PhysicalAssetCapabilityTestSpecificationIDType
PhysicalAssetCapabilityTestSpecificationType
PhysicalAssetClassIDType
PhysicalAssetClassPropertyType
PhysicalAssetClassType
PhysicalAssetIDType
PhysicalAssetInformationType
PhysicalAssetPropertyType
PhysicalAssetSegmentSpecificationPropertyType
PhysicalAssetSegmentSpecificationType

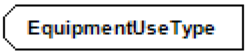
PhysicalAssetType
PhysicalAssetUseType
PriorityType
ProcessSegmentIDType
ProcessSegmentInformationType
ProcessSegmentType
ProductProductionRuleIDType
ProductSegmentIDType
PropertyIDType
PublishedDateType
QualificationTestSpecificationIDType
QualificationTestSpecificationType
QuantityStringType
QuantityValueType
RequestState1Type
RequestStateType
RequiredByRequestedSegmentResponse1Type
RequiredByRequestedSegmentResponseType
ResultType
SegmentDependencyType
StartTimeType
StorageLocationType
TestDateTimeType
TestedEquipmentClassPropertyType
TestedEquipmentPropertyType
TestedPersonnelClassPropertyType
TestedPersonPropertyType
TestedPhysicalAssetClassPropertyType
TestedPhysicalAssetPropertyType
TestResultType
TextType
UnitOfMeasureType
ValueStringType
ValueType
VersionType
WorkAlertDefinitionType
WorkAlertInformationType
WorkAlertPropertyType
WorkAlertType
WorkType1Type
WorkTypeType

element **EndTimeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

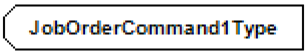
source	<xsd:complexType name="EndTimeType">
	    <xsd:simpleContent>
	        <xsd:restriction base="B2MML:DateTimeType"/>
	    </xsd:simpleContent>
	</xsd:complexType>

element **EquipmentUseType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

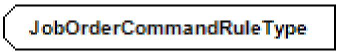
source	<xsd:complexType name="EquipmentUseType">
	    <xsd:simpleContent>
	        <xsd:restriction base="B2MML:CodeType"/>
	    </xsd:simpleContent>
	</xsd:complexType>

element **JobOrderCommand1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="JobOrderCommand1Type">
	    <xsd:simpleContent>
	        <xsd:restriction base="B2MML:CodeType">
	            <xsd:enumeration value="Start"/>
	            <xsd:enumeration value="Stop"/>
	            <xsd:enumeration value="Hold"/>
	            <xsd:enumeration value="Restart"/>
	            <xsd:enumeration value="Abort"/>
	            <xsd:enumeration value="Reset"/>
	            <xsd:enumeration value="Pause"/>
	            <xsd:enumeration value="Resume"/>
	            <xsd:enumeration value="Other"/>
	        </xsd:restriction>
	    </xsd:simpleContent>
	</xsd:complexType>

element **JobOrderCommandRuleType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="JobOrderCommandRuleType">
	    <xsd:simpleContent>
	        <xsd:restriction base="B2MML:TextType"/>

	</xsd:simpleContent> </xsd:complexType>

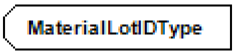
element **JobOrderCommandType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="JobOrderCommandType"> <xsd:simpleContent> <xsd:extension base="B2MML:JobOrderCommand1Type"> <xsd:attribute name="OtherValue" type="xsd:string"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

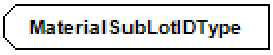
element **JobOrderDispatchStatusType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="JobOrderDispatchStatusType"> <xsd:simpleContent> <xsd:restriction base="B2MML:CodeType"/> </xsd:simpleContent> </xsd:complexType>

element **MaterialLotIDType**

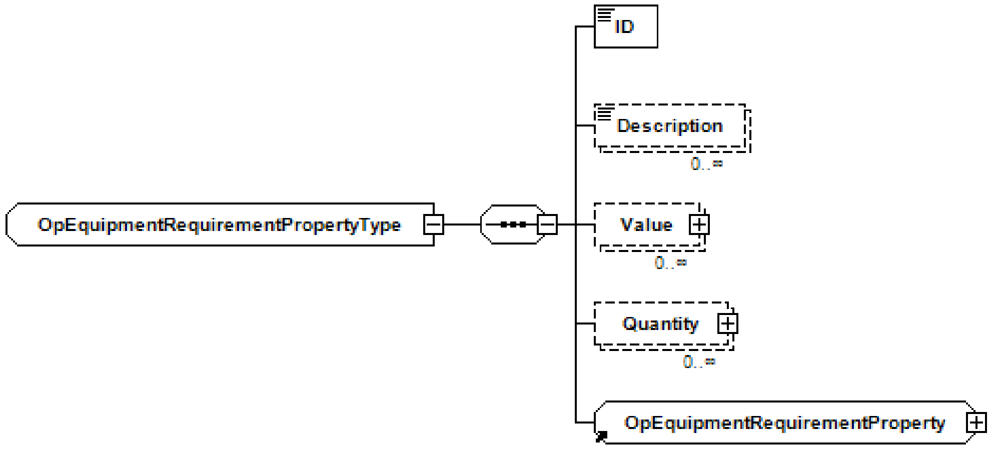
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="MaterialLotIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

element **MaterialSubLotIDType**

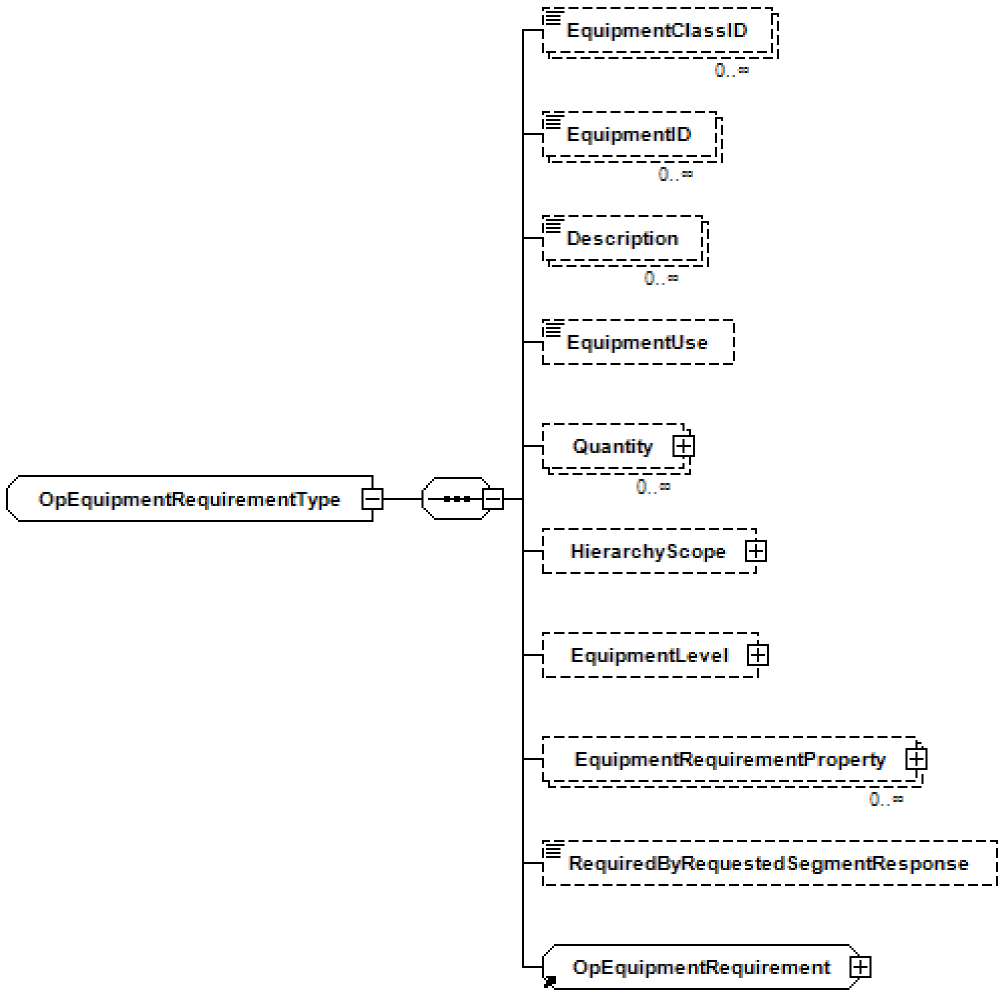
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="MaterialSubLotIDType"> <xsd:simpleContent>

	<xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

element **OpEquipmentRequirementPropertyType**

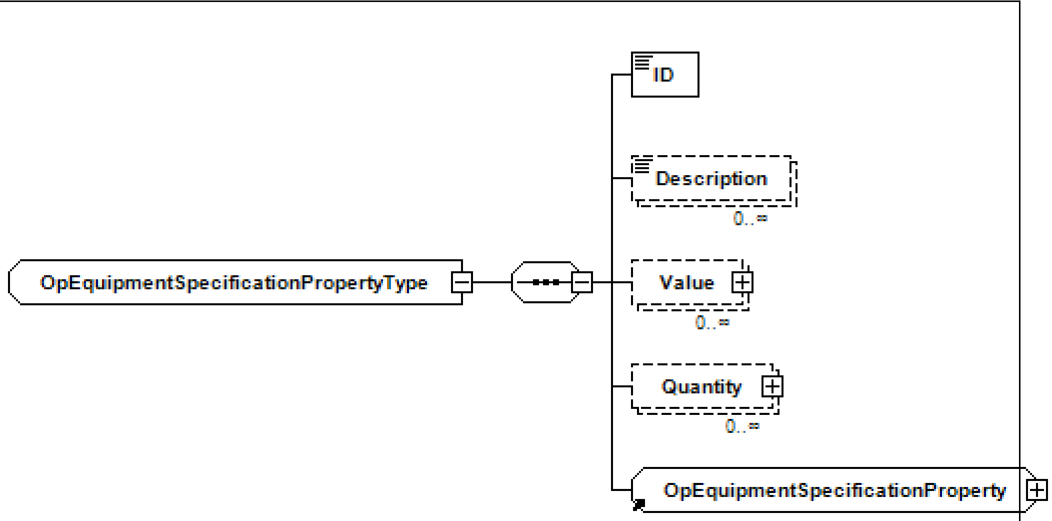
diagram	 OpEquipmentRequirementPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ OpEquipmentRequirementProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpEquipmentRequirementPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpEquipmentRequirementProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpEquipmentRequirementType**

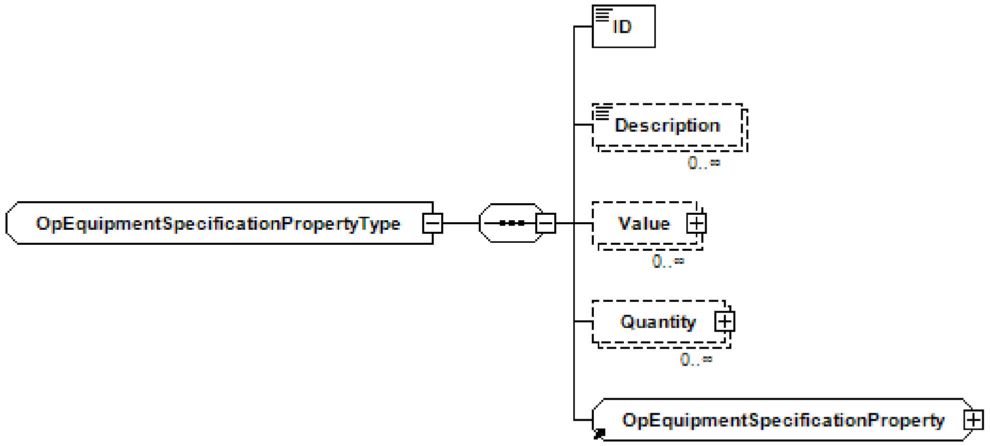
diagram	 OpEquipmentRequirementType EquipmentClassID 0..∞ EquipmentID 0..∞ Description 0..∞ EquipmentUse Quantity 0..∞ HierarchyScope EquipmentLevel EquipmentRequirementProperty 0..∞ RequiredByRequestedSegmentResponse OpEquipmentRequirement
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpEquipmentRequirementType"> <xsd:sequence> <xsd:element name="EquipmentClassID" type="B2MML:EquipmentClassIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentUse" type="B2MML:EquipmentUseType" minOccurs="0"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>

	<xsd:element name="EquipmentLevel" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="EquipmentRequirementProperty" type="B2MML:OpEquipmentRequirementPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="RequiredByRequestedSegmentResponse" type="B2MML:RequiredByRequestedSegmentResponseType" minOccurs="0"/> <xsd:group ref="B2MML:OpEquipmentRequirement" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpEquipmentSpecificationPropertyType**

diagram	 OpEquipmentSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ OpEquipmentSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpEquipmentSpecificationPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpEquipmentSpecificationProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpEquipmentSpecificationType**

diagram	 OpEquipmentSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ OpEquipmentSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

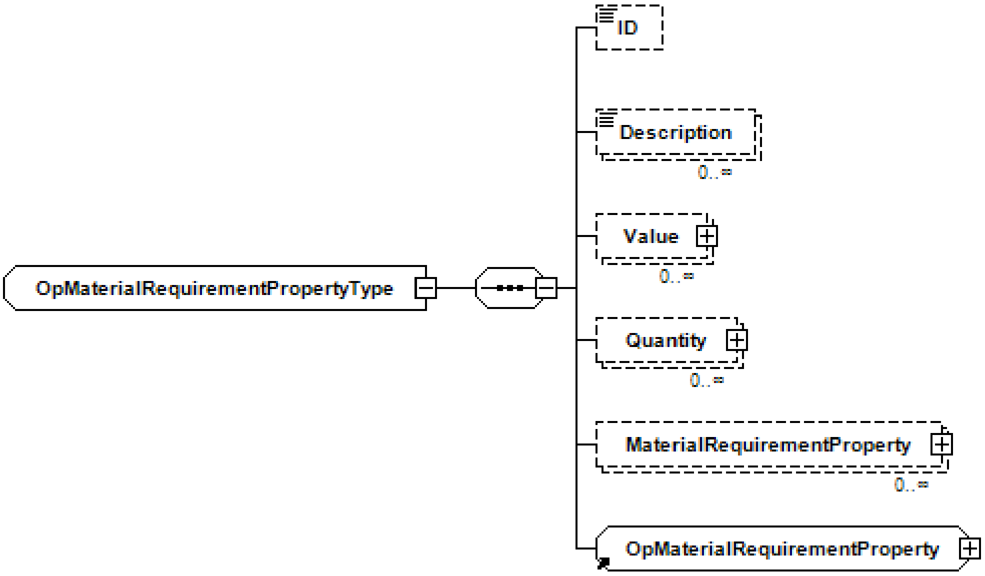
source	<xsd:complexType name="OpEquipmentSpecificationType">
	    <xsd:sequence>
	        <xsd:element name="EquipmentClassID" type="B2MML:EquipmentClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="EquipmentUse" type="B2MML:EquipmentUseType" minOccurs="0"/>
	        <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="EquipmentSpecificationProperty" type="B2MML:OpEquipmentSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:group ref="B2MML:OpEquipmentSpecification" minOccurs="0" maxOccurs="1"/>
	    </xsd:sequence>
	</xsd:complexType>

element **OperationsDefinitionIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

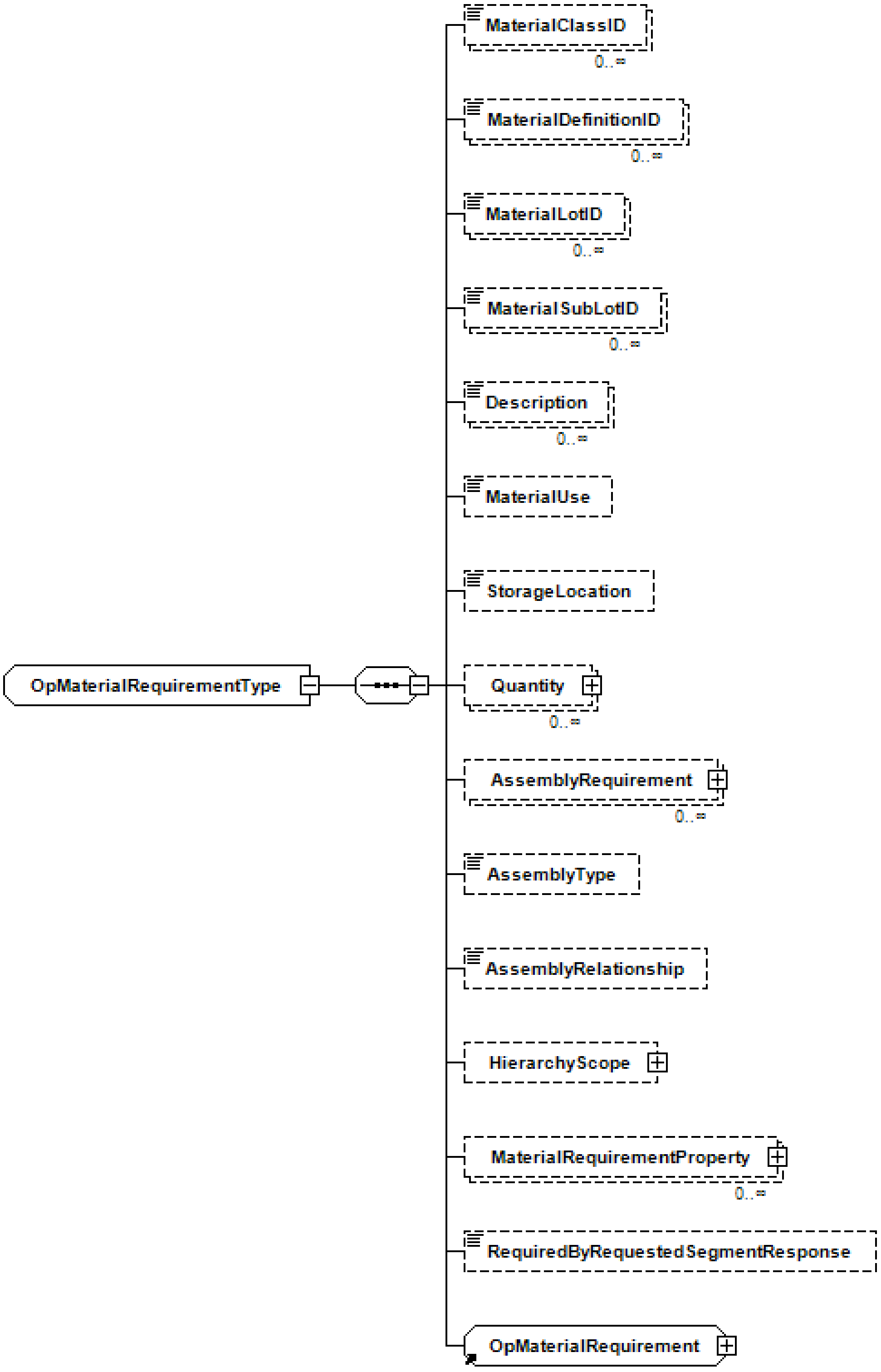
source	<xsd:complexType name="OperationsDefinitionIDType">
	    <xsd:simpleContent>
	        <xsd:restriction base="B2MML:IdentifierType"/>
	    </xsd:simpleContent>
	</xsd:complexType>

element **OpMaterialRequirementPropertyType**

diagram	 OpMaterialRequirementPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ MaterialRequirementProperty 0..∞ OpMaterialRequirementProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

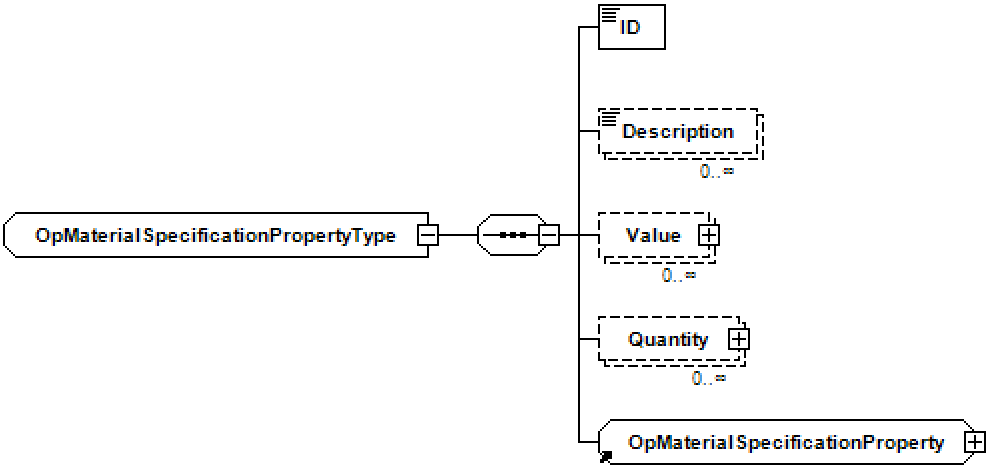
source	<xsd:complexType name="OpMaterialRequirementPropertyType">
	    <xsd:sequence>
	      <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/>
	      <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	      <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/>
	      <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	      <xsd:element name="MaterialRequirementProperty" type="B2MML:OpMaterialRequirementPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	      <xsd:group ref="B2MML:OpMaterialRequirementProperty" minOccurs="0" maxOccurs="1"/>
	    </xsd:sequence>
	  </xsd:complexType>

element **OpMaterialRequirementType**

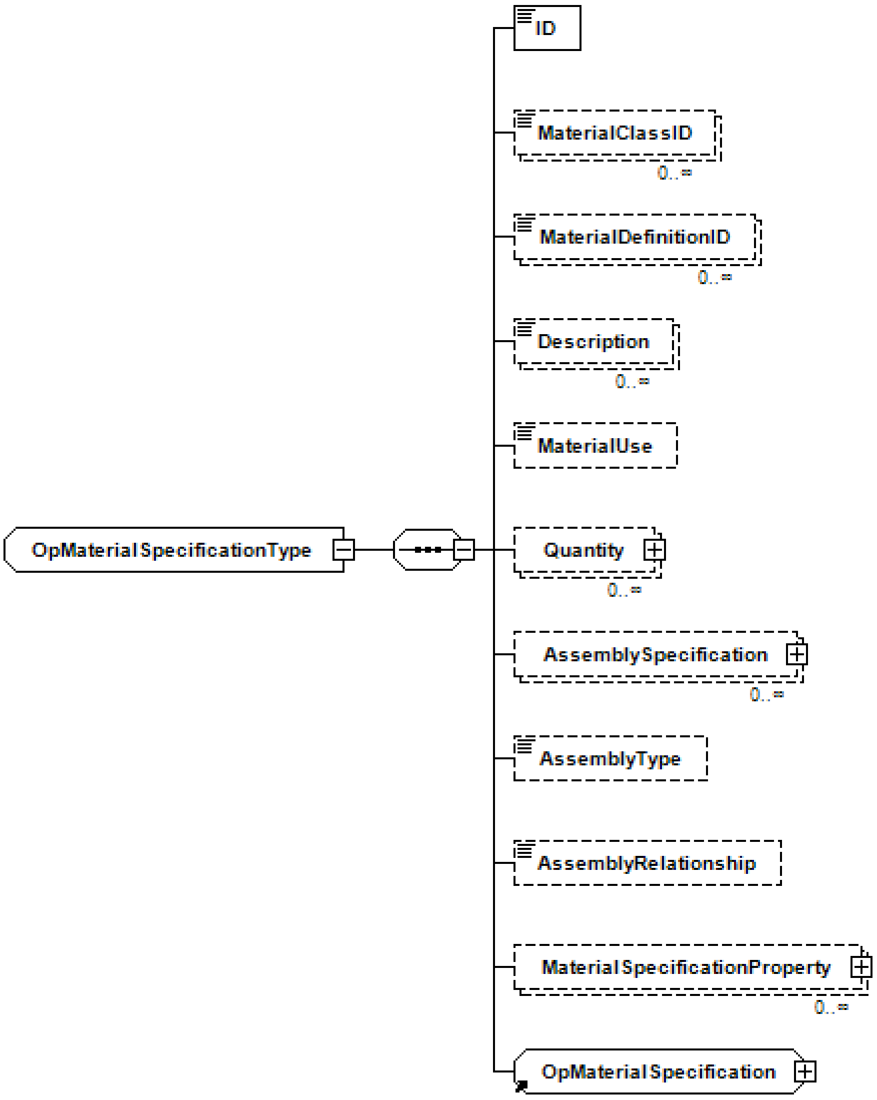
diagram	 OpMaterialRequirementType MaterialClassID 0..∞ MaterialDefinitionID 0..∞ MaterialLotID 0..∞ MaterialSubLotID 0..∞ Description 0..∞ MaterialUse StorageLocation Quantity 0..∞ AssemblyRequirement 0..∞ AssemblyType AssemblyRelationship HierarchyScope MaterialRequirementProperty 0..∞ RequiredByRequestedSegmentResponse OpMaterialRequirement
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="OpMaterialRequirementType"> <xsd:sequence> <xsd:element name="MaterialClassID" type="B2MML:MaterialClassIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="MaterialDefinitionID" type="B2MML:MaterialDefinitionIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="MaterialLotID" type="B2MML:MaterialLotIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="MaterialSubLotID" type="B2MML:MaterialSubLotIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="MaterialUse" type="B2MML:MaterialUseType" minOccurs="0"/> <xsd:element name="StorageLocation" type="B2MML:StorageLocationType" minOccurs="0"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="AssemblyRequirement" type="B2MML:OpMaterialRequirementType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="AssemblyType" type="B2MML:AssemblyTypeType" minOccurs="0"/> <xsd:element name="AssemblyRelationship" type="B2MML:AssemblyRelationshipType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="MaterialRequirementProperty" type="B2MML:OpMaterialRequirementPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="RequiredByRequestedSegmentResponse" type="B2MML:RequiredByRequestedSegmentResponseType" minOccurs="0"/> <xsd:group ref="B2MML:OpMaterialRequirement" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpMaterialSpecificationPropertyType**

diagram	 OpMaterialSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ OpMaterialSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpMaterialSpecificationPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpMaterialSpecificationProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

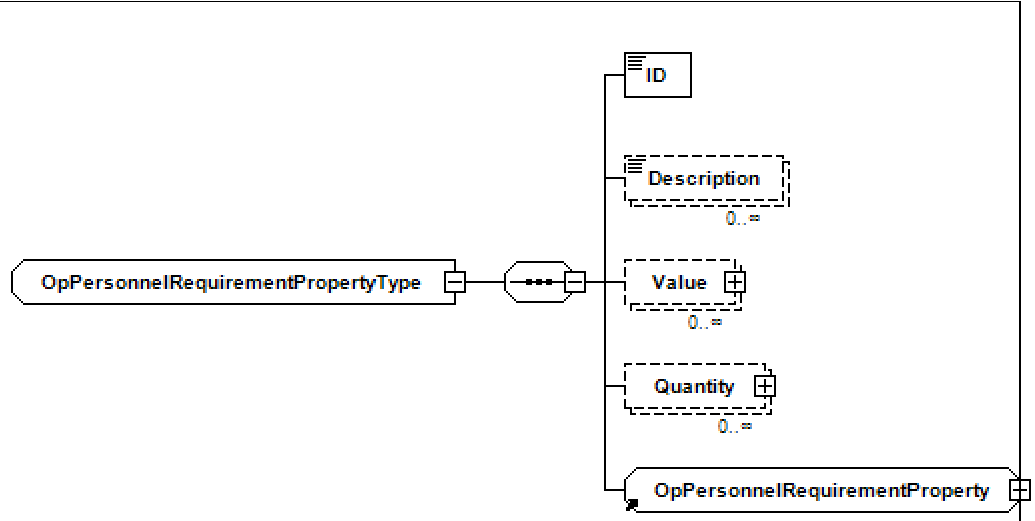
element **OpMaterialSpecificationType**

diagram	 ID MaterialClassID 0..∞ MaterialDefinitionID 0..∞ Description 0..∞ MaterialUse OpMaterialSpecificationType Quantity 0..∞ AssemblySpecification 0..∞ AssemblyType AssemblyRelationship MaterialSpecificationProperty 0..∞ OpMaterialSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

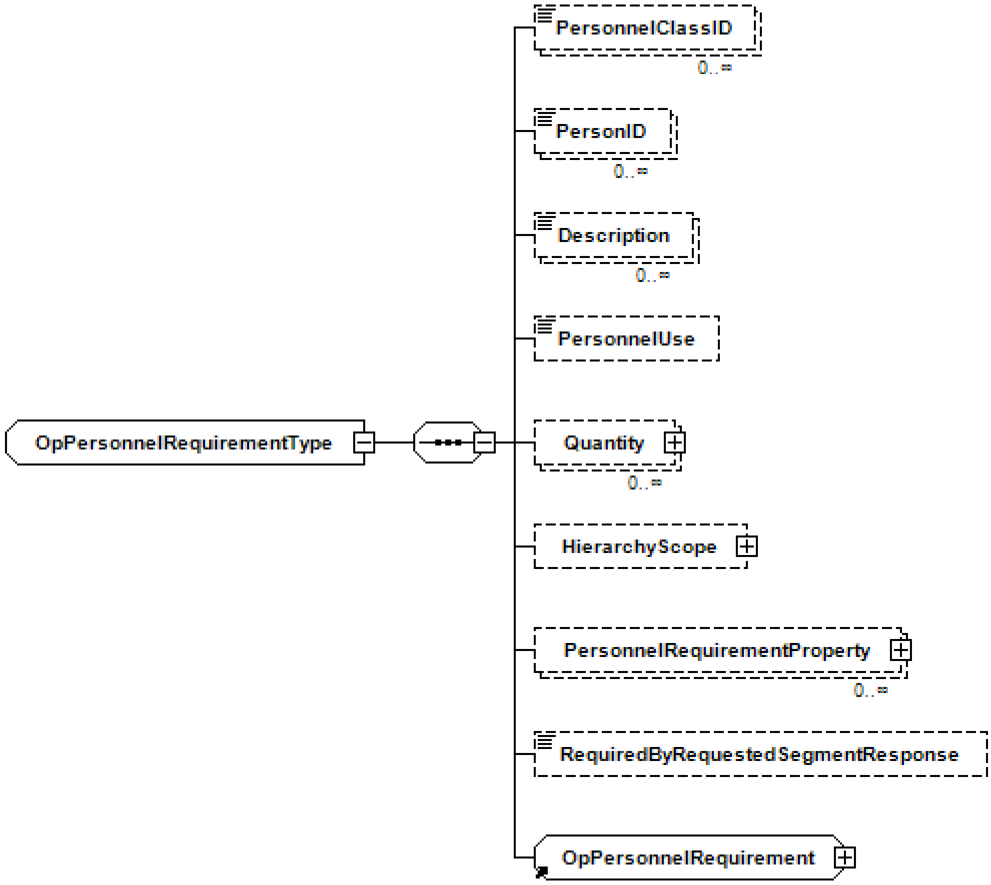
source	<xsd:complexType name="OpMaterialSpecificationType">
	    <xsd:sequence>
	        <xsd:element name="ID" type="B2MML:IdentifierType"/>
	        <xsd:element name="MaterialClassID" type="B2MML:MaterialClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="MaterialDefinitionID" type="B2MML:MaterialDefinitionIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="MaterialUse" type="B2MML:MaterialUseType" minOccurs="0"/>
	        <xsd:element name="Quantity" type="B2MML:QuantityValueType"

	minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="AssemblySpecification" type="B2MML:OpMaterialSpecificationType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="AssemblyType" type="B2MML:AssemblyTypeType" minOccurs="0"/> <xsd:element name="AssemblyRelationship" type="B2MML:AssemblyRelationshipType" minOccurs="0"/> <xsd:element name="MaterialSpecificationProperty" type="B2MML:OpMaterialSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpMaterialSpecification" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpPersonnelRequirementPropertyType**

diagram	 ID Description 0..∞ OpPersonnelRequirementPropertyType Value 0..∞ Quantity 0..∞ OpPersonnelRequirementProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpPersonnelRequirementPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpPersonnelRequirementProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

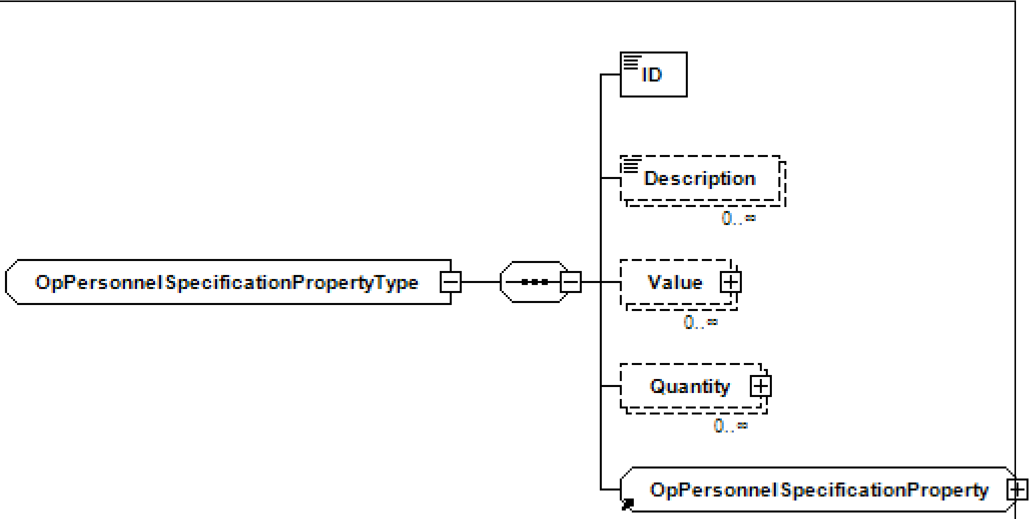
element **OpPersonnelRequirementType**

diagram	 PersonnelClassID 0..∞ PersonID 0..∞ Description 0..∞ PersonnelUse OpPersonnelRequirementType Quantity 0..∞ HierarchyScope PersonnelRequirementProperty 0..∞ RequiredByRequestedSegmentResponse OpPersonnelRequirement
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

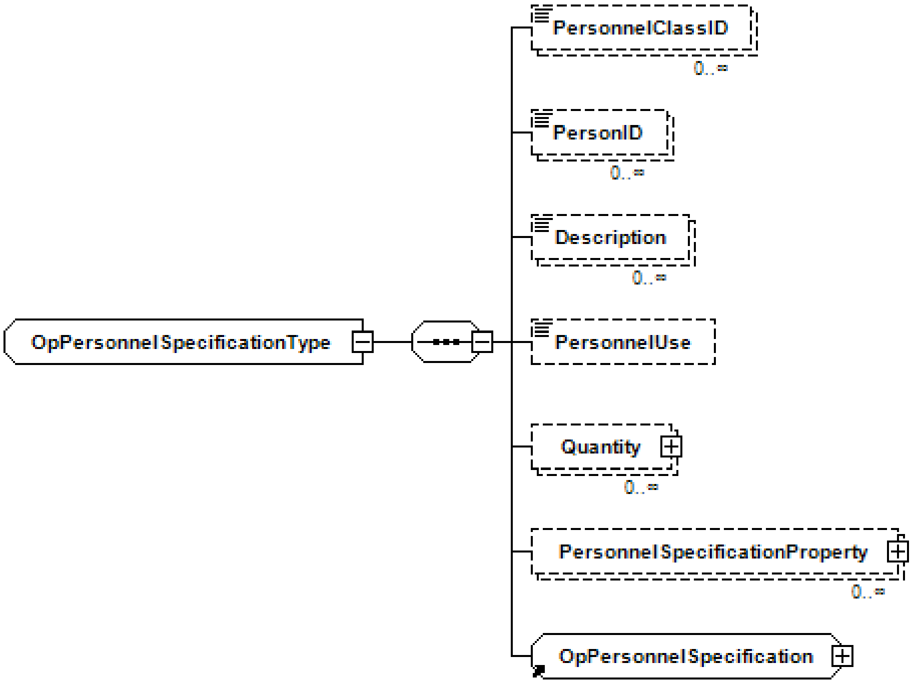
source	<xsd:complexType name="OpPersonnelRequirementType">
	    <xsd:sequence>
	        <xsd:element name="PersonnelClassID" type="B2MML:PersonnelClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PersonID" type="B2MML:PersonIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PersonnelUse" type="B2MML:PersonnelUseType" minOccurs="0"/>
	        <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	        <xsd:element name="PersonnelRequirementProperty" type="B2MML:OpPersonnelRequirementPropertyType" minOccurs="0"

	maxOccurs="unbounded"/> <xsd:element name="RequiredByRequestedSegmentResponse" type="B2MML:RequiredByRequestedSegmentResponseType" minOccurs="0"/> <xsd:group ref="B2MML:OpPersonnelRequirement" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpPersonnelSpecificationPropertyType**

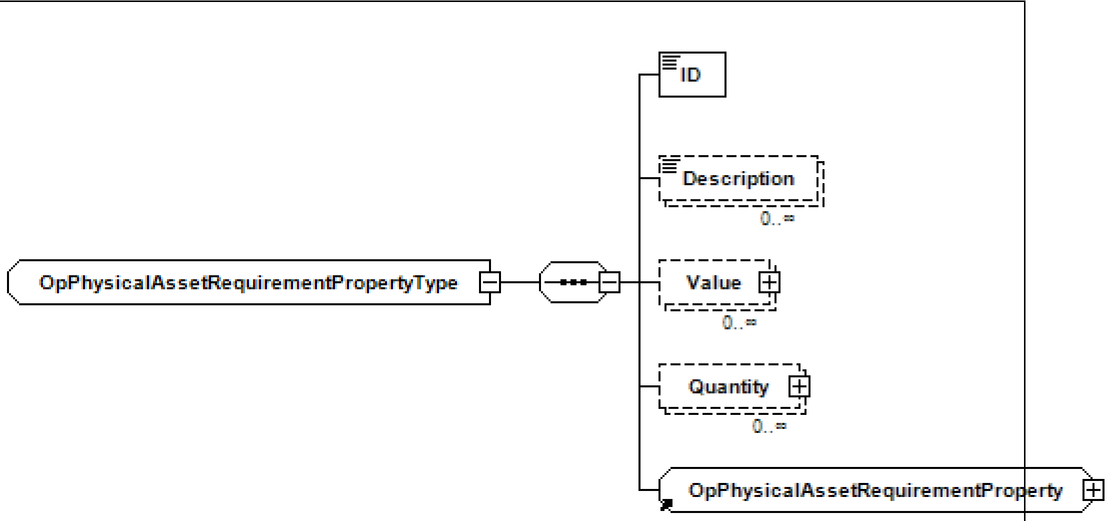
diagram	 ID Description 0..∞ OpPersonnelSpecificationPropertyType Value 0..∞ Quantity 0..∞ OpPersonnelSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpPersonnelSpecificationPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpPersonnelSpecificationProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpPersonnelSpecificationType**

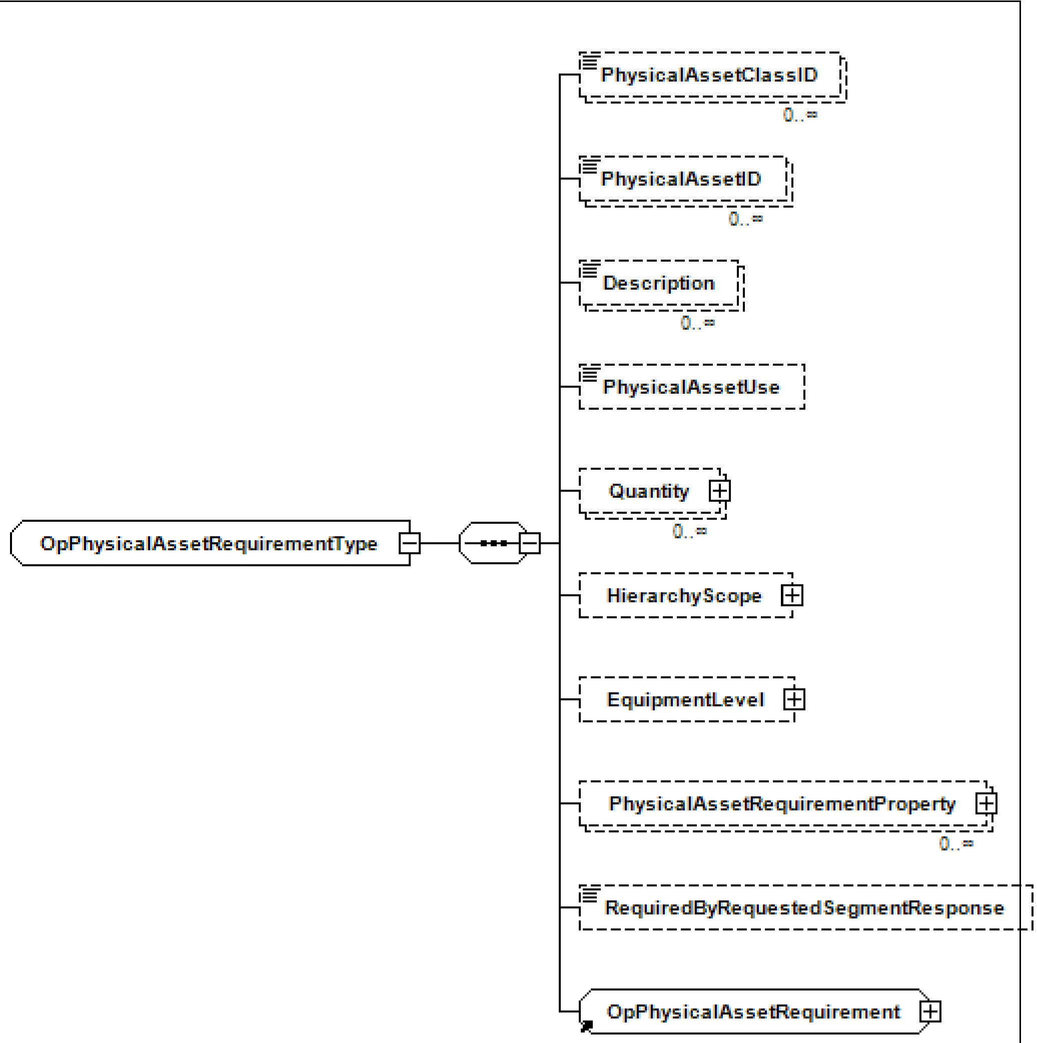
diagram	 PersonnelClassID 0..∞ PersonID 0..∞ Description 0..∞ OpPersonnelSpecificationType PersonnelUse Quantity 0..∞ PersonnelSpecificationProperty 0..∞ OpPersonnelSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="OpPersonnelSpecificationType">
	    <xsd:sequence>
	        <xsd:element name="PersonnelClassID" type="B2MML:PersonnelClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PersonID" type="B2MML:PersonIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PersonnelUse" type="B2MML:PersonnelUseType" minOccurs="0"/>
	        <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PersonnelSpecificationProperty" type="B2MML:OpPersonnelSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:group ref="B2MML:OpPersonnelSpecification" minOccurs="0" maxOccurs="1"/>
	    </xsd:sequence>
	</xsd:complexType>

element **OpPhysicalAssetRequirementPropertyType**

diagram	 OpPhysicalAssetRequirementPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ OpPhysicalAssetRequirementProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpPhysicalAssetRequirementPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpPhysicalAssetRequirementProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

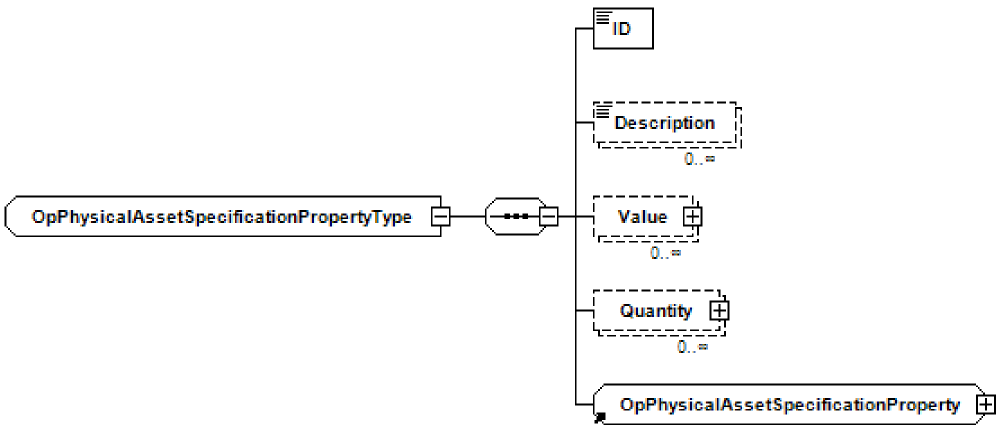
element **OpPhysicalAssetRequirementType**

diagram	 PhysicalAssetClassID 0..∞ PhysicalAssetID 0..∞ Description 0..∞ PhysicalAssetUse Quantity 0..∞ OpPhysicalAssetRequirementType HierarchyScope EquipmentLevel PhysicalAssetRequirementProperty 0..∞ RequiredByRequestedSegmentResponse OpPhysicalAssetRequirement
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

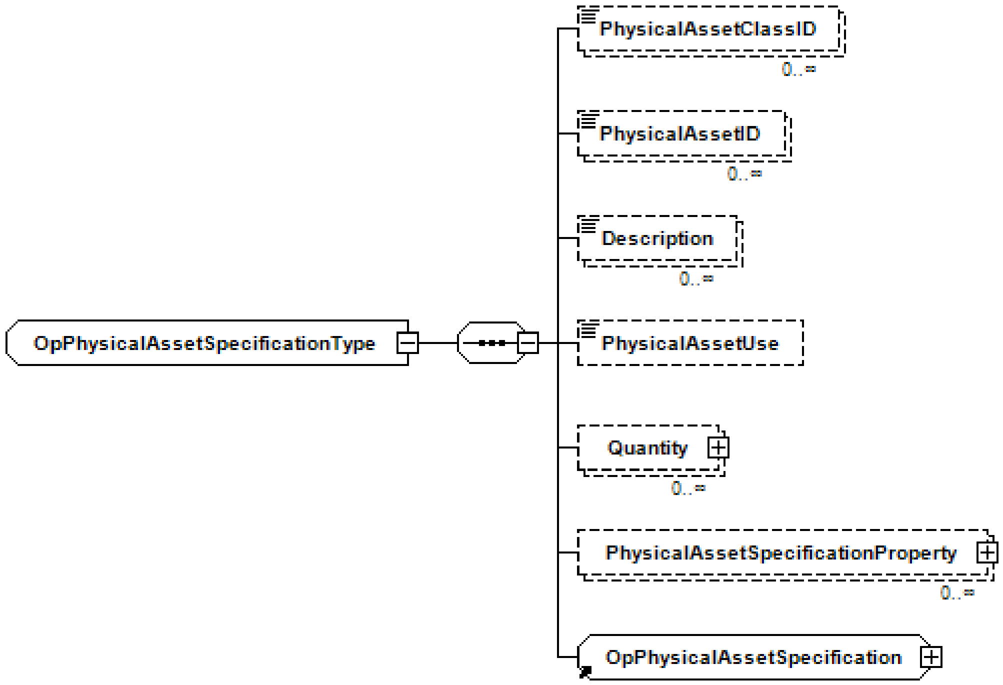
source	<xsd:complexType name="OpPhysicalAssetRequirementType">
	    <xsd:sequence>
	        <xsd:element name="PhysicalAssetClassID" type="B2MML:PhysicalAssetClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PhysicalAssetID" type="B2MML:PhysicalAssetIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PhysicalAssetUse" type="B2MML:PhysicalAssetUseType" minOccurs="0"/>
	        <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>

	<xsd:element name="EquipmentLevel" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PhysicalAssetRequirementProperty" type="B2MML:OpPhysicalAssetRequirementPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="RequiredByRequestedSegmentResponse" type="B2MML:RequiredByRequestedSegmentResponseType" minOccurs="0"/> <xsd:group ref="B2MML:OpPhysicalAssetRequirement" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpPhysicalAssetSpecificationPropertyType**

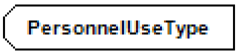
diagram	 OpPhysicalAssetSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ OpPhysicalAssetSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="OpPhysicalAssetSpecificationPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:OpPhysicalAssetSpecificationProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

element **OpPhysicalAssetSpecificationType**

diagram	 OpPhysicalAssetSpecificationType PhysicalAssetClassID 0..∞ PhysicalAssetID 0..∞ Description 0..∞ PhysicalAssetUse Quantity 0..∞ PhysicalAssetSpecificationProperty 0..∞ OpPhysicalAssetSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

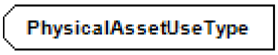
source	<xsd:complexType name="OpPhysicalAssetSpecificationType">
	    <xsd:sequence>
	        <xsd:element name="PhysicalAssetClassID" type="B2MML:PhysicalAssetClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PhysicalAssetID" type="B2MML:PhysicalAssetIDType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PhysicalAssetUse" type="B2MML:PhysicalAssetUseType" minOccurs="0"/>
	        <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:element name="PhysicalAssetSpecificationProperty" type="B2MML:OpPhysicalAssetSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	        <xsd:group ref="B2MML:OpPhysicalAssetSpecification" minOccurs="0" maxOccurs="1"/>
	    </xsd:sequence>
	</xsd:complexType>

element **PersonnelUseType**

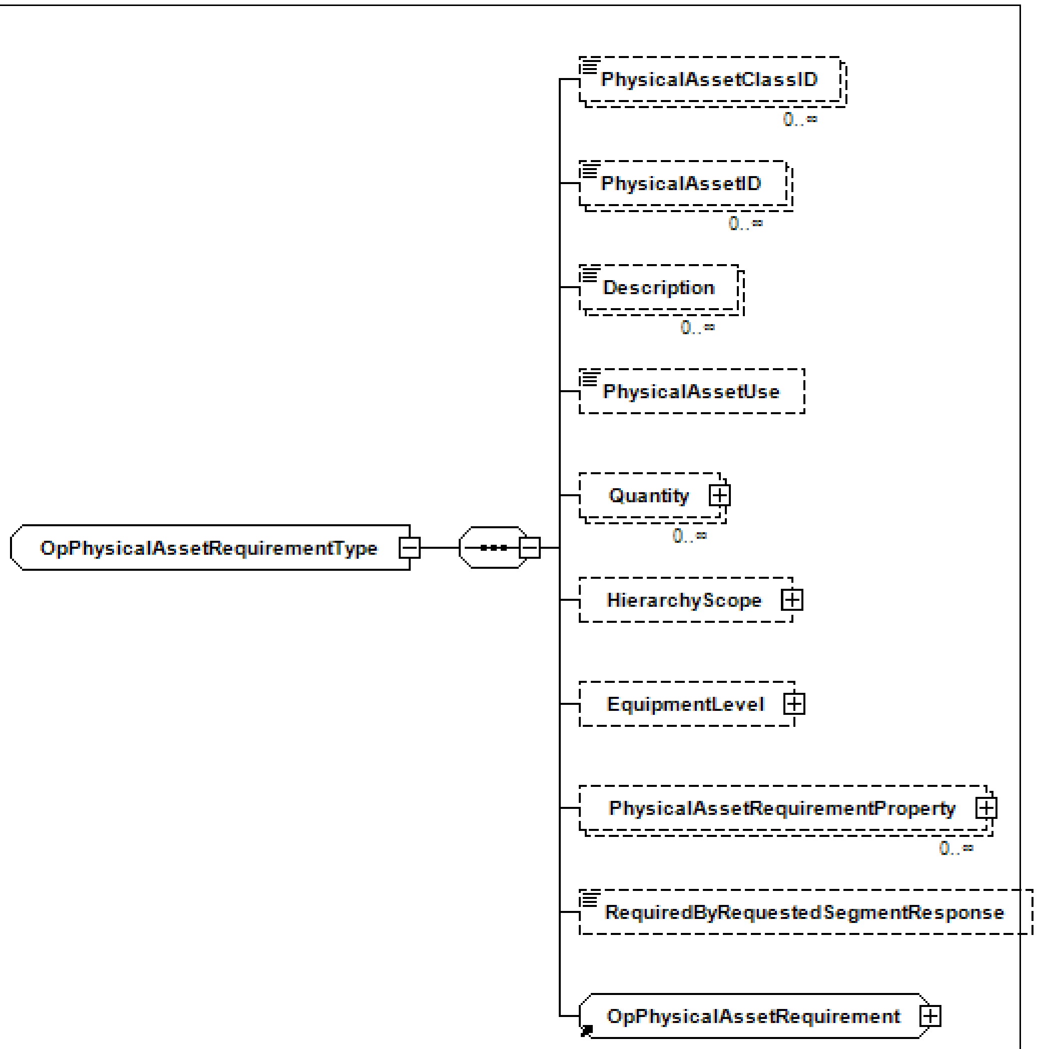
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="PersonnelUseType">
	    <xsd:simpleContent>
	        <xsd:restriction base="B2MML:CodeType"/>
	    </xsd:simpleContent>
	</xsd:complexType>

element **PhysicalAssetUseType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="PhysicalAssetUseType">
	    <xsd:simpleContent>
	        <xsd:restriction

	 PhysicalAssetClassID 0..∞ PhysicalAssetID 0..∞ Description 0..∞ PhysicalAssetUse Quantity 0..∞ OpPhysicalAssetRequirementType HierarchyScope EquipmentLevel PhysicalAssetRequirementProperty 0..∞ RequiredByRequestedSegmentResponse OpPhysicalAssetRequirement base="B2MML:CodeType"/> </xsd:simpleContent> </xsd:complexType>

element **EquipmentInformation**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:element name="EquipmentInformation" type="B2MML:EquipmentInformationType"/>

element **PersonnelInformation**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:element name="PersonnelInformation" type="B2MML:PersonnelInformationType"/>

element **PhysicalAssetInformation**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:element name="PhysicalAssetInformation" type="B2MML:PhysicalAssetInformationType"/>

element **ProcessSegmentInformation**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:element name="ProcessSegmentInformation" type="B2MML:ProcessSegmentInformationType"/>

complexType **AnyGenericValueType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="AnyGenericValueType"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="currencyID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="currencyCodeListVersionID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="encodingCode" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="format" type="xsd:string" use="optional"/> <xsd:attribute name="characterSetCode" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="listID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="listAgencyID" type="xsd:normalizedString" use="optional"/>

```
                <xsd:attribute name="listAgencyName" type="xsd:string" use="optional"/>
                <xsd:attribute name="listName" type="xsd:string" use="optional"/>
                <xsd:attribute name="listVersionID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="languageID" type="xsd:language" use="optional"/>
                <xsd:attribute name="languageLocaleID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="listURI" type="xsd:anyURI" use="optional"/>
                <xsd:attribute name="listSchemaURI" type="xsd:anyURI" use="optional"/>
                <xsd:attribute name="mimeCode" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="name" type="xsd:string" use="optional"/>
                <xsd:attribute name="schemaID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="schemaName" type="xsd:string" use="optional"/>
                <xsd:attribute name="schemaAgencyID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="schemaAgencyName" type="xsd:string" use="optional"/>
                <xsd:attribute name="schemaVersionID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="schemaDataURI" type="xsd:anyURI" use="optional"/>
                <xsd:attribute name="schemaURI" type="xsd:anyURI" use="optional"/>
                <xsd:attribute name="unitCode" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="unitCodeListID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="unitCodeListAgencyID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="unitCodeListAgencyName" type="xsd:string" use="optional"/>
                <xsd:attribute name="unitCodeListVersionID" type="xsd:normalizedString" use="optional"/>
                <xsd:attribute name="filename" type="xsd:string" use="optional"/>
                <xsd:attribute name="uri" type="xsd:anyURI" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:complexType>
```

complexType **AssemblyRelationship1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="AssemblyRelationship1Type">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType">
	      <xsd:enumeration value="B2MML:Permanent"/>
	      <xsd:enumeration value="Transient"/>
	      <xsd:enumeration value="Other"/>
	    </xsd:restriction>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **AssemblyRelationshipType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="AssemblyRelationshipType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:AssemblyRelationship1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **AssemblyType1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="AssemblyType1Type">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType">
	      <xsd:enumeration value="Physical"/>
	      <xsd:enumeration value="Logical"/>
	      <xsd:enumeration value="Other"/>
	    </xsd:restriction>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **AssemblyTypeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="AssemblyTypeType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:AssemblyType1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **DataType1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="DataType1Type">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType">
	      <xsd:enumeration value="Amount"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="BinaryObject"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Code"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="DateTime"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Identifier"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Indicator"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Measure"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Numeric"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Quantity"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="Text"/>
	      <!-- UN/CEFACT Core Component Type -->
	      <xsd:enumeration value="string"/>
	      <xsd:enumeration value="byte"/>
	      <xsd:enumeration value="unsignedByte"/>
	      <xsd:enumeration value="binary"/>
	      <xsd:enumeration value="integer"/>

```
            <xsd:enumeration value="positiveInteger"/>
            <xsd:enumeration value="negativeInteger"/>
            <xsd:enumeration value="nonNegativeInteger"/>
            <xsd:enumeration value="nonPositiveInteger"/>
            <xsd:enumeration value="int"/>
            <xsd:enumeration value="unsignedInt"/>
            <xsd:enumeration value="long"/>
            <xsd:enumeration value="unsignedLong"/>
            <xsd:enumeration value="short"/>
            <xsd:enumeration value="unsignedShort"/>
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="float"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="boolean"/>
            <xsd:enumeration value="time"/>
            <xsd:enumeration value="timeInstant"/>
            <xsd:enumeration value="timePeriod"/>
            <xsd:enumeration value="duration"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="dateTime"/>
            <xsd:enumeration value="month"/>
            <xsd:enumeration value="year"/>
            <xsd:enumeration value="century"/>
            <xsd:enumeration value="recurringDay"/>
            <xsd:enumeration value="recurringDate"/>
            <xsd:enumeration value="recurringDuration"/>
            <xsd:enumeration value="Name"/>
            <xsd:enumeration value="QName"/>
            <xsd:enumeration value="NCName"/>
            <xsd:enumeration value="uriReference"/>
            <xsd:enumeration value="language"/>
            <xsd:enumeration value="ID"/>
            <xsd:enumeration value="IDREF"/>
            <xsd:enumeration value="IDREFS"/>
            <xsd:enumeration value="ENTITY"/>
            <xsd:enumeration value="ENTITIES"/>
            <xsd:enumeration value="NOTATION"/>
            <xsd:enumeration value="NMTOKEN"/>
            <xsd:enumeration value="NMTOKENS"/>
            <xsd:enumeration value="Enumeration"/>
            <xsd:enumeration value="SVG"/>
            <xsd:enumeration value="Other"/>
        </xsd:restriction>
    </xsd:simpleContent>
</xsd:complexType>
```

complexType **DataTypeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="DataTypeType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:DataType1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **Dependency1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="Dependency1Type">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType">
	      <xsd:enumeration value="NotFollow"/>
	      <xsd:enumeration value="PossibleParallel"/>
	      <xsd:enumeration value="NotInParallel"/>
	      <xsd:enumeration value="AtStart"/>
	      <xsd:enumeration value="AfterStart"/>
	      <xsd:enumeration value="AfterEnd"/>
	      <xsd:enumeration value="NoLaterAfterStart"/>
	      <xsd:enumeration value="NoEarlierAfterStart"/>
	      <xsd:enumeration value="NoLaterAfterEnd"/>
	      <xsd:enumeration value="NoEarlierAfterEnd"/>
	      <xsd:enumeration value="Other"/>
	    </xsd:restriction>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **DependencyType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

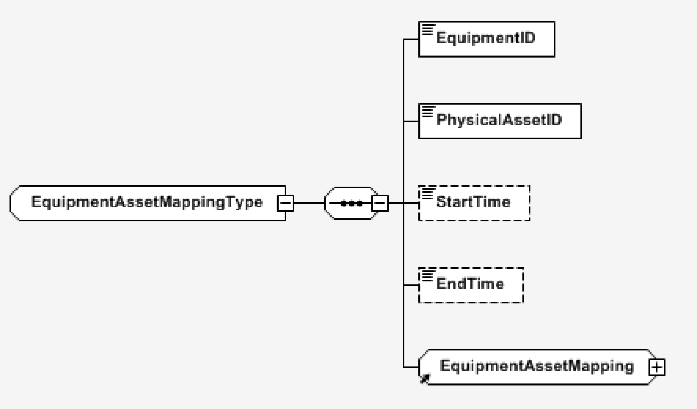
source	<xsd:complexType name="DependencyType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:Dependency1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>

	</xsd:simpleContent> </xsd:complexType>

complexType **DescriptionType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="DescriptionType"> <xsd:simpleContent> <xsd:restriction base="B2MML:TextType"/> </xsd:simpleContent> </xsd:complexType>

complexType **EquipmentAssetMappingType**

diagram	 EquipmentAssetMappingType EquipmentID PhysicalAssetID StartTime EndTime EquipmentAssetMapping
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentAssetMappingType"> <xsd:sequence> <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType"/> <xsd:element name="PhysicalAssetID" type="B2MML:PhysicalAssetIDType"/> <xsd:element name="StartTime" type="B2MML:DateTimeType" minOccurs="0"/> <xsd:element name="EndTime" type="B2MML:DateTimeType" minOccurs="0"/> <xsd:group ref="B2MML:EquipmentAssetMapping" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **EquipmentCapabilityTestSpecificationIDType**

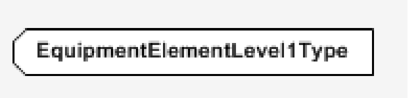
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentCapabilityTestSpecificationIDType">

	<xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

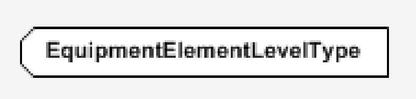
complexType **EquipmentClassIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentClassIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **EquipmentElementLevel1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentElementLevel1Type"> <xsd:simpleContent> <xsd:restriction base="B2MML:CodeType"> <xsd:enumeration value="Enterprise"/> <xsd:enumeration value="Site"/> <xsd:enumeration value="Area"/> <xsd:enumeration value="ProcessCell"/> <xsd:enumeration value="Unit"/> <xsd:enumeration value="ProductionLine"/> <xsd:enumeration value="WorkCell"/> <xsd:enumeration value="ProductionUnit"/> <xsd:enumeration value="StorageZone"/> <xsd:enumeration value="StorageUnit"/> <xsd:enumeration value="WorkCenter"/> <xsd:enumeration value="WorkUnit"/> <xsd:enumeration value="EquipmentModule"/> <xsd:enumeration value="ControlModule"/> <xsd:enumeration value="Other"/> </xsd:restriction> </xsd:simpleContent> </xsd:complexType>

complexType **EquipmentElementLevelType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="EquipmentElementLevelType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:EquipmentElementLevel1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **EquipmentIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

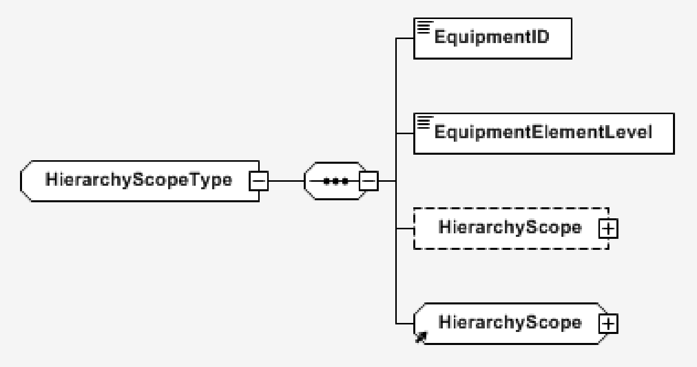
source	<xsd:complexType name="EquipmentIDType">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:IdentifierType"/>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **ExpirationTimeType**

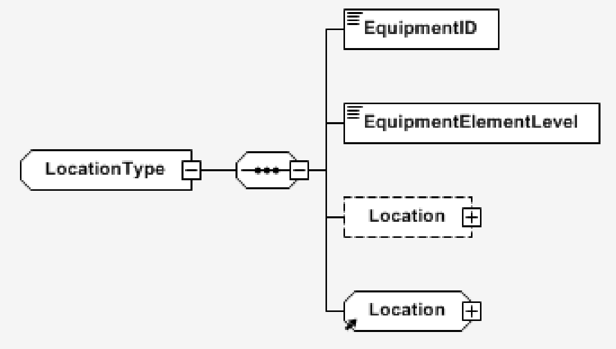
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="ExpirationTimeType">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:DateTimeType"/>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **HierarchyScopeType**

diagram	 HierarchyScopeType EquipmentID EquipmentElementLevel HierarchyScope HierarchyScope
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="HierarchyScopeType"> <xsd:sequence> <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType"/> <xsd:element name="EquipmentElementLevel" type="B2MML:EquipmentElementLevelType"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:group ref="B2MML:HierarchyScope" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **LocationType**

diagram	 LocationType EquipmentID EquipmentElementLevel Location Location
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="LocationType"> <xsd:sequence> <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType"/> <xsd:element name="EquipmentElementLevel" type="B2MML:EquipmentElementLevelType"/> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:group ref="B2MML:Location" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **MaterialClassIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="MaterialClassIDType">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:IdentifierType"/>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **MaterialDefinitionIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="MaterialDefinitionIDType">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:IdentifierType"/>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **MaterialUse1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="MaterialUse1Type">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType">
	      <xsd:enumeration value="Consumed"/>
	      <xsd:enumeration value="Produced"/>
	      <xsd:enumeration value="Consumable"/>
	      <xsd:enumeration value="Replaced Assetn"/>
	      <xsd:enumeration value="Replacement Asset"/>
	      <xsd:enumeration value="Sample"/>
	      <xsd:enumeration value="Resurned Sample"/>
	      <xsd:enumeration value="Carrier"/>
	      <xsd:enumeration value="Returned Carrier"/>
	      <xsd:enumeration value="Other"/>
	    </xsd:restriction>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **MaterialUseType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="MaterialUseType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:MaterialUse1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **OperationsType1Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

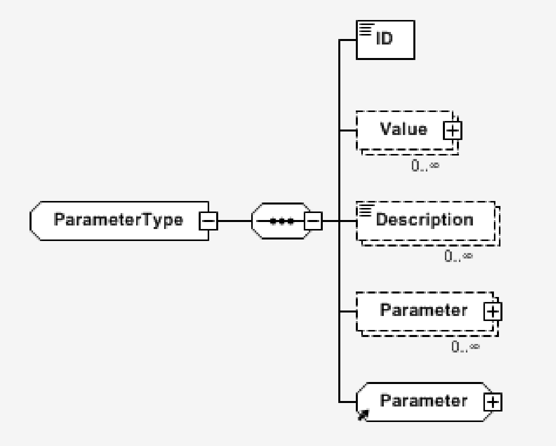
source	<xsd:complexType name="OperationsType1Type">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType">
	      <xsd:enumeration value="Production"/>
	      <xsd:enumeration value="Maintenance"/>
	      <xsd:enumeration value="Quality"/>
	      <xsd:enumeration value="Inventory"/>
	      <xsd:enumeration value="Mixed"/>
	      <xsd:enumeration value="Other"/>
	    </xsd:restriction>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **OperationsTypeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="OperationsTypeType">
	  <xsd:simpleContent>
	    <xsd:extension base="B2MML:OperationsType1Type">
	      <xsd:attribute name="OtherValue" type="xsd:string"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

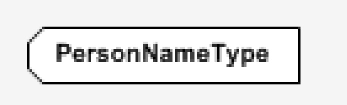
complexType **ParameterType**

diagram	 ID Value 0..∞ ParameterType Description 0..∞ Parameter 0..∞ Parameter
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ParameterType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Parameter" type="B2MML:ParameterType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:Parameter" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PersonIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **PersonNameType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonNameType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/>

	</xsd:simpleContent> </xsd:complexType>

complexType **PersonnelClassIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonnelClassIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **PhysicalAssetCapabilityTestSpecificationIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetCapabilityTestSpecificationIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **PhysicalAssetClassIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetClassIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **PhysicalAssetIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetIDType"> <xsd:simpleContent>

	<xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **PriorityType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PriorityType"> <xsd:simpleContent> <xsd:restriction base="B2MML:NumericType"/> </xsd:simpleContent> </xsd:complexType>

complexType **ProcessSegmentIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ProcessSegmentIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **ProductProductionRuleIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ProductProductionRuleIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **ProductSegmentIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ProductSegmentIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent>

	</xsd:complexType>

complexType **PropertyIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PropertyIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **PublishedDateType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PublishedDateType"> <xsd:simpleContent> <xsd:restriction base="B2MML:DateTimeType"/> </xsd:simpleContent> </xsd:complexType>

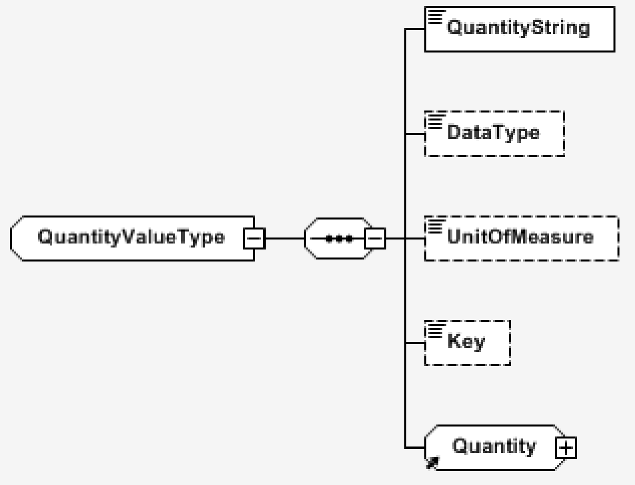
complexType **QualificationTestSpecificationIDType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="QualificationTestSpecificationIDType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **QuantityStringType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="QuantityStringType"> <xsd:simpleContent> <xsd:restriction base="B2MML:AnyGenericValueType"/> </xsd:simpleContent> </xsd:complexType>

complexType **QuantityValueType**

diagram	 QuantityValueType QuantityString DataType UnitOfMeasure Key Quantity
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	`<xsd:complexType name="QuantityValueType">`
	`<xsd:sequence>`
	`<xsd:element name="QuantityString" type="B2MML:QuantityStringType" nillable="true"/>`
	`<xsd:element name="DataType" type="B2MML:DataTypeType" nillable="true" minOccurs="0"/>`
	`<xsd:element name="UnitOfMeasure" type="B2MML:UnitOfMeasureType" nillable="true" minOccurs="0"/>`
	`<xsd:element name="Key" type="B2MML:IdentifierType" minOccurs="0" maxOccurs="1"/>`
	`<xsd:group ref="B2MML:Quantity" minOccurs="0" maxOccurs="1"/>`
	`</xsd:sequence>`
	`</xsd:complexType>`

complexType **RequiredByRequestedSegmentResponse1Type**

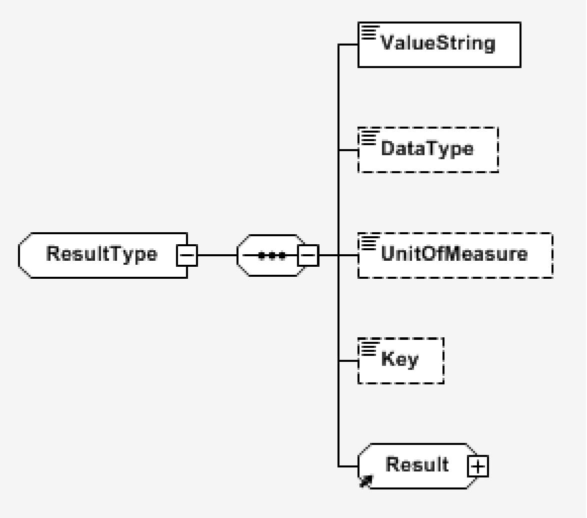
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	`<xsd:complexType name="RequiredByRequestedSegmentResponse1Type">`
	`<xsd:simpleContent>`
	`<xsd:restriction base="B2MML:CodeType">`
	`<xsd:enumeration value="Required"/>`
	`<xsd:enumeration value="Optional"/>`
	`<xsd:enumeration value="Other"/>`
	`</xsd:restriction>`
	`</xsd:simpleContent>`
	`</xsd:complexType>`

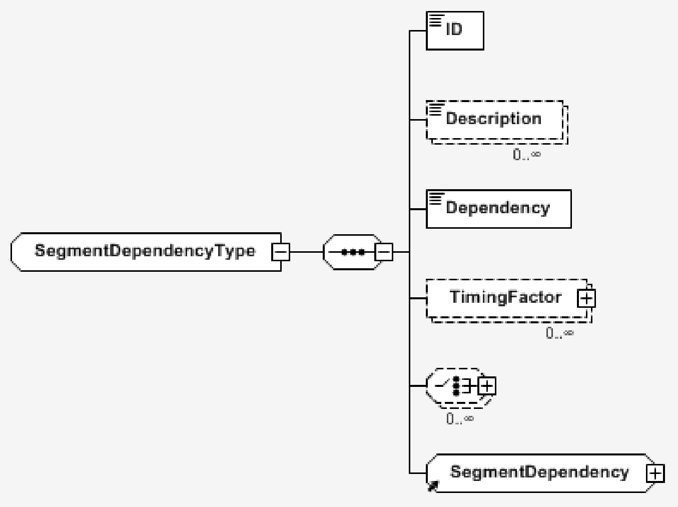
complexType **RequiredByRequestedSegmentResponseType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="RequiredByRequestedSegmentResponseType"> <xsd:simpleContent> <xsd:extension base="B2MML:RequiredByRequestedSegmentResponse1Type"> <xsd:attribute name="OtherValue" type="xsd:string"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

complexType **ResultType**

diagram	 ValueString DataType ResultType UnitOfMeasure Key Result
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ResultType"> <xsd:sequence> <xsd:element name="ValueString" type="B2MML:ValueStringType" nillable="true"/> <xsd:element name="DataType" type="B2MML:DataTypeType" nillable="true" minOccurs="0"/> <xsd:element name="UnitOfMeasure" type="B2MML:UnitOfMeasureType" nillable="true" minOccurs="0"/> <xsd:element name="Key" type="B2MML:IdentifierType" minOccurs="0" maxOccurs="1"/> <xsd:group ref="B2MML:Result" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **SegmentDependencyType**

diagram	 ID Description 0..∞ Dependency SegmentDependencyType TimingFactor 0..∞ 0..∞ SegmentDependency
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="SegmentDependencyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Dependency" type="B2MML:DependencyType"/> <xsd:element name="TimingFactor" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="ProductSegmentID" type="B2MML:ProductSegmentIDType"/> <xsd:element name="ProcessSegmentID" type="B2MML:ProcessSegmentIDType"/> <xsd:element name="SegmentID" type="B2MML:IdentifierType"/> </xsd:choice> <xsd:group ref="B2MML:SegmentDependency" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

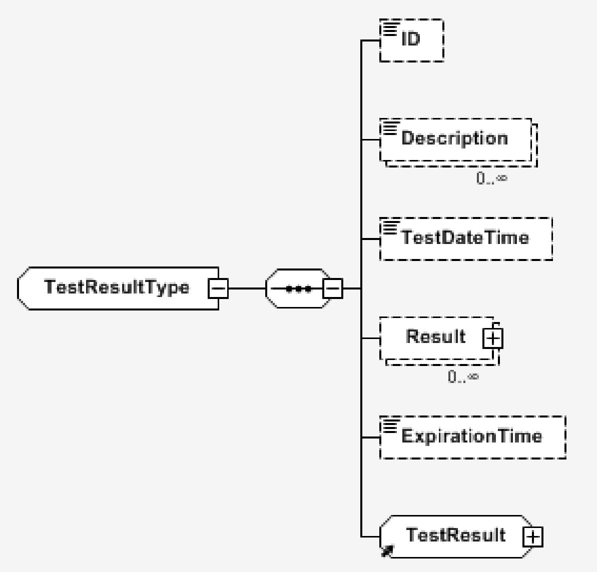
complexType **StartTimeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="StartTimeType"> <xsd:simpleContent> <xsd:restriction base="B2MML:DateTimeType"/> </xsd:simpleContent> </xsd:complexType>

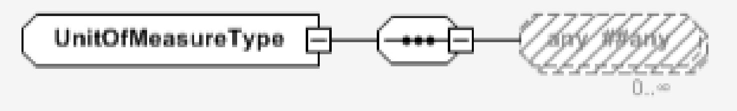
complexType **TestDateTimeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="TestDateTimeType"> <xsd:simpleContent> <xsd:restriction base="B2MML:DateTimeType"/> </xsd:simpleContent> </xsd:complexType>

complexType **TestResultType**

diagram	 ID Description 0..∞ TestDateTime TestResultType Result 0..∞ ExpirationTime TestResult
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="TestResultType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="TestDateTime" type="B2MML:TestDateTimeType" minOccurs="0"/> <xsd:element name="Result" type="B2MML:ResultType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="ExpirationTime" type="B2MML:ExpirationTimeType" minOccurs="0"/> <xsd:group ref="B2MML:TestResult" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **UnitOfMeasureType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

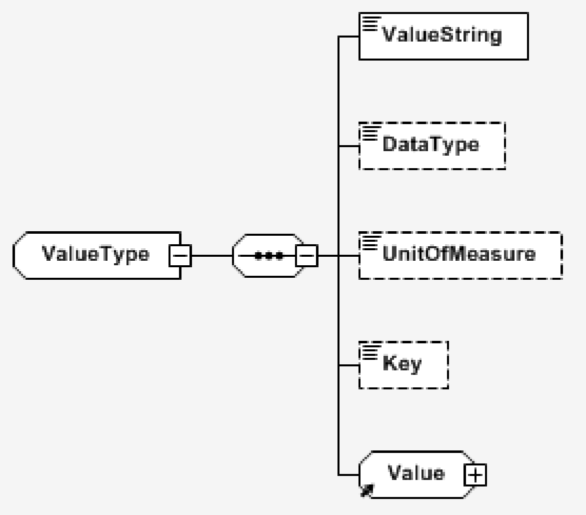
source	<xsd:complexType name="UnitOfMeasureType">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:CodeType"/>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **ValueStringType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="ValueStringType">
	  <xsd:simpleContent>
	    <xsd:restriction base="B2MML:AnyGenericValueType"/>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **ValueType**

diagram	 ValueType ValueString DataType UnitOfMeasure Key Value
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="ValueType">
	  <xsd:sequence>
	    <xsd:element name="ValueString" type="B2MML:ValueStringType" nillable="true"/>
	    <xsd:element name="DataType" type="B2MML:DataTypeType" nillable="true" minOccurs="0"/>
	    <xsd:element name="UnitOfMeasure" type="B2MML:UnitOfMeasureType" nillable="true" minOccurs="0"/>

	<xsd:element name="Key" type="B2MML:IdentifierType" minOccurs="0" maxOccurs="1"/> <xsd:group ref="B2MML:Value" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **VersionType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="VersionType"> <xsd:simpleContent> <xsd:restriction base="B2MML:IdentifierType"/> </xsd:simpleContent> </xsd:complexType>

complexType **CodeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="CodeType"> <xsd:simpleContent> <xsd:extension base="xsd:normalizedString"> <xsd:attribute name="listID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="listAgencyID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="listAgencyName" type="xsd:string" use="optional"/> <xsd:attribute name="listName" type="xsd:string" use="optional"/> <xsd:attribute name="listVersionID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="name" type="xsd:string" use="optional"/> <xsd:attribute name="languageID" type="xsd:language" use="optional"/> <xsd:attribute name="listURI" type="xsd:anyURI" use="optional"/> <xsd:attribute name="listSchemeURI" type="xsd:anyURI" use="optional"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

complexType **DateTimeType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="DateTimeType"> <xsd:simpleContent>

	<xsd:extension base="xsd:dateTime"> <xsd:attribute name="format" type="xsd:string" use="optional"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

complexType **IdentifierType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="IdentifierType"> <xsd:simpleContent> <xsd:extension base="xsd:normalizedString"> <xsd:attribute name="schemeID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="schemeName" type="xsd:string" use="optional"/> <xsd:attribute name="schemeAgencyID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="schemeAgencyName" type="xsd:string" use="optional"/> <xsd:attribute name="schemeVersionID" type="xsd:normalizedString" use="optional"/> <xsd:attribute name="schemeDataURI" type="xsd:anyURI" use="optional"/> <xsd:attribute name="schemeURI" type="xsd:anyURI" use="optional"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

complexType **NameType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="NameType"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="languageID" type="xsd:language" use="optional"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

complexType **NumericType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

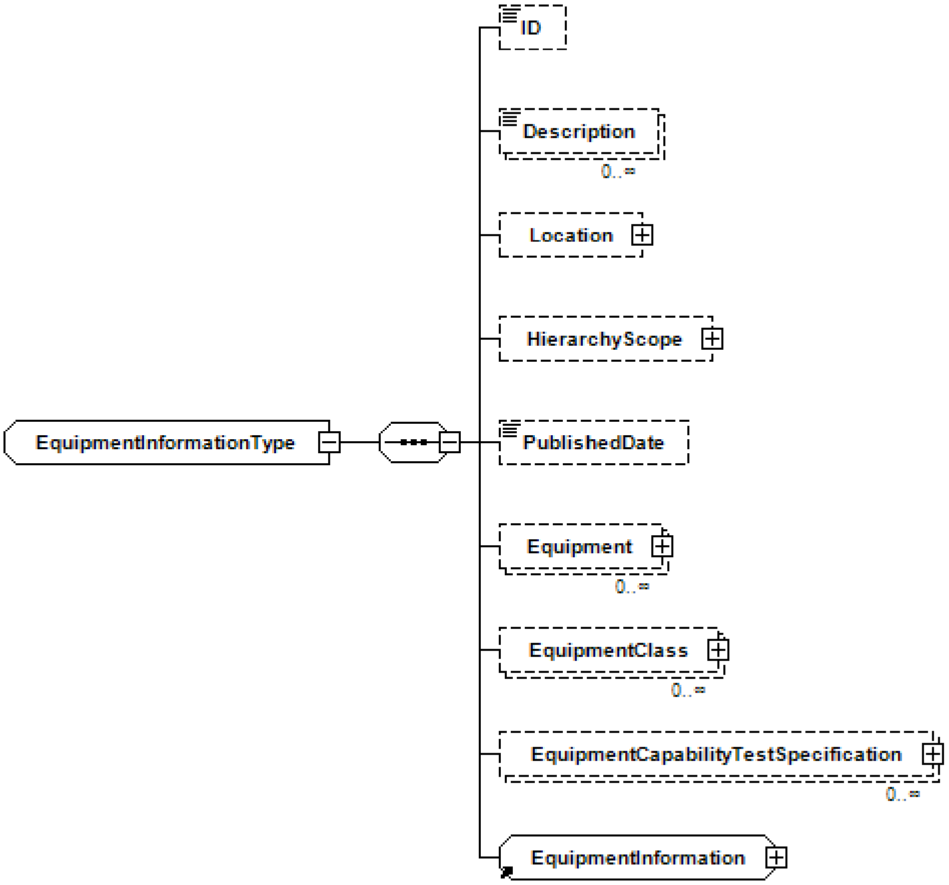
source	<xsd:complexType name="NumericType">
	  <xsd:simpleContent>
	    <xsd:extension base="xsd:decimal">
	      <xsd:attribute name="format" type="xsd:string" use="optional"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **TextType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

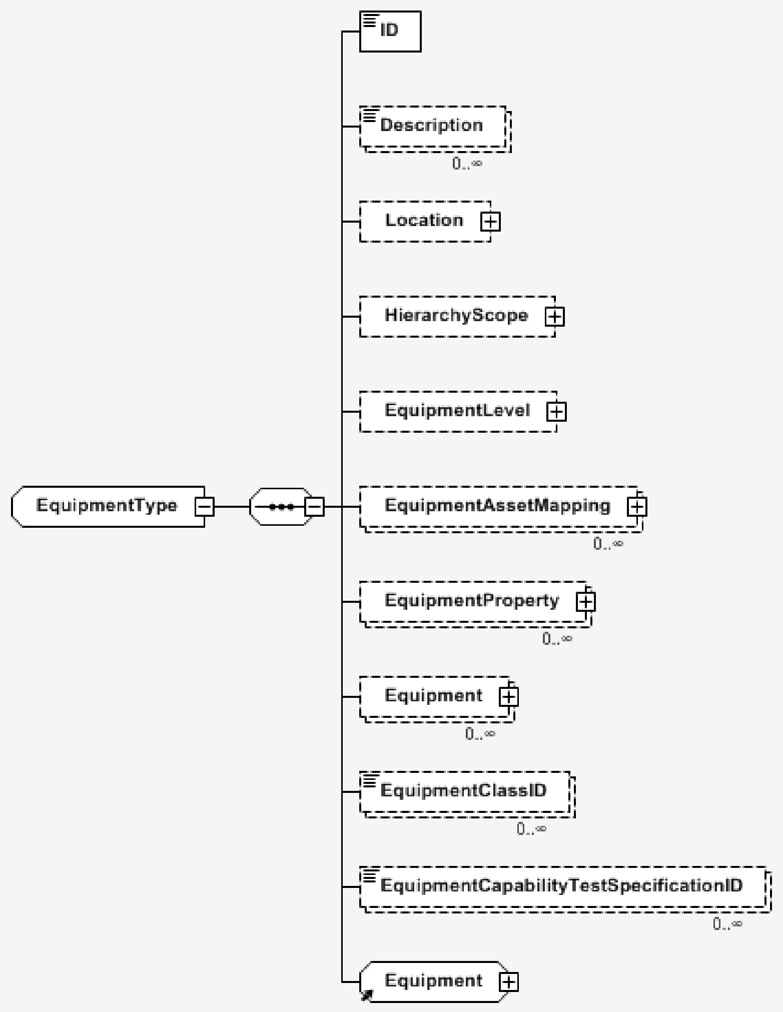
source	<xsd:complexType name="TextType">
	  <xsd:simpleContent>
	    <xsd:extension base="xsd:string">
	      <xsd:attribute name="languageID" type="xsd:language" use="optional"/>
	    </xsd:extension>
	  </xsd:simpleContent>
	</xsd:complexType>

complexType **EquipmentInformationType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentInformationType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead --> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PublishedDate" type="B2MML:PublishedDateType" minOccurs="0"/> <xsd:element name="Equipment" type="B2MML:EquipmentType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentClass" type="B2MML:EquipmentClassType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentCapabilityTestSpecification" type="B2MML:EquipmentCapabilityTestSpecificationType" minOccurs="0"

	maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:EquipmentInformation" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	 </xsd:complexType>

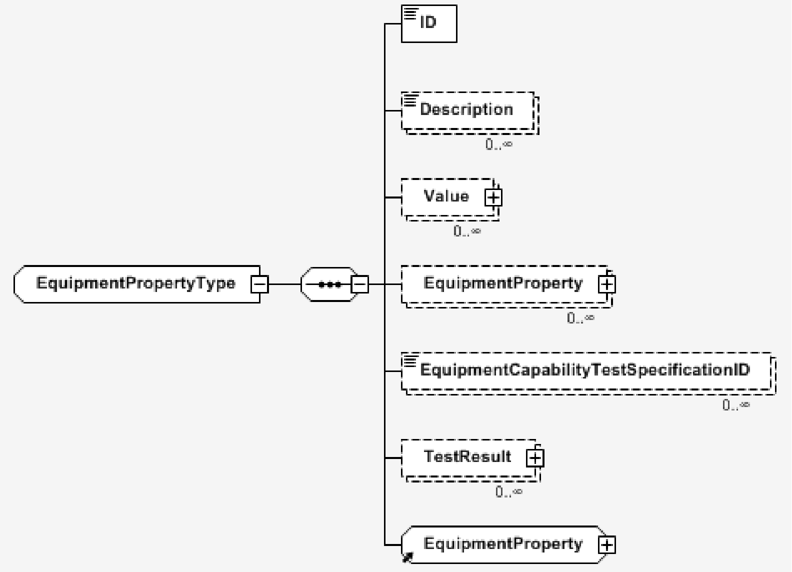
complexType **EquipmentType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="EquipmentType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead -->
	    <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/>
	    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	    <xsd:element name="EquipmentLevel" type="B2MML:HierarchyScopeType"

	minOccurs="0"/>
	    <xsd:element name="EquipmentAssetMapping" type="B2MML:EquipmentAssetMappingType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentProperty" type="B2MML:EquipmentPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Equipment" type="B2MML:EquipmentType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentClassID" type="B2MML:EquipmentClassIDType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentCapabilityTestSpecificationID" type="B2MML:EquipmentCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:Equipment" minOccurs="0" maxOccurs="1"/>
	   </xsd:sequence>
	  </xsd:complexType>

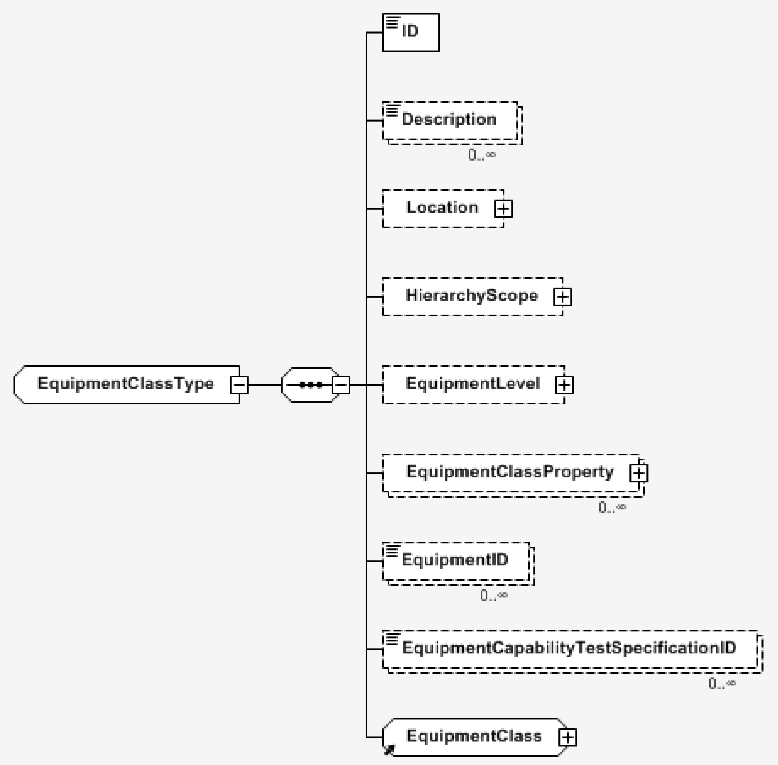
complexType **EquipmentPropertyType**

diagram	 ID Description 0..∞ Value 0..∞ EquipmentPropertyType EquipmentProperty 0..∞ EquipmentCapabilityTestSpecificationID 0..∞ TestResult 0..∞ EquipmentProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="EquipmentPropertyType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentProperty" type="B2MML:EquipmentPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentCapabilityTestSpecificationID" type="B2MML:EquipmentCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/>

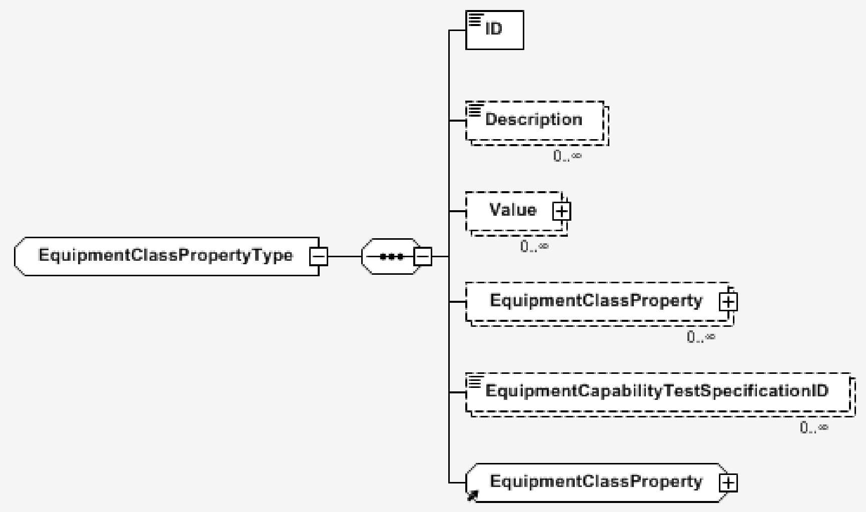
	<xsd:element name="TestResult" type="B2MML:TestResultType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:EquipmentProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **EquipmentClassType**

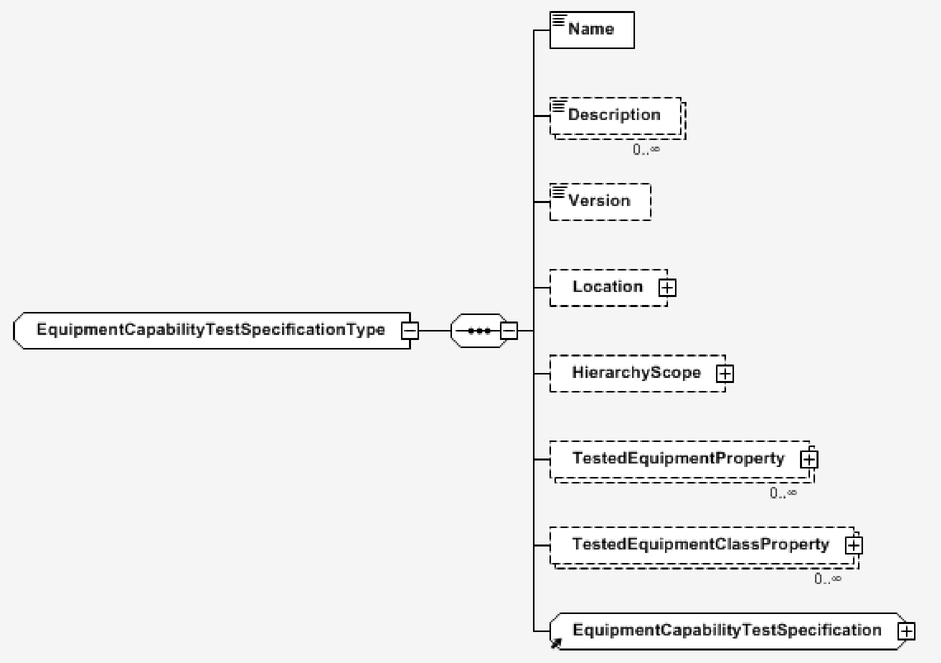
diagram	 EquipmentClassType ID Description 0..∞ Location HierarchyScope EquipmentLevel EquipmentClassProperty 0..∞ EquipmentID 0..∞ EquipmentCapabilityTestSpecificationID 0..∞ EquipmentClass
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentClassType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead --> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="EquipmentLevel" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="EquipmentClassProperty" type="B2MML:EquipmentClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType" minOccurs="0" maxOccurs="unbounded"/>

	<xsd:element name="EquipmentCapabilityTestSpecificationID" type="B2MML:EquipmentCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:EquipmentClass" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

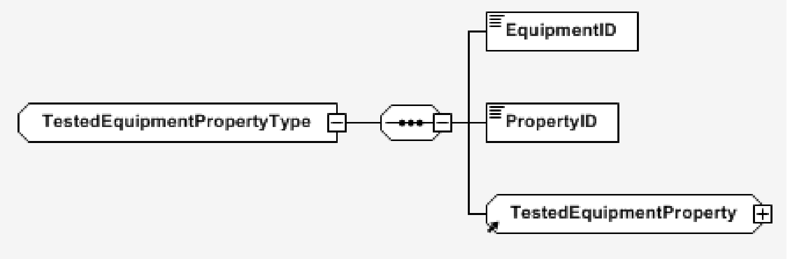
complexType **EquipmentClassPropertyType**

diagram	 ID Description 0..∞ Value 0..∞ EquipmentClassPropertyType EquipmentClassProperty 0..∞ EquipmentCapabilityTestSpecificationID 0..∞ EquipmentClassProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentClassPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentClassProperty" type="B2MML:EquipmentClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="EquipmentCapabilityTestSpecificationID" type="B2MML:EquipmentCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:EquipmentClassProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **EquipmentCapabilityTestSpecificationType**

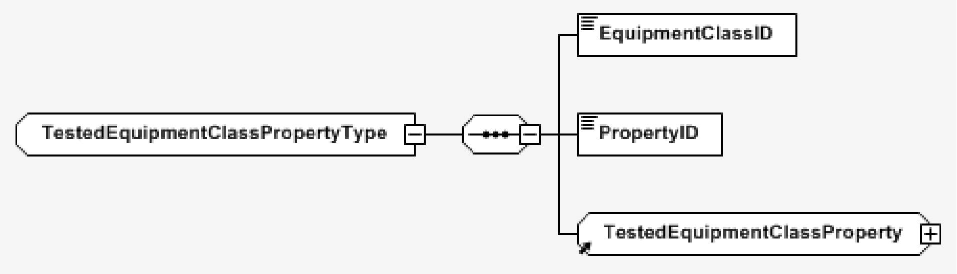
diagram	 EquipmentCapabilityTestSpecificationType Name Description 0..∞ Version Location HierarchyScope TestedEquipmentProperty 0..∞ TestedEquipmentClassProperty 0..∞ EquipmentCapabilityTestSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentCapabilityTestSpecificationType"> <xsd:sequence> <xsd:element name="Name" type="B2MML:NameType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Version" type="B2MML:VersionType" minOccurs="0"/> <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead --> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="TestedEquipmentProperty" type="B2MML:TestedEquipmentPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="TestedEquipmentClassProperty" type="B2MML:TestedEquipmentClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:EquipmentCapabilityTestSpecification" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **TestedEquipmentPropertyType**

diagram	 TestedEquipmentPropertyType EquipmentID PropertyID TestedEquipmentProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

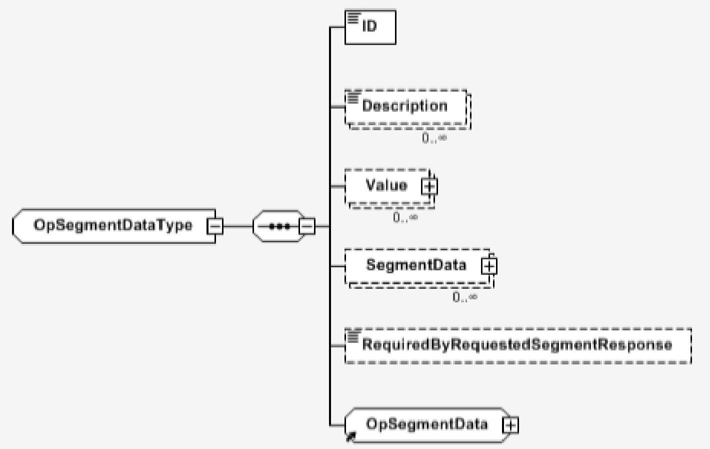
source	`<xsd:complexType name="TestedEquipmentPropertyType">`
	`  <xsd:sequence>`
	`    <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType"/>`
	`    <xsd:element name="PropertyID" type="B2MML:PropertyIDType"/>`
	`    <xsd:group ref="B2MML:TestedEquipmentProperty" minOccurs="0" maxOccurs="1"/>`
	`  </xsd:sequence>`
	`</xsd:complexType>`

complexType **TestedEquipmentClassPropertyType**

diagram	 TestedEquipmentClassPropertyType EquipmentClassID PropertyID TestedEquipmentClassProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

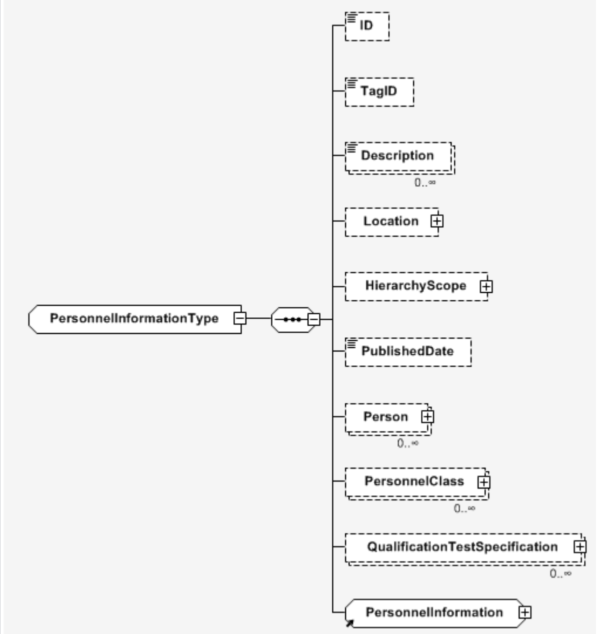
source	`<xsd:complexType name="TestedEquipmentClassPropertyType">`
	`  <xsd:sequence>`
	`    <xsd:element name="EquipmentClassID" type="B2MML:EquipmentClassIDType"/>`
	`    <xsd:element name="PropertyID" type="B2MML:PropertyIDType"/>`
	`    <xsd:group ref="B2MML:TestedEquipmentClassProperty" minOccurs="0" maxOccurs="1"/>`
	`  </xsd:sequence>`
	`</xsd:complexType>`

complexType **OpSegmentDataType**

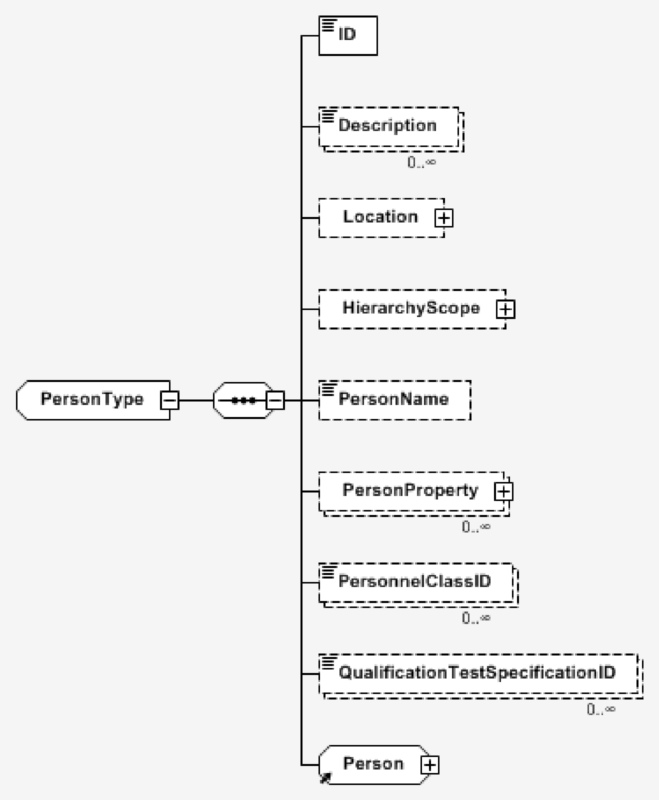
diagram	 OpSegmentDataType ID Description 0..∞ Value 0..∞ SegmentData 0..∞ RequiredByRequestedSegmentResponse OpSegmentData
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="OpSegmentDataType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="SegmentData" type="B2MML:OpSegmentDataType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="RequiredByRequestedSegmentResponse" type="B2MML:RequiredByRequestedSegmentResponseType" minOccurs="0"/>
	    <xsd:group ref="B2MML:OpSegmentData" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

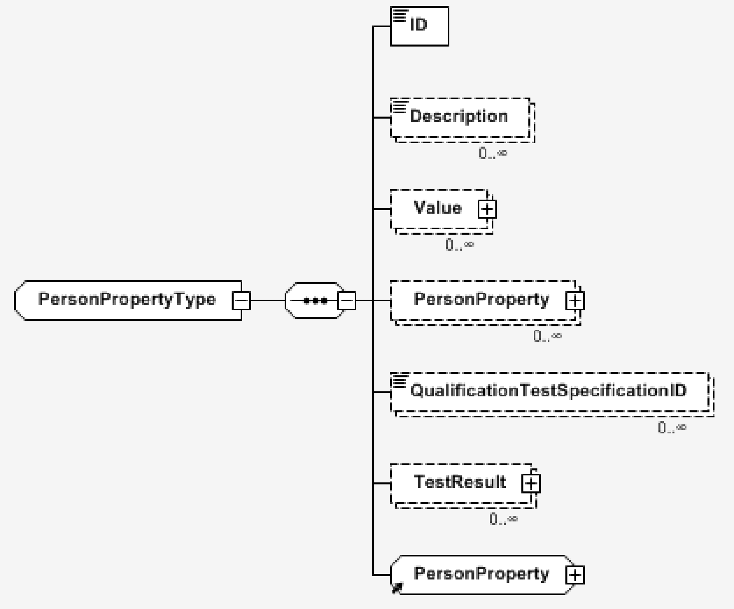
complexType **PersonnelInformationType**

diagram	 ID TagID Description 0..∞ Location HierarchyScope PersonnelInformationType PublishedDate Person 0..∞ PersonnelClass 0..∞ QualificationTestSpecification 0..∞ PersonnelInformation
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonnelInformationType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/> <xsd:element type="xsd:string" name="TagID" minOccurs="0" /> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead --> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PublishedDate" type="B2MML:PublishedDateType" minOccurs="0"/> <xsd:element name="Person" type="B2MML:PersonType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PersonnelClass" type="B2MML:PersonnelClassType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="QualificationTestSpecification" type="B2MML:QualificationTestSpecificationType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PersonnelInformation" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PersonType**

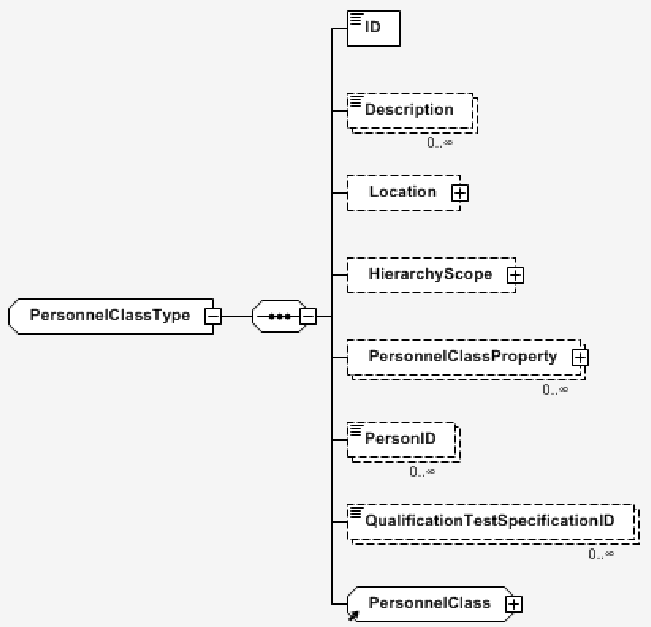
diagram	 ID Description 0..∞ Location HierarchyScope PersonType PersonName PersonProperty 0..∞ PersonnelClassID 0..∞ QualificationTestSpecificationID 0..∞ Person
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead --> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PersonName" type="B2MML:PersonNameType" minOccurs="0"/> <xsd:element name="PersonProperty" type="B2MML:PersonPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PersonnelClassID" type="B2MML:PersonnelClassIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="QualificationTestSpecificationID" type="B2MML:QualificationTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:Person" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PersonPropertyType**

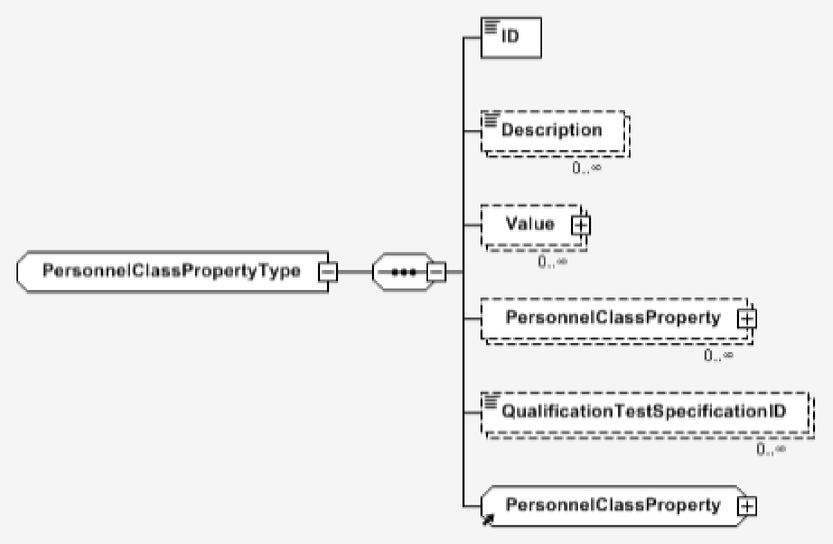
diagram	 ID Description 0..∞ Value 0..∞ PersonPropertyType PersonProperty 0..∞ QualificationTestSpecificationID 0..∞ TestResult 0..∞ PersonProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="PersonPropertyType">
	 <xsd:sequence>
	  <xsd:element name="ID" type="B2MML:IdentifierType"/>
	  <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	  <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/>
	  <xsd:element name="PersonProperty" type="B2MML:PersonPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	  <xsd:element name="QualificationTestSpecificationID" type="B2MML:QualificationTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/>
	  <xsd:element name="TestResult" type="B2MML:TestResultType" minOccurs="0" maxOccurs="unbounded"/>
	  <xsd:group ref="B2MML:PersonProperty" minOccurs="0" maxOccurs="1"/>
	 </xsd:sequence>
	</xsd:complexType>

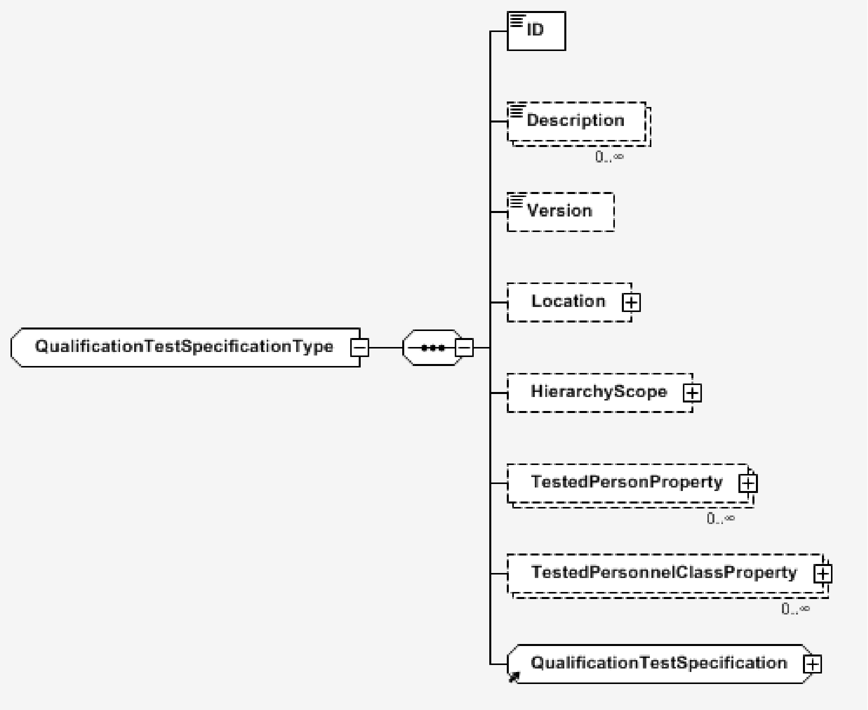
complexType **PersonnelClassType**

diagram	 PersonnelClassType ID Description 0..∞ Location HierarchyScope PersonnelClassProperty 0..∞ PersonID 0..∞ QualificationTestSpecificationID 0..∞ PersonnelClass
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonnelClassType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PersonnelClassProperty" type="B2MML:PersonnelClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PersonID" type="B2MML:PersonIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="QualificationTestSpecificationID" type="B2MML:QualificationTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PersonnelClass" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PersonnelClassPropertyType**

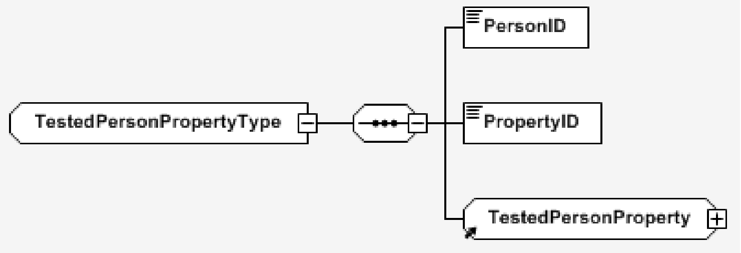
diagram	 PersonnelClassPropertyType ID Description 0..∞ Value 0..∞ PersonnelClassProperty 0..∞ QualificationTestSpecificationID 0..∞ PersonnelClassProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonnelClassPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PersonnelClassProperty" type="B2MML:PersonnelClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="QualificationTestSpecificationID" type="B2MML:QualificationTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PersonnelClassProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **QualificationTestSpecificationType**

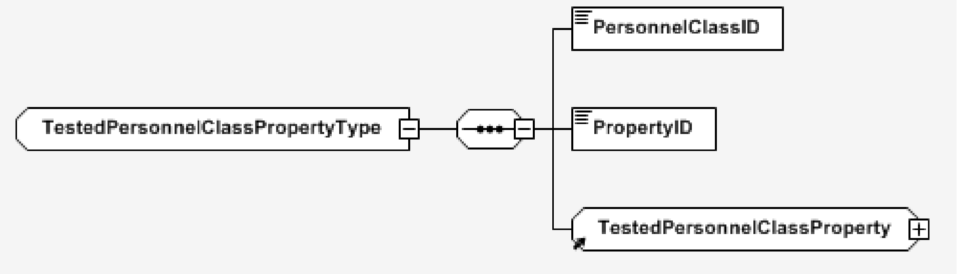
diagram	 ID Description 0..∞ Version Location QualificationTestSpecificationType HierarchyScope TestedPersonProperty 0..∞ TestedPersonnelClassProperty 0..∞ QualificationTestSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="QualificationTestSpecificationType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Version" type="B2MML:VersionType" minOccurs="0"/>
	    <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead -->
	    <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/>
	    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	    <xsd:element name="TestedPersonProperty" type="B2MML:TestedPersonPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="TestedPersonnelClassProperty" type="B2MML:TestedPersonnelClassPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:QualificationTestSpecification" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

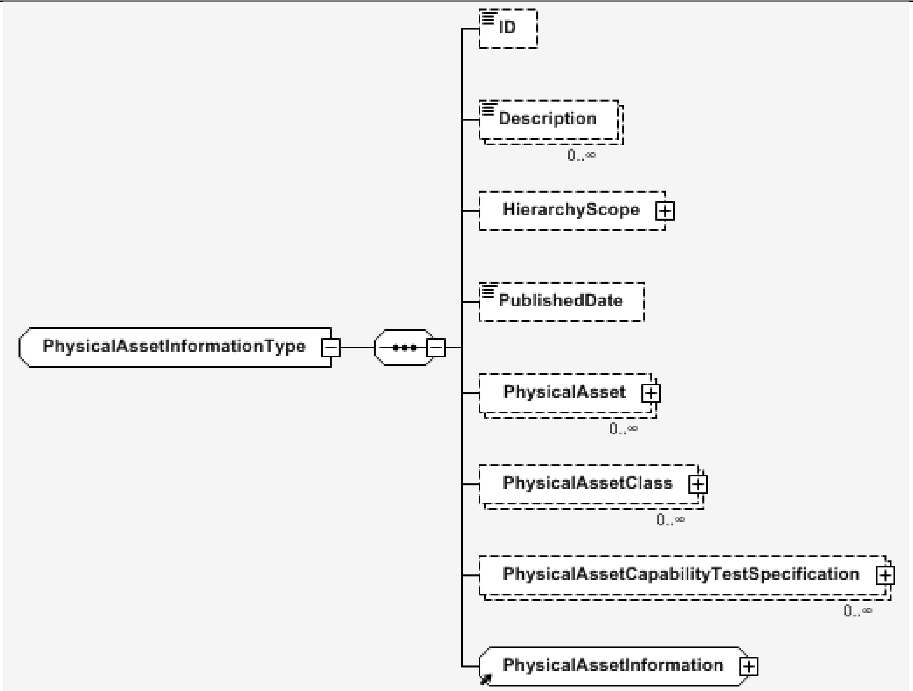
complexType **TestedPersonPropertyType**

diagram	 TestedPersonPropertyType PersonID PropertyID TestedPersonProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="TestedPersonPropertyType"> <xsd:sequence> <xsd:element name="PersonID" type="B2MML:PersonIDType"/> <xsd:element name="PropertyID" type="B2MML:PropertyIDType"/> <xsd:group ref="B2MML:TestedPersonProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **TestedPersonnelClassPropertyType**

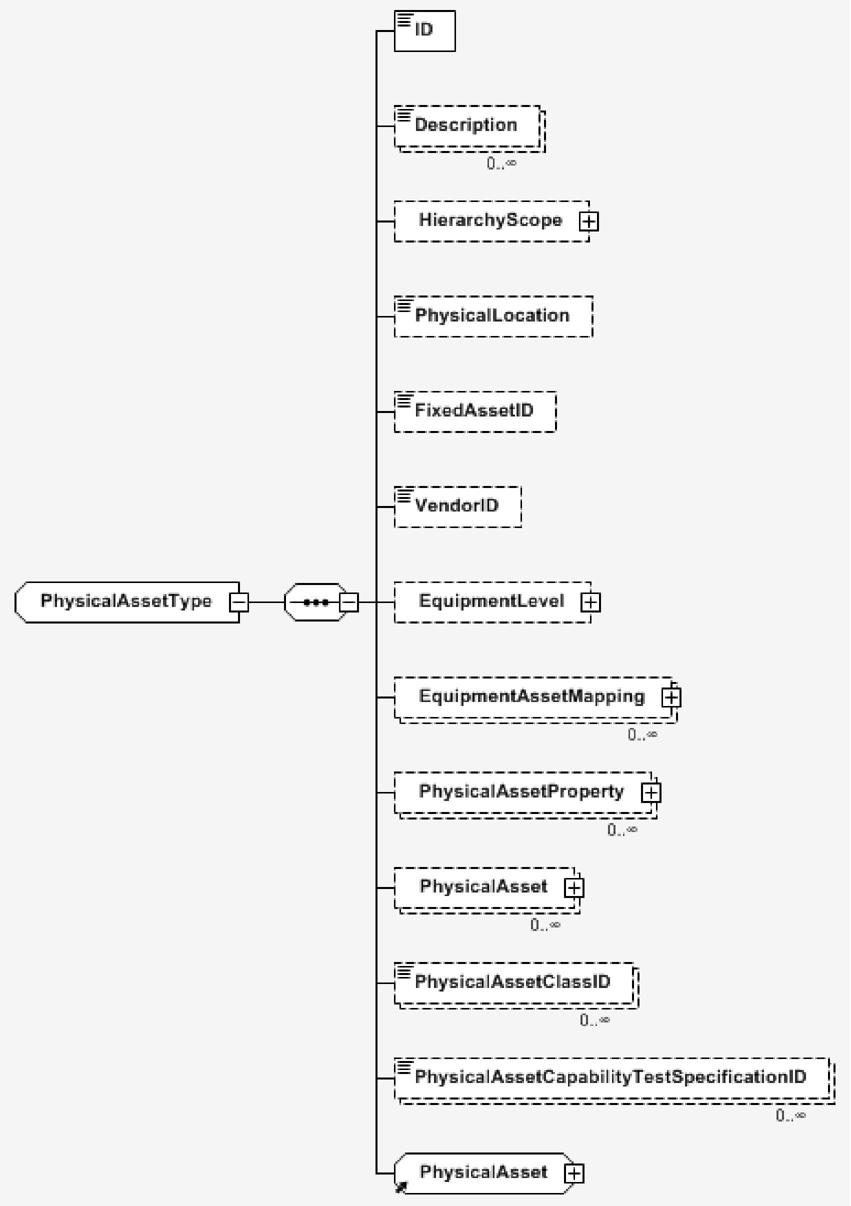
diagram	 TestedPersonnelClassPropertyType PersonnelClassID PropertyID TestedPersonnelClassProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="TestedPersonnelClassPropertyType"> <xsd:sequence> <xsd:element name="PersonnelClassID" type="B2MML:PersonnelClassIDType"/> <xsd:element name="PropertyID" type="B2MML:PropertyIDType"/> <xsd:group ref="B2MML:TestedPersonnelClassProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PhysicalAssetInformationType**

diagram	 ID Description 0..∞ HierarchyScope PublishedDate PhysicalAssetInformationType PhysicalAsset 0..∞ PhysicalAssetClass 0..∞ PhysicalAssetCapabilityTestSpecification 0..∞ PhysicalAssetInformation
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

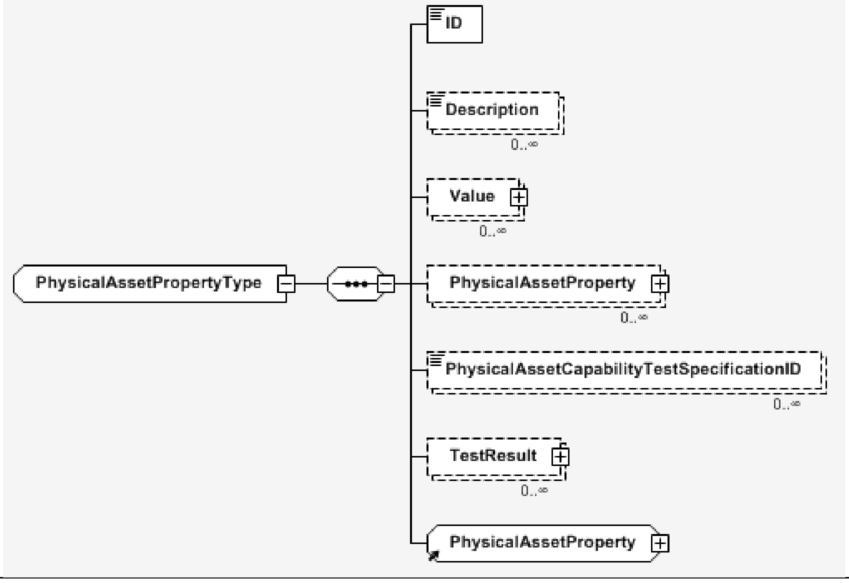
source	<xsd:complexType name="PhysicalAssetInformationType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	    <xsd:element name="PublishedDate" type="B2MML:PublishedDateType" minOccurs="0"/>
	    <xsd:element name="PhysicalAsset" type="B2MML:PhysicalAssetType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="PhysicalAssetClass" type="B2MML:PhysicalAssetClassType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="PhysicalAssetCapabilityTestSpecification" type="B2MML:PhysicalAssetCapabilityTestSpecificationType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:PhysicalAssetInformation" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **PhysicalAssetType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PhysicalLocation" type="B2MML:IdentifierType" minOccurs="0"/> <xsd:element name="FixedAssetID" type="B2MML:IdentifierType" minOccurs="0"/>

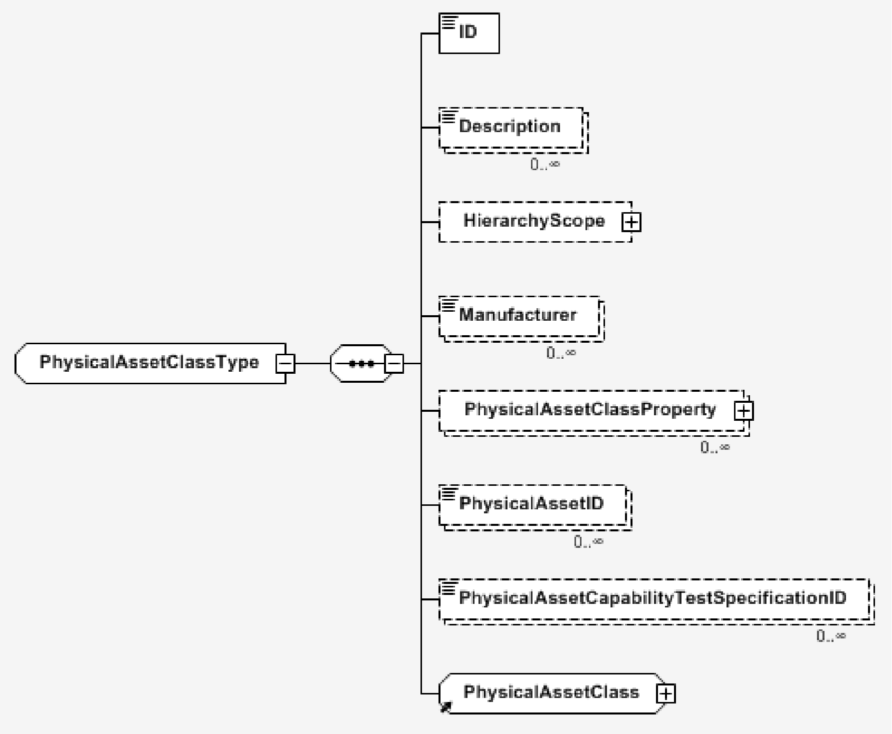
	<xsd:element name="VendorID" type="B2MML:IdentifierType" minOccurs="0"/> <xsd:element name="EquipmentLevel" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="EquipmentAssetMapping" type="B2MML:EquipmentAssetMappingType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetProperty" type="B2MML:PhysicalAssetPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAsset" type="B2MML:PhysicalAssetType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetClassID" type="B2MML:PhysicalAssetClassIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetCapabilityTestSpecificationID" type="B2MML:PhysicalAssetCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PhysicalAsset" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PhysicalAssetPropertyType**

diagram	 PhysicalAssetPropertyType ID Description 0..∞ Value 0..∞ PhysicalAssetProperty 0..∞ PhysicalAssetCapabilityTestSpecificationID 0..∞ TestResult 0..∞ PhysicalAssetProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetProperty" type="B2MML:PhysicalAssetPropertyType" minOccurs="0" maxOccurs="unbounded"/>

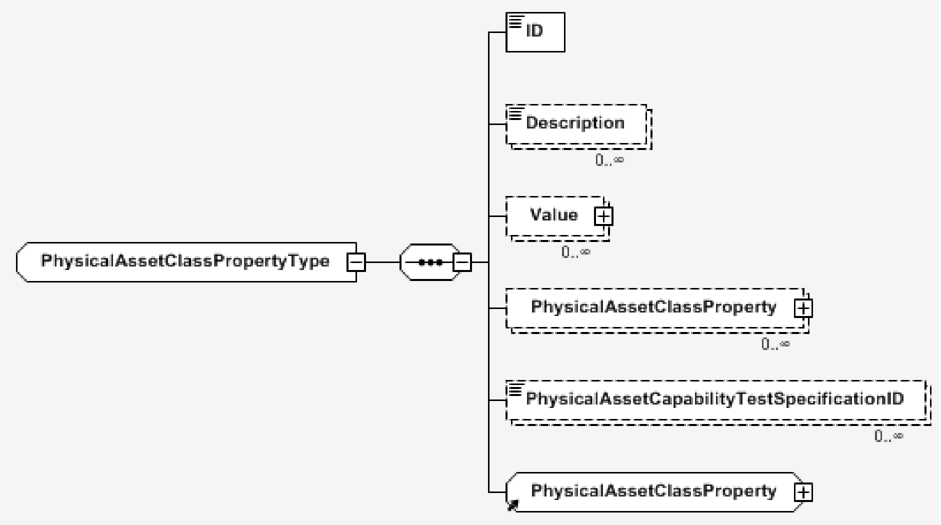
	<xsd:element name="PhysicalAssetCapabilityTestSpecificationID" type="B2MML:PhysicalAssetCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="TestResult" type="B2MML:TestResultType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PhysicalAssetProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PhysicalAssetClassType**

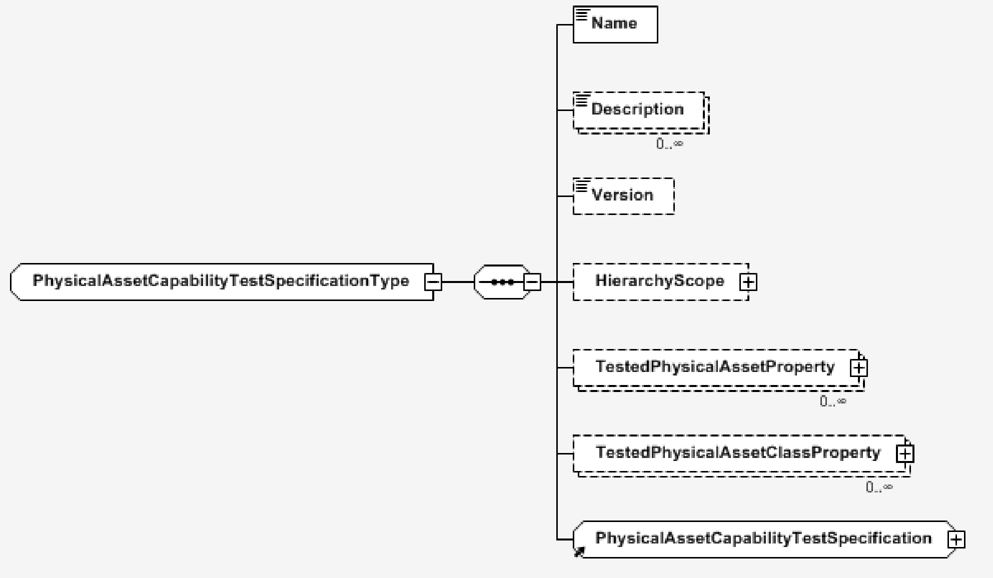
diagram	 ID Description 0..∞ HierarchyScope Manufacturer 0..∞ PhysicalAssetClassType PhysicalAssetClassProperty 0..∞ PhysicalAssetID 0..∞ PhysicalAssetCapabilityTestSpecificationID 0..∞ PhysicalAssetClass
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetClassType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="Manufacturer" type="B2MML:NameType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetClassProperty" type="B2MML:PhysicalAssetClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetID" type="B2MML:PhysicalAssetIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetCapabilityTestSpecificationID"

	type="B2MML:PhysicalAssetCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PhysicalAssetClass" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PhysicalAssetClassPropertyType**

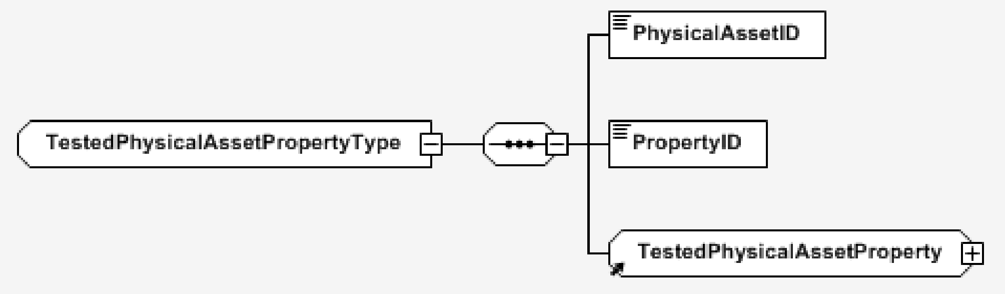
diagram	 PhysicalAssetClassPropertyType ID Description 0..∞ Value 0..∞ PhysicalAssetClassProperty 0..∞ PhysicalAssetCapabilityTestSpecificationID 0..∞ PhysicalAssetClassProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PhysicalAssetClassPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetClassProperty" type="B2MML:PhysicalAssetClassPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PhysicalAssetCapabilityTestSpecificationID" type="B2MML:PhysicalAssetCapabilityTestSpecificationIDType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PhysicalAssetClassProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PhysicalAssetCapabilityTestSpecificationType**

diagram	 Name Description 0..∞ Version PhysicalAssetCapabilityTestSpecificationType HierarchyScope TestedPhysicalAssetProperty 0..∞ TestedPhysicalAssetClassProperty 0..∞ PhysicalAssetCapabilityTestSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

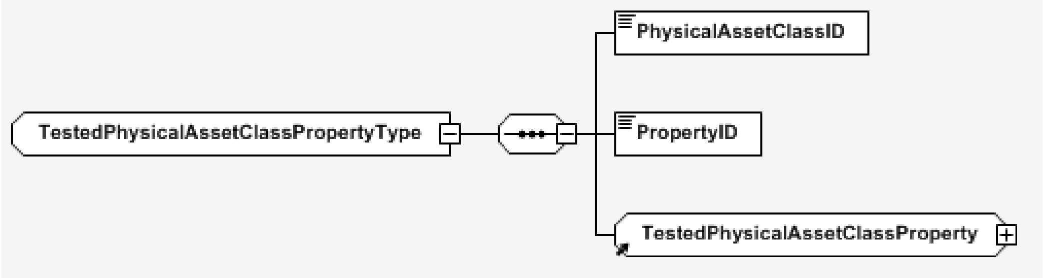
source	<xsd:complexType name="PhysicalAssetCapabilityTestSpecificationType">
	  <xsd:sequence>
	    <xsd:element name="Name" type="B2MML:NameType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Version" type="B2MML:VersionType" minOccurs="0"/>
	    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	    <xsd:element name="TestedPhysicalAssetProperty" type="B2MML:TestedPhysicalAssetPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="TestedPhysicalAssetClassProperty" type="B2MML:TestedPhysicalAssetClassPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:PhysicalAssetCapabilityTestSpecification" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **TestedPhysicalAssetPropertyType**

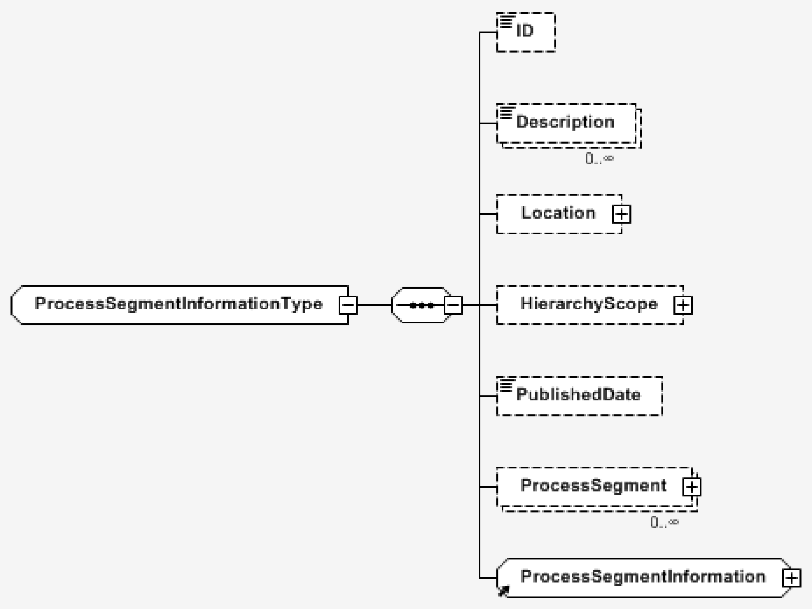
diagram	 PhysicalAssetID TestedPhysicalAssetPropertyType PropertyID TestedPhysicalAssetProperty

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="TestedPhysicalAssetPropertyType"> <xsd:sequence> <xsd:element name="PhysicalAssetID" type="B2MML:PhysicalAssetIDType"/> <xsd:element name="PropertyID" type="B2MML:PropertyIDType"/> <xsd:group ref="B2MML:TestedPhysicalAssetProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

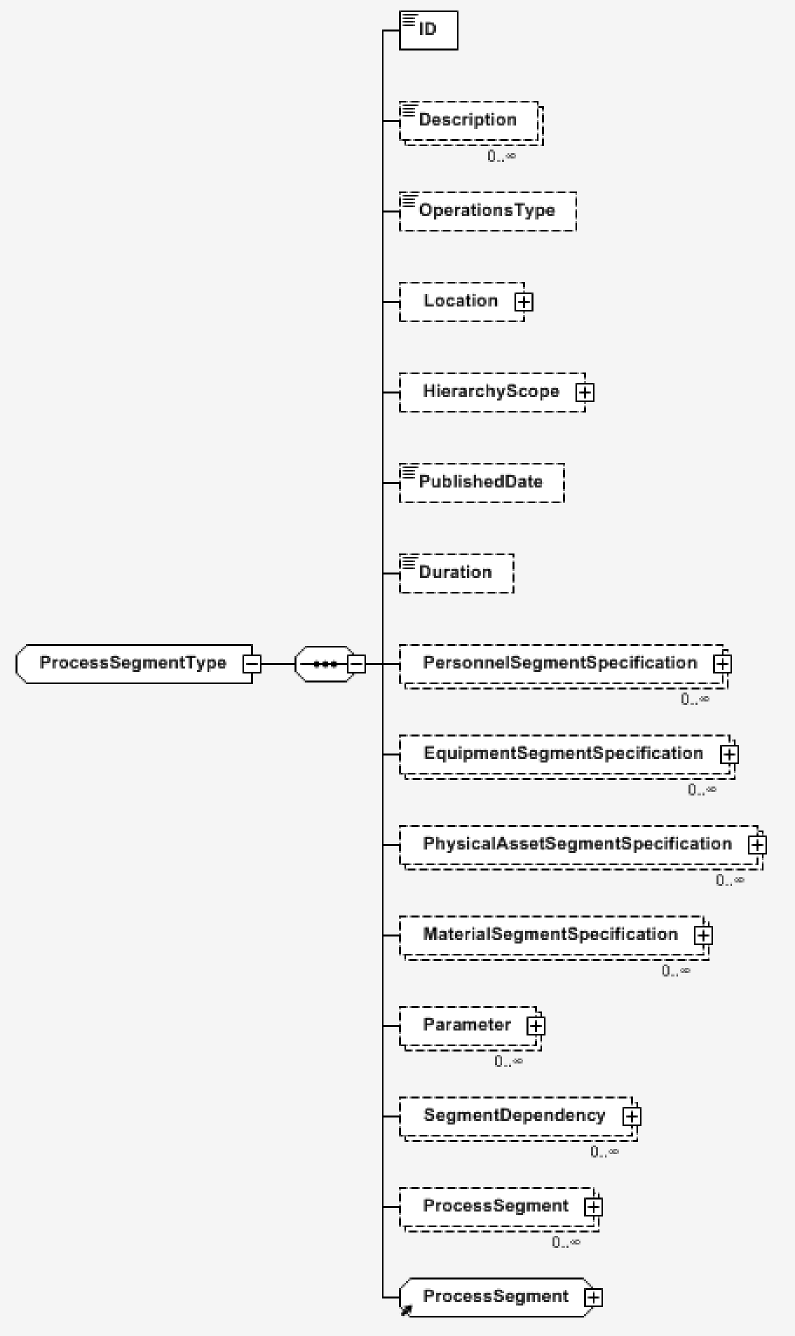
complexType **TestedPhysicalAssetClassPropertyType**

diagram	 TestedPhysicalAssetClassPropertyType PhysicalAssetClassID PropertyID TestedPhysicalAssetClassProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="TestedPhysicalAssetClassPropertyType"> <xsd:sequence> <xsd:element name="PhysicalAssetClassID" type="B2MML:PhysicalAssetClassIDType"/> <xsd:element name="PropertyID" type="B2MML:PropertyIDType"/> <xsd:group ref="B2MML:TestedPhysicalAssetClassProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **ProcessSegmentInformationType**

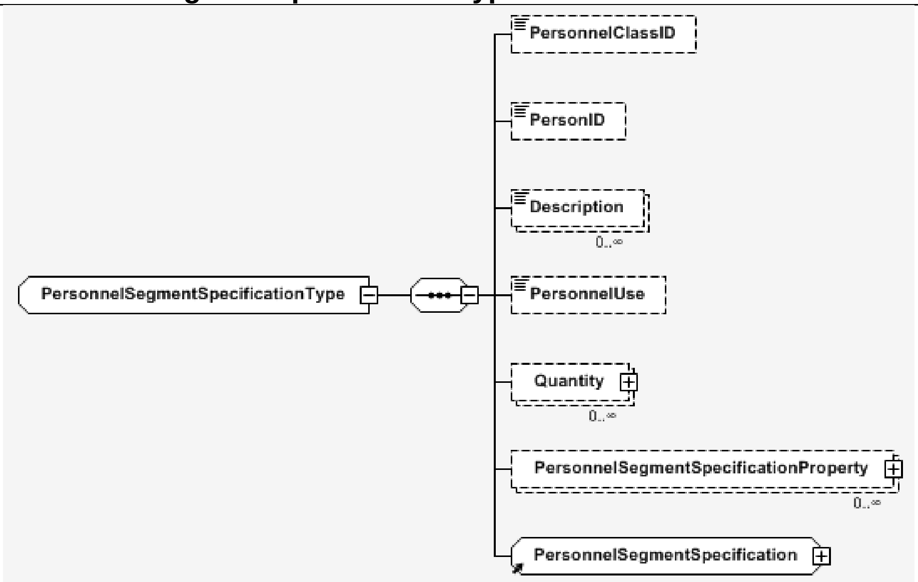
diagram	 ProcessSegmentInformationType ID Description 0..∞ Location HierarchyScope PublishedDate ProcessSegment 0..∞ ProcessSegmentInformation
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ProcessSegmentInformationType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead --> <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/> <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/> <xsd:element name="PublishedDate" type="B2MML:PublishedDateType" minOccurs="0"/> <xsd:element name="ProcessSegment" type="B2MML:ProcessSegmentType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:ProcessSegmentInformation" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **ProcessSegmentType**

diagram	 ProcessSegmentType ID Description 0..∞ OperationsType Location HierarchyScope PublishedDate Duration PersonnelSegmentSpecification 0..∞ EquipmentSegmentSpecification 0..∞ PhysicalAssetSegmentSpecification 0..∞ MaterialSegmentSpecification 0..∞ Parameter 0..∞ SegmentDependency 0..∞ ProcessSegment 0..∞ ProcessSegment
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="ProcessSegmentType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="OperationsType" type="B2MML:OperationsTypeType" minOccurs="0"/>

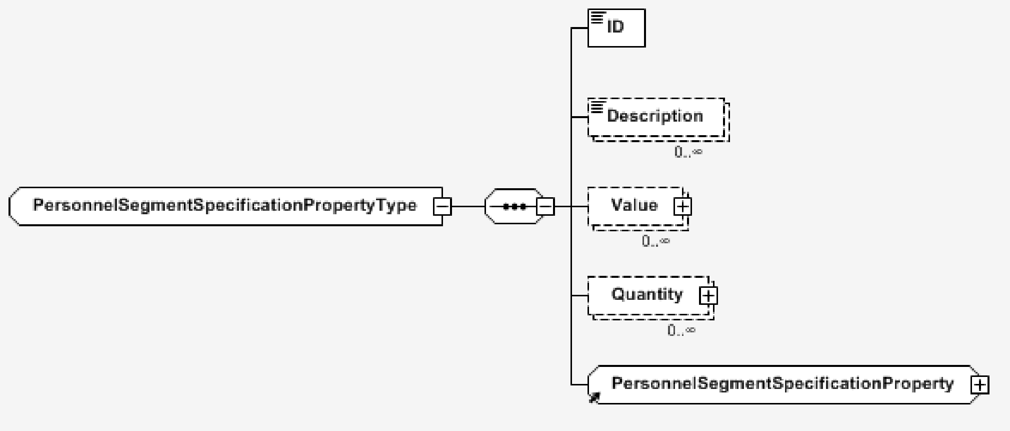
```
    <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use HierarchyScope instead -->
    <xsd:element name="Location" type="B2MML:LocationType" minOccurs="0"/>
    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
    <xsd:element name="PublishedDate" type="B2MML:PublishedDateType" minOccurs="0"/>
    <xsd:element name="Duration" type="B2MML:DurationType" minOccurs="0"/>
    <xsd:element name="PersonnelSegmentSpecification" type="B2MML:PersonnelSegmentSpecificationType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="EquipmentSegmentSpecification" type="B2MML:EquipmentSegmentSpecificationType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="PhysicalAssetSegmentSpecification" type="B2MML:PhysicalAssetSegmentSpecificationType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="MaterialSegmentSpecification" type="B2MML:MaterialSegmentSpecificationType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Parameter" type="B2MML:ParameterType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="SegmentDependency" type="B2MML:SegmentDependencyType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="ProcessSegment" type="B2MML:ProcessSegmentType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:group ref="B2MML:ProcessSegment" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
 </xsd:complexType>
```

complexType **PersonnelSegmentSpecificationType**

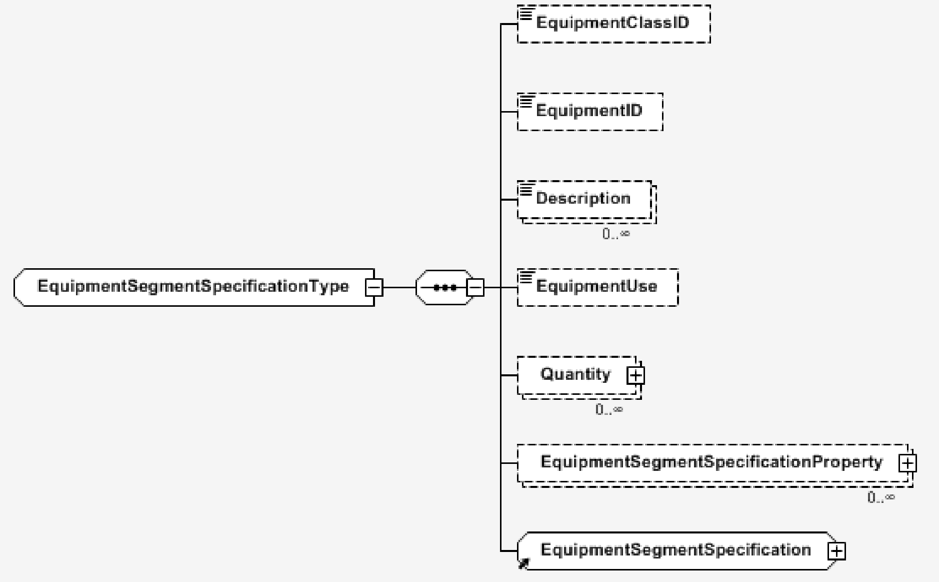
diagram	 PersonnelSegmentSpecificationType PersonnelClassID PersonID Description 0..∞ PersonnelUse Quantity 0..∞ PersonnelSegmentSpecificationProperty 0..∞ PersonnelSegmentSpecification

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonnelSegmentSpecificationType"> <xsd:sequence> <xsd:element name="PersonnelClassID" type="B2MML:PersonnelClassIDType" minOccurs="0"/> <xsd:element name="PersonID" type="B2MML:PersonIDType" minOccurs="0"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PersonnelUse" type="B2MML:CodeType" minOccurs="0"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="PersonnelSegmentSpecificationProperty" type="B2MML:PersonnelSegmentSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PersonnelSegmentSpecification" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PersonnelSegmentSpecificationPropertyType**

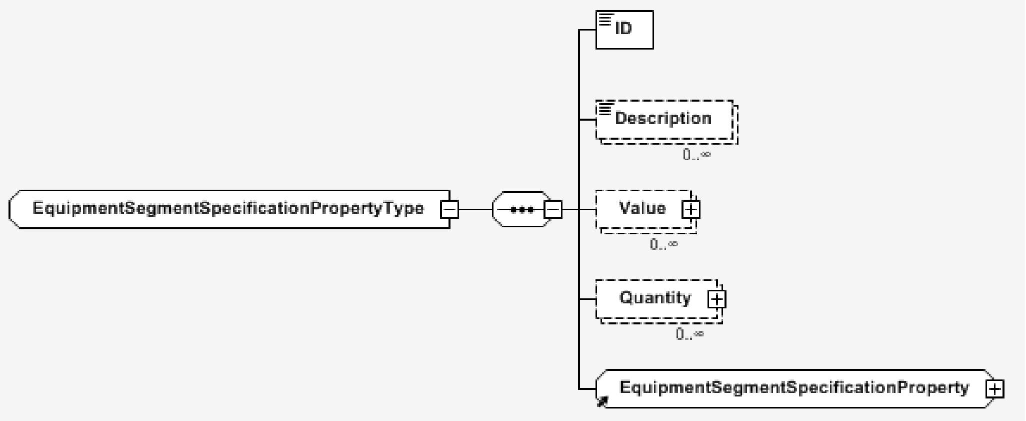
diagram	 PersonnelSegmentSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ PersonnelSegmentSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="PersonnelSegmentSpecificationPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:PersonnelSegmentSpecificationProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **EquipmentSegmentSpecificationType**

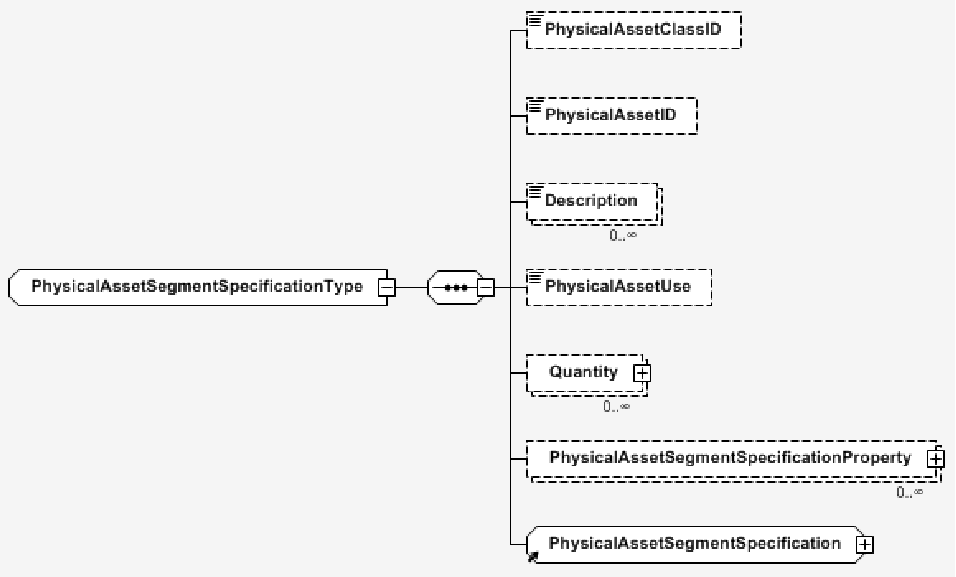
diagram	 EquipmentSegmentSpecificationType EquipmentClassID EquipmentID Description 0..∞ EquipmentUse Quantity 0..∞ EquipmentSegmentSpecificationProperty 0..∞ EquipmentSegmentSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="EquipmentSegmentSpecificationType">
	  <xsd:sequence>
	    <xsd:element name="EquipmentClassID" type="B2MML:EquipmentClassIDType" minOccurs="0"/>
	    <xsd:element name="EquipmentID" type="B2MML:EquipmentIDType" minOccurs="0"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentUse" type="B2MML:CodeType" minOccurs="0"/>
	    <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="EquipmentSegmentSpecificationProperty" type="B2MML:EquipmentSegmentSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:EquipmentSegmentSpecification" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **EquipmentSegmentSpecificationPropertyType**

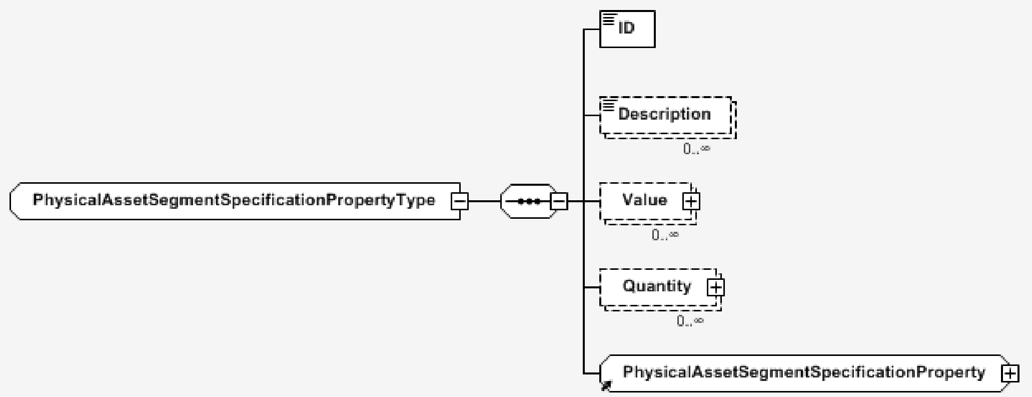
diagram	 EquipmentSegmentSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ EquipmentSegmentSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="EquipmentSegmentSpecificationPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:EquipmentSegmentSpecificationProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

complexType **PhysicalAssetSegmentSpecificationType**

diagram	 PhysicalAssetSegmentSpecificationType PhysicalAssetClassID PhysicalAssetID Description 0..∞ PhysicalAssetUse Quantity 0..∞ PhysicalAssetSegmentSpecificationProperty 0..∞ PhysicalAssetSegmentSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="PhysicalAssetSegmentSpecificationType">
	  <xsd:sequence>
	    <xsd:element name="PhysicalAssetClassID" type="B2MML:PhysicalAssetClassIDType" minOccurs="0"/>
	    <xsd:element name="PhysicalAssetID" type="B2MML:PhysicalAssetIDType" minOccurs="0"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="PhysicalAssetUse" type="B2MML:CodeType" minOccurs="0"/>
	    <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="PhysicalAssetSegmentSpecificationProperty" type="B2MML:PhysicalAssetSegmentSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:PhysicalAssetSegmentSpecification" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

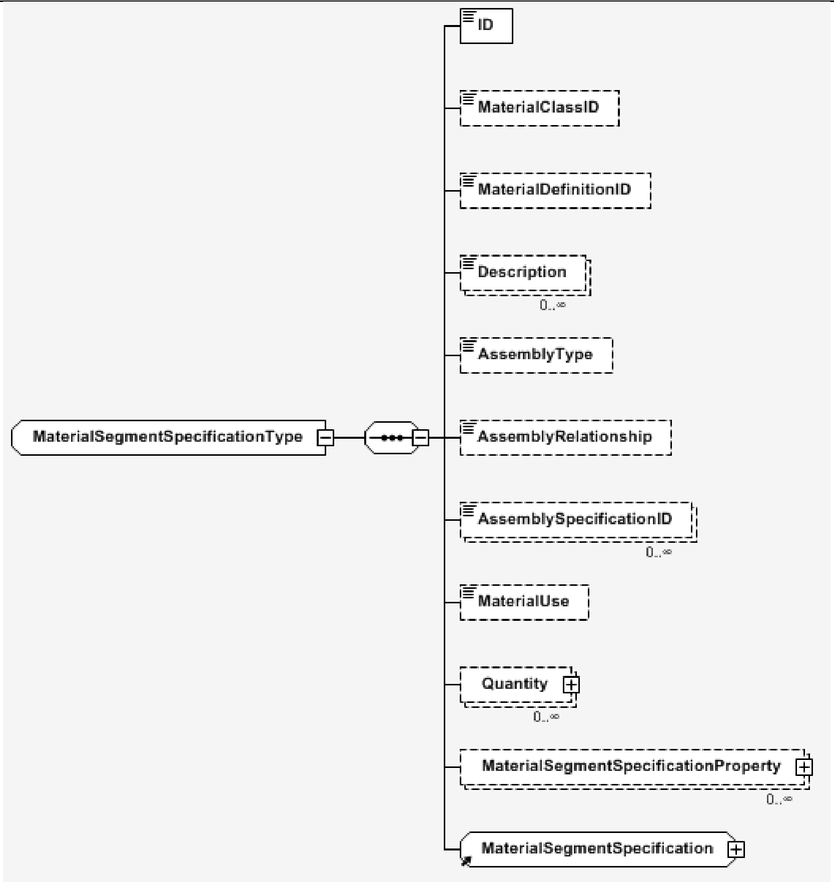
complexType **PhysicalAssetSegmentSpecificationPropertyType**

diagram	 ID Description 0..∞ PhysicalAssetSegmentSpecificationPropertyType Value 0..∞ Quantity 0..∞ PhysicalAssetSegmentSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="PhysicalAssetSegmentSpecificationPropertyType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:PhysicalAssetSegmentSpecificationProperty" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>

	</xsd:complexType>

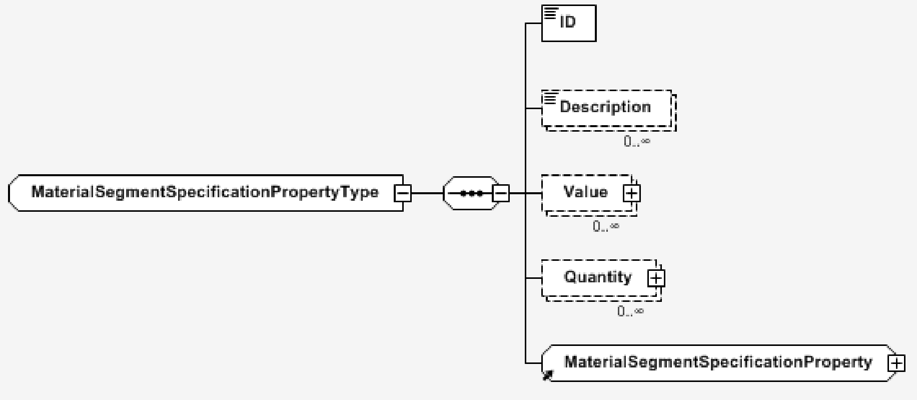
complexType **MaterialSegmentSpecificationType**

diagram	 MaterialSegmentSpecificationType ID MaterialClassID MaterialDefinitionID Description 0..∞ AssemblyType AssemblyRelationship AssemblySpecificationID 0..∞ MaterialUse Quantity 0..∞ MaterialSegmentSpecificationProperty 0..∞ MaterialSegmentSpecification
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="MaterialSegmentSpecificationType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="MaterialClassID" type="B2MML:MaterialClassIDType" minOccurs="0"/>
	    <xsd:element name="MaterialDefinitionID" type="B2MML:MaterialDefinitionIDType" minOccurs="0"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="AssemblyType" type="B2MML:AssemblyTypeType" minOccurs="0"/>
	    <xsd:element name="AssemblyRelationship" type="B2MML:AssemblyRelationshipType" minOccurs="0"/>
	    <xsd:element name="AssemblySpecificationID" type="B2MML:IdentifierType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="MaterialUse" type="B2MML:MaterialUseType" minOccurs="0"/>
	    <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0"

	maxOccurs="unbounded"/>
	    <xsd:element name="MaterialSegmentSpecificationProperty" type="B2MML:MaterialSegmentSpecificationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:MaterialSegmentSpecification" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	 </xsd:complexType>

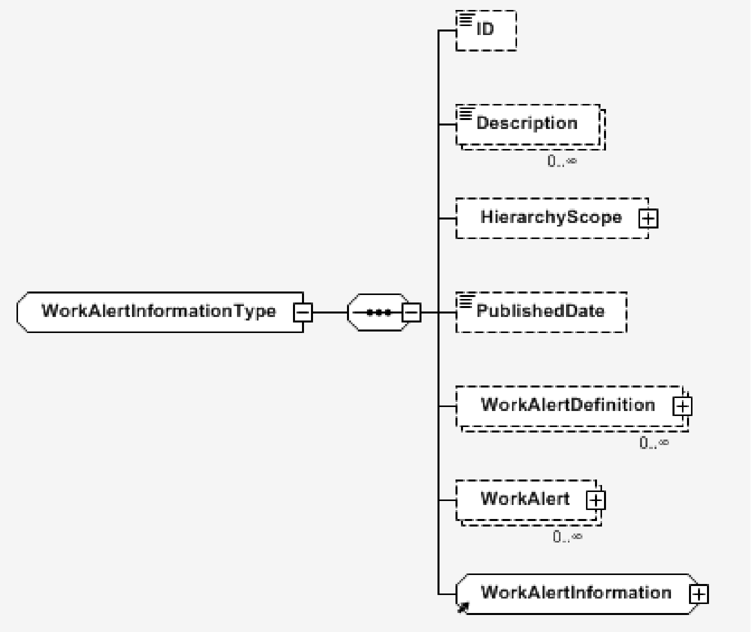
complexType **MaterialSegmentSpecificationPropertyType**

diagram	 MaterialSegmentSpecificationPropertyType ID Description 0..∞ Value 0..∞ Quantity 0..∞ MaterialSegmentSpecificationProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="MaterialSegmentSpecificationPropertyType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Quantity" type="B2MML:QuantityValueType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:MaterialSegmentSpecificationProperty" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	 </xsd:complexType>

complexType **WorkAlertInformationType**

diagram

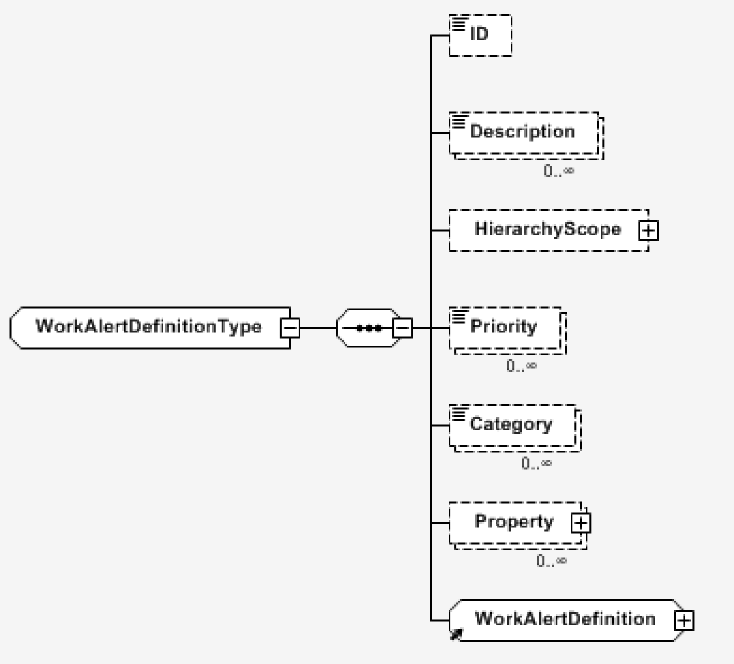


namespace http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source

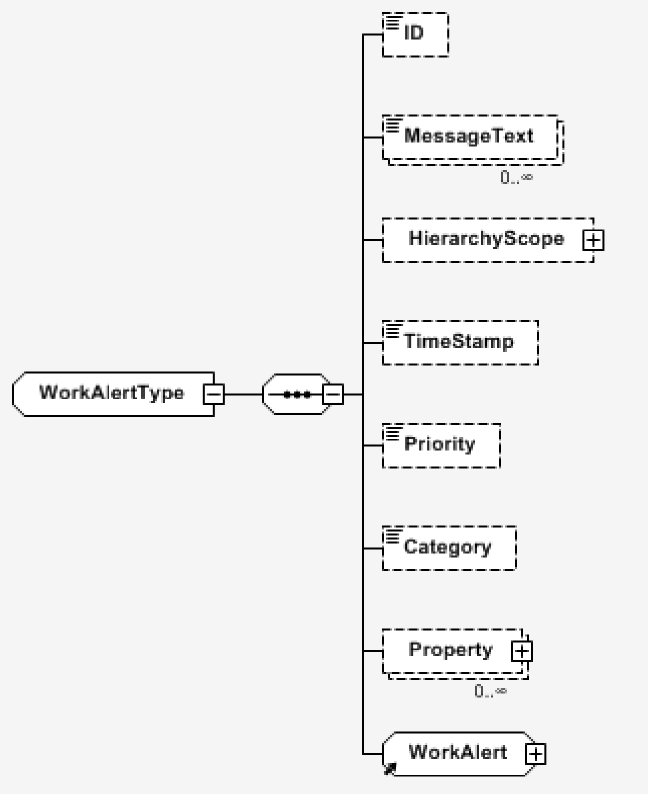
```
<xsd:complexType name="WorkAlertInformationType">
  <xsd:sequence>
    <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0" nillable="true"/>
    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded" nillable="true"/>
    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0" nillable="true"/>
    <xsd:element name="PublishedDate" type="B2MML:PublishedDateType" minOccurs="0" nillable="true"/>
    <xsd:element name="WorkAlertDefinition" type="B2MML:WorkAlertDefinitionType" minOccurs="0" maxOccurs="unbounded" nillable="true"/>
    <xsd:element name="WorkAlert" type="B2MML:WorkAlertType" minOccurs="0" maxOccurs="unbounded" nillable="true"/>
    <xsd:group ref="B2MML:WorkAlertInformation" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

complexType **WorkAlertDefinitionType**

diagram	 WorkAlertDefinitionType ID Description 0..∞ HierarchyScope Priority 0..∞ Category 0..∞ Property 0..∞ WorkAlertDefinition
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

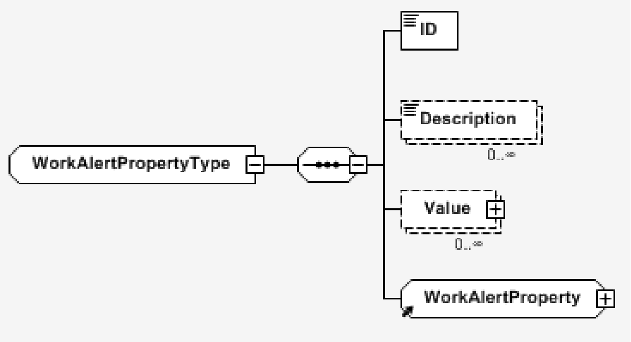
source	<xsd:complexType name="WorkAlertDefinitionType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/>
	    <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	    <xsd:element name="Priority" type="B2MML:PriorityType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Category" type="B2MML:IdentifierType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="Property" type="B2MML:WorkAlertPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:WorkAlertDefinition" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **WorkAlertType**

diagram	 WorkAlertType ID MessageText 0..∞ HierarchyScope TimeStamp Priority Category Property 0..∞ WorkAlert
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML

source	<xsd:complexType name="WorkAlertType">
	  <xsd:sequence>
	    <xsd:element name="ID" type="B2MML:IdentifierType" minOccurs="0"/>
	    <xsd:element name="MessageText" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:element name="HierarchyScope" type="B2MML:HierarchyScopeType" minOccurs="0"/>
	    <xsd:element name="TimeStamp" type="B2MML:StartTimeType" minOccurs="0"/>
	    <xsd:element name="Priority" type="B2MML:PriorityType" minOccurs="0"/>
	    <xsd:element name="Category" type="B2MML:IdentifierType" minOccurs="0"/>
	    <xsd:element name="Property" type="B2MML:WorkAlertPropertyType" minOccurs="0" maxOccurs="unbounded"/>
	    <xsd:group ref="B2MML:WorkAlert" minOccurs="0" maxOccurs="1"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **WorkAlertPropertyType**

diagram	 ID Description 0..∞ WorkAlertPropertyType Value 0..∞ WorkAlertProperty
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
source	<xsd:complexType name="WorkAlertPropertyType"> <xsd:sequence> <xsd:element name="ID" type="B2MML:IdentifierType"/> <xsd:element name="Description" type="B2MML:DescriptionType" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="Value" type="B2MML:ValueType" minOccurs="0" maxOccurs="unbounded"/> <xsd:group ref="B2MML:WorkAlertProperty" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>

simpleType **DurationType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/B2MML
type	duration
source	<xsd:simpleType name="DurationType"> <xsd:restriction base="xsd:duration"/> </xsd:simpleType>

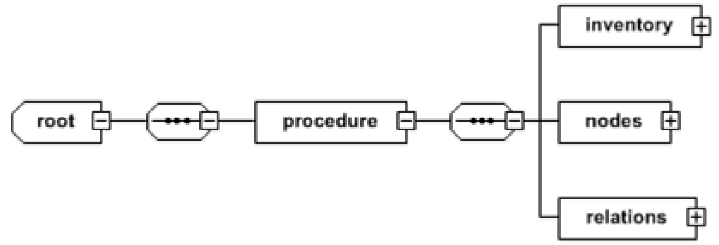
SCHEMA R3D.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.red.org/R3D**

Elements	Complex Types	Simple Types
	root	
inventory		
nodes		
relations		

complexType **root**

diagram	
namespace	http://www.red.org/R3D

source	<xs:complexType name="root">
	  <xs:sequence>
	    <xs:element name="procedure">
	      <xs:complexType>
	        <xs:sequence>
	          <xs:element name="inventory">
	            <xs:complexType>
	              <xs:sequence>
	                <xs:element name="warehouse_element" maxOccurs="unbounded" minOccurs="0">
	                  <xs:complexType>
	                    <xs:sequence>
	                      <xs:element name="attached_resources">
	                        <xs:complexType>
	                          <xs:sequence>
	                            <xs:element name="attached_resource">
	                              <xs:complexType>
	                                <xs:simpleContent>
	                                  <xs:extension base="xs:string">
	                                    <xs:attribute type="xs:string" name="resource_reference" use="optional"/>
	                                    <xs:attribute type="xs:string" name="resource_reference_type" use="optional"/>
	                                  </xs:extension>

```
                  </xs:simpleContent>
                 </xs:complexType>
                </xs:element>
               </xs:sequence>
              </xs:complexType>
             </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="element_id" use="optional"/>
            <xs:attribute type="xs:string" name="type" use="optional"/>
            <xs:attribute type="xs:string" name="code" use="optional"/>
            <xs:attribute type="xs:string" name="name" use="optional"/>
            <xs:attribute type="xs:string" name="short_description"
use="optional"/>
            <xs:attribute type="xs:string" name="description" use="optional"/>
            <xs:attribute type="xs:byte" name="amount" use="optional"/>
           </xs:complexType>
          </xs:element>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
       <xs:element name="nodes">
        <xs:complexType>
         <xs:choice maxOccurs="unbounded" minOccurs="0">
          <xs:element name="root">
           <xs:complexType>
            <xs:simpleContent>
             <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="node_type"/>
              <xs:attribute type="xs:string" name="internal_id"/>
              <xs:attribute type="xs:byte" name="order"/>
             </xs:extension>
            </xs:simpleContent>
           </xs:complexType>
          </xs:element>
          <xs:element name="operation">
           <xs:complexType>
            <xs:simpleContent>
             <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="node_type"/>
              <xs:attribute type="xs:string" name="internal_id"/>
              <xs:attribute type="xs:byte" name="order"/>
              <xs:attribute type="xs:string" name="name"/>
             </xs:extension>
            </xs:simpleContent>
           </xs:complexType>
          </xs:element>
          <xs:element name="step">
           <xs:complexType>
```

```
            <xs:simpleContent>
             <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="node_type"/>
              <xs:attribute type="xs:string" name="internal_id"/>
              <xs:attribute type="xs:byte" name="order"/>
              <xs:attribute type="xs:string" name="name"/>
             </xs:extension>
            </xs:simpleContent>
           </xs:complexType>
          </xs:element>
          <xs:element name="action" maxOccurs="unbounded" minOccurs="0">
           <xs:complexType mixed="true">
            <xs:sequence>
             <xs:element name="descriptive_layers" minOccurs="0">
              <xs:complexType>
               <xs:sequence>
                <xs:element name="descriptive_layer" maxOccurs="unbounded"
minOccurs="0">
                 <xs:complexType>
                  <xs:simpleContent>
                   <xs:extension base="xs:string">
                    <xs:attribute type="xs:string" name="type" use="optional"/>
                    <xs:attribute type="xs:string" name="value" use="optional"/>
                    <xs:attribute type="xs:string"
name="resource_reference_type" use="optional"/>
                   </xs:extension>
                  </xs:simpleContent>
                 </xs:complexType>
                </xs:element>
               </xs:sequence>
              </xs:complexType>
             </xs:element>
             <xs:element name="objects" minOccurs="0">
              <xs:complexType>
               <xs:sequence>
                <xs:element name="objectid">
                 <xs:complexType>
                  <xs:simpleContent>
                   <xs:extension base="xs:string">
                    <xs:attribute type="xs:string" name="entry_id"
use="optional"/>
                    <xs:attribute type="xs:byte" name="amount" use="optional"/>
                   </xs:extension>
                  </xs:simpleContent>
                 </xs:complexType>
                </xs:element>
               </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="objects_to" minOccurs="0">
              <xs:complexType mixed="true">
                <xs:sequence>
                  <xs:element name="objectid" minOccurs="0">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:string">
                          <xs:attribute type="xs:string" name="entry_id"
use="optional"/>
                          <xs:attribute type="xs:byte" name="amount" use="optional"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="objects_with" minOccurs="0">
              <xs:complexType mixed="true">
                <xs:sequence>
                  <xs:element name="objectid" minOccurs="0">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:string">
                          <xs:attribute type="xs:string" name="entry_id"/>
                          <xs:attribute type="xs:byte" name="amount"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute type="xs:string" name="node_type" use="optional"/>
          <xs:attribute type="xs:string" name="internal_id" use="optional"/>
          <xs:attribute type="xs:byte" name="order" use="optional"/>
          <xs:attribute type="xs:string" name="name" use="optional"/>
          <xs:attribute type="xs:string" name="type" use="optional"/>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="relations">
  <xs:complexType>
    <xs:sequence>
```

```
          <xs:element name="consecutive_relation" maxOccurs="unbounded"
minOccurs="0">
           <xs:complexType>
            <xs:simpleContent>
             <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="internal_id" use="optional"/>
              <xs:attribute type="xs:byte" name="order" use="optional"/>
              <xs:attribute type="xs:string" name="source_id" use="optional"/>
              <xs:attribute type="xs:string" name="target_id" use="optional"/>
              <xs:attribute type="xs:string" name="is_direct" use="optional"/>
              <xs:attribute type="xs:byte" name="weight" use="optional"/>
              <xs:attribute type="xs:string" name="type" use="optional"/>
             </xs:extension>
            </xs:simpleContent>
           </xs:complexType>
          </xs:element>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
      <xs:attribute type="xs:string" name="internal_id"/>
      <xs:attribute type="xs:byte" name="order"/>
      <xs:attribute type="xs:string" name="code"/>
      <xs:attribute type="xs:string" name="name"/>
      <xs:attribute type="xs:string" name="short_description"/>
      <xs:attribute type="xs:string" name="description"/>
      <xs:attribute type="xs:dateTime" name="creation_date"/>
      <xs:attribute type="xs:dateTime" name="last_update"/>
      <xs:attribute type="xs:float" name="version"/>
      <xs:attribute type="xs:string" name="confidentiality_level"/>
      <xs:attribute type="xs:string" name="authors"/>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attribute type="xs:byte" name="operation_count"/>
   <xs:attribute type="xs:byte" name="step_count"/>
   <xs:attribute type="xs:byte" name="action_count"/>
   <xs:attribute type="xs:byte" name="relation_count"/>
  </xs:complexType>
</xs:schema>
```

Schema SCORM.xsd

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM**

Elements

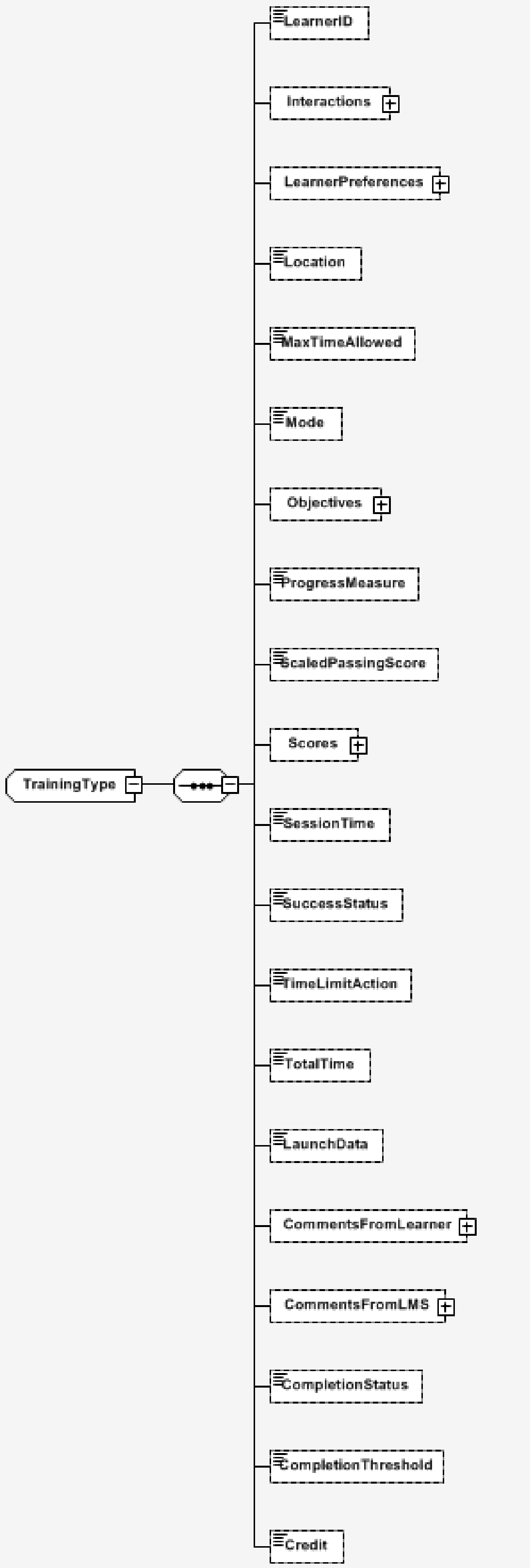
Complex Types

TrainingType
CommentList_Type
Comment_Type
InteractionList_Type
Interaction_Type
LearnerPreferencesList_Type
LearnerPreferences_Type
ObjectivesList_Type
Objective_Type
ScoreList_Type
Score_Type
result_Type

Simple Types

completionStatusEnum
creditEnum
interactionTypeEnum
resultTypeEnum
audioCaptioningEnum
modeEnum
successStatusEnum
timeLimitActionEnum

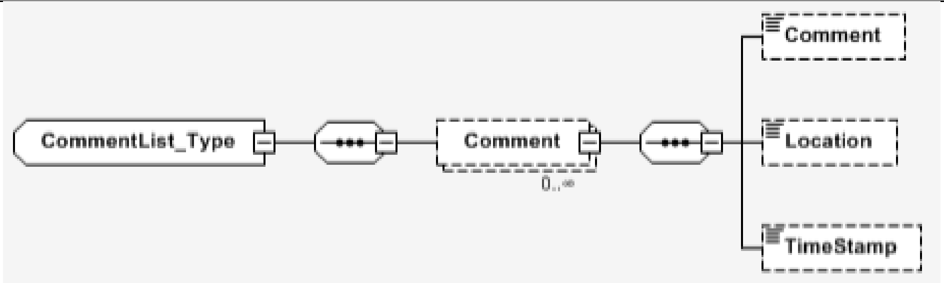
complexType **TrainingType**

diagram	

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

source	<xsd:complexType name="TrainingType">
	    <xsd:sequence>
	        <xsd:element name="LearnerID" type="xsd:string" minOccurs="0"/>
	        <xsd:element name="Interactions" type="InteractionList_Type" minOccurs="0"/>
	        <xsd:element name="LearnerPreferences" type="LearnerPreferencesList_Type" minOccurs="0"/>
	        <xsd:element name="Location" type="xsd:string" minOccurs="0"/>
	        <xsd:element name="MaxTimeAllowed" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Mode" type="modeEnum" minOccurs="0"/>
	        <xsd:element name="Objectives" type="ObjectivesList_Type" minOccurs="0"/>
	        <xsd:element name="ProgressMeasure" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="ScaledPassingScore" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Scores" type="ScoreList_Type" minOccurs="0"/>
	        <xsd:element name="SessionTime" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="SuccessStatus" type="successStatusEnum" minOccurs="0"/>
	        <xsd:element name="TimeLimitAction" type="timeLimitActionEnum" minOccurs="0"/>
	        <xsd:element name="TotalTime" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="LaunchData" type="xsd:string" minOccurs="0"/>
	        <xsd:element name="CommentsFromLearner" type="CommentList_Type" minOccurs="0"/>
	        <xsd:element name="CommentsFromLMS" type="CommentList_Type" minOccurs="0"/>
	        <xsd:element name="CompletionStatus" type="completionStatusEnum" minOccurs="0"/>
	        <xsd:element name="CompletionThreshold" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Credit" type="creditEnum" minOccurs="0"/>
	    </xsd:sequence>
	</xsd:complexType>

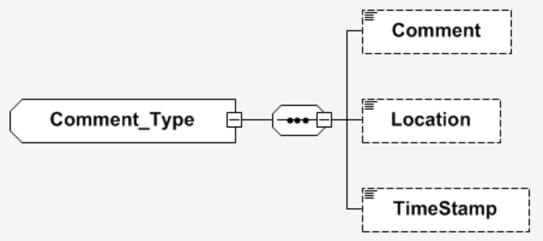
complexType **CommentList_Type**

diagram	 CommentList_Type Comment 0..∞ Comment Location TimeStamp
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

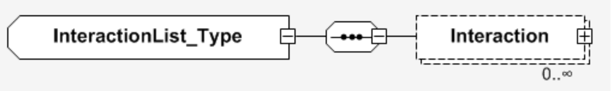
source	<xsd:complexType name="CommentList_Type">
	    <xsd:sequence>
	        <xsd:element name="Comment" type="Comment_Type" minOccurs="0" maxOccurs="unbounded"/>
	    </xsd:sequence>

	</xsd:complexType>

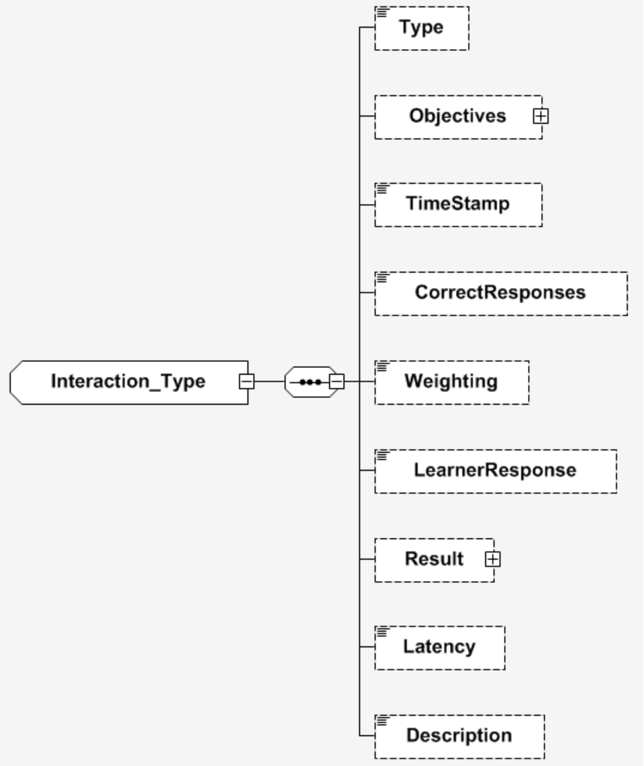
complexType **Comment_Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
source	<xsd:complexType name="Comment_Type"> <xsd:sequence> <xsd:element name="Comment" type="xsd:string" minOccurs="0"/> <xsd:element name="Location" type="xsd:string" minOccurs="0"/> <xsd:element name="TimeStamp" type="xsd:string" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

complexType **InteractionList_Type**

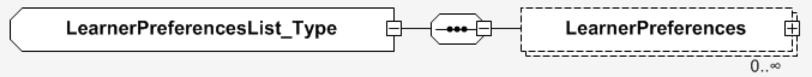
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
source	<xsd:complexType name="InteractionList_Type"> <xsd:sequence> <xsd:element name="Interaction" type="Interaction_Type" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> </xsd:complexType>

complexType **Interaction_Type**

diagram	 Interaction_Type Type Objectives TimeStamp CorrectResponses Weighting LearnerResponse Result Latency Description
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

source	<xsd:complexType name="Interaction_Type">
	  <xsd:sequence>
	    <xsd:element name="Type" type="interactionTypeEnum" minOccurs="0"/>
	    <xsd:element name="Objectives" type="ObjectivesList_Type" minOccurs="0"/>
	    <xsd:element name="TimeStamp" type="xsd:string" minOccurs="0"/>
	    <xsd:element name="CorrectResponses" type="xsd:integer" minOccurs="0"/>
	    <xsd:element name="Weighting" type="xsd:float" minOccurs="0"/>
	    <xsd:element name="LearnerResponse" type="xsd:string" minOccurs="0"/>
	    <xsd:element name="Result" type="result_Type" minOccurs="0"/>
	    <xsd:element name="Latency" type="xsd:float" minOccurs="0"/>
	    <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

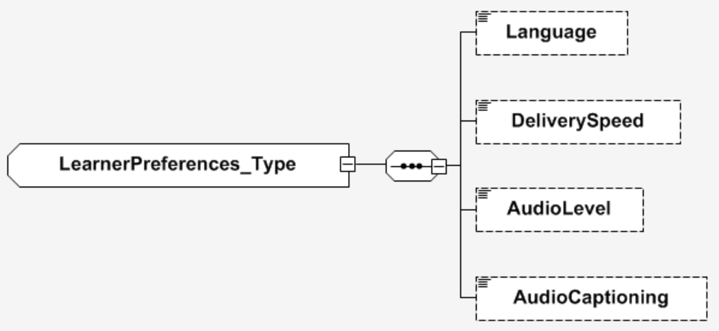
complexType **LearnerPreferencesList_Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

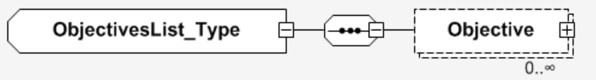
source	<xsd:complexType name="LearnerPreferencesList_Type">
	  <xsd:sequence>
	    <xsd:element name="LearnerPreferences" type="LearnerPreferences_Type" minOccurs="0" maxOccurs="unbounded"/>
	  </xsd:sequence>

	</xsd:complexType>

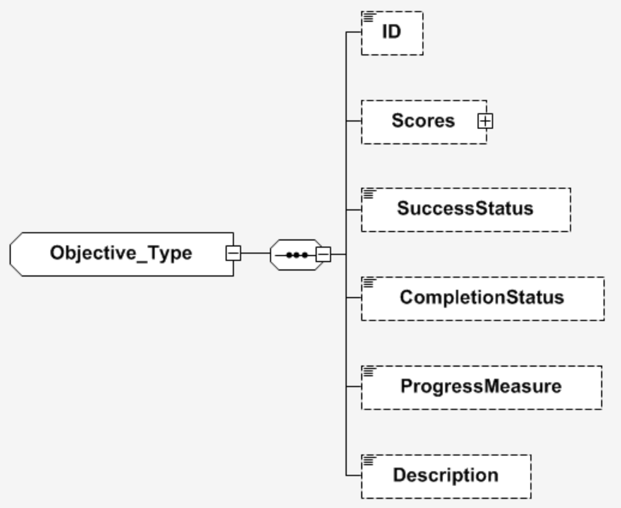
complexType **LearnerPreferences_Type**

diagram	 LearnerPreferences_Type Language DeliverySpeed AudioLevel AudioCaptioning
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
source	<xsd:complexType name="LearnerPreferences_Type"> <xsd:sequence> <xsd:element name="Language" type="xsd:string" minOccurs="0"/> <xsd:element name="DeliverySpeed" type="xsd:float" minOccurs="0"/> <xsd:element name="AudioLevel" type="xsd:float" minOccurs="0"/> <xsd:element name="AudioCaptioning" type="audioCaptioningEnum" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

complexType **ObjectivesList_Type**

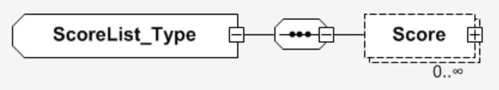
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
source	<xsd:complexType name="ObjectivesList_Type"> <xsd:sequence> <xsd:element name="Objective" type="Objective_Type" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> </xsd:complexType>

complexType **Objective_Type**

diagram	 Objective_Type ID Scores SuccessStatus CompletionStatus ProgressMeasure Description
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

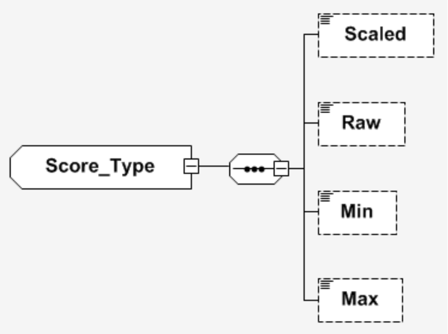
source	<xsd:complexType name="Objective_Type">
	    <xsd:sequence>
	        <xsd:element name="ID" type="xsd:string" minOccurs="0"/>
	        <xsd:element name="Scores" type="ScoreList_Type" minOccurs="0"/>
	        <xsd:element name="SuccessStatus" type="successStatusEnum" minOccurs="0"/>
	        <xsd:element name="CompletionStatus" type="completionStatusEnum" minOccurs="0"/>
	        <xsd:element name="ProgressMeasure" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
	    </xsd:sequence>
	</xsd:complexType>

complexType **ScoreList_Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

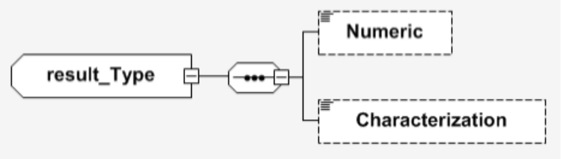
source	<xsd:complexType name="ScoreList_Type">
	    <xsd:sequence>
	        <xsd:element name="Score" type="Score_Type" minOccurs="0" maxOccurs="unbounded"/>
	    </xsd:sequence>
	</xsd:complexType>

complexType **Score_Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

source	<xsd:complexType name="Score_Type">
	    <xsd:sequence>
	        <xsd:element name="Scaled" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Raw" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Min" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Max" type="xsd:float" minOccurs="0"/>
	    </xsd:sequence>
	</xsd:complexType>

complexType **result_Type**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM

source	<xsd:complexType name="result_Type">
	    <xsd:sequence>
	        <xsd:element name="Numeric" type="xsd:float" minOccurs="0"/>
	        <xsd:element name="Characterization" type="resultTypeEnum" minOccurs="0"/>
	    </xsd:sequence>
	</xsd:complexType>

simpleType **completionStatusEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string

source	<xsd:simpleType name="completionStatusEnum" final="restriction">
	    <xsd:restriction base="xsd:string">
	        <xsd:enumeration value="Completed"/>
	        <xsd:enumeration value="Incomplete"/>
	        <xsd:enumeration value="Not attended"/>
	        <xsd:enumeration value="Unknown"/>
	    </xsd:restriction>
	</xsd:simpleType>

simpleType **creditEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string

source	<xsd:simpleType name="creditEnum" final="restriction">
	  <xsd:restriction base="xsd:string">
	    <xsd:enumeration value="Credit"/>
	    <xsd:enumeration value="No-Credit"/>
	  </xsd:restriction>
	</xsd:simpleType>

simpleType **interactionTypeEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string

source	<xsd:simpleType name="interactionTypeEnum" final="restriction">
	  <xsd:restriction base="xsd:string">
	    <xsd:enumeration value="true-false"/>
	    <xsd:enumeration value="choice"/>
	    <xsd:enumeration value="fill-in"/>
	    <xsd:enumeration value="long-fill-in"/>
	    <xsd:enumeration value="matching"/>
	    <xsd:enumeration value="performance"/>
	    <xsd:enumeration value="sequencing"/>
	    <xsd:enumeration value="likert"/>
	    <xsd:enumeration value="numeric"/>
	    <xsd:enumeration value="other"/>
	  </xsd:restriction>
	</xsd:simpleType>

simpleType **resultTypeEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string

source	<xsd:simpleType name="resultTypeEnum" final="restriction">
	  <xsd:restriction base="xsd:string">
	    <xsd:enumeration value="Correct"/>
	    <xsd:enumeration value="choice"/>
	    <xsd:enumeration value="Incorrect"/>
	    <xsd:enumeration value="Unanticipated"/>
	    <xsd:enumeration value="Neutral"/>
	  </xsd:restriction>
	</xsd:simpleType>

simpleType **audioCaptioningEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	integer
source	<xsd:simpleType name="audioCaptioningEnum" final="restriction">

	<xsd:restriction base="xsd:integer"> <xsd:enumeration value="-1"/> <xsd:enumeration value="0"/> <xsd:enumeration value="1"/> </xsd:restriction> </xsd:simpleType>

simpleType **modeEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string
source	<xsd:simpleType name="modeEnum" final="restriction"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Browse"/> <xsd:enumeration value="Normal"/> <xsd:enumeration value="Review"/> </xsd:restriction> </xsd:simpleType>

simpleType **successStatusEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string
source	<xsd:simpleType name="successStatusEnum" final="restriction"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Passed"/> <xsd:enumeration value="Failed"/> <xsd:enumeration value="Unknown"/> </xsd:restriction> </xsd:simpleType>

simpleType **timeLimitActionEnum**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/SCORM
type	string
source	<xsd:simpleType name="timeLimitActionEnum" final="restriction"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Exit"/> <xsd:enumeration value="Continue"/> </xsd:restriction> </xsd:simpleType>

SCHEMA OPENSOCIAL.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial**

Elements

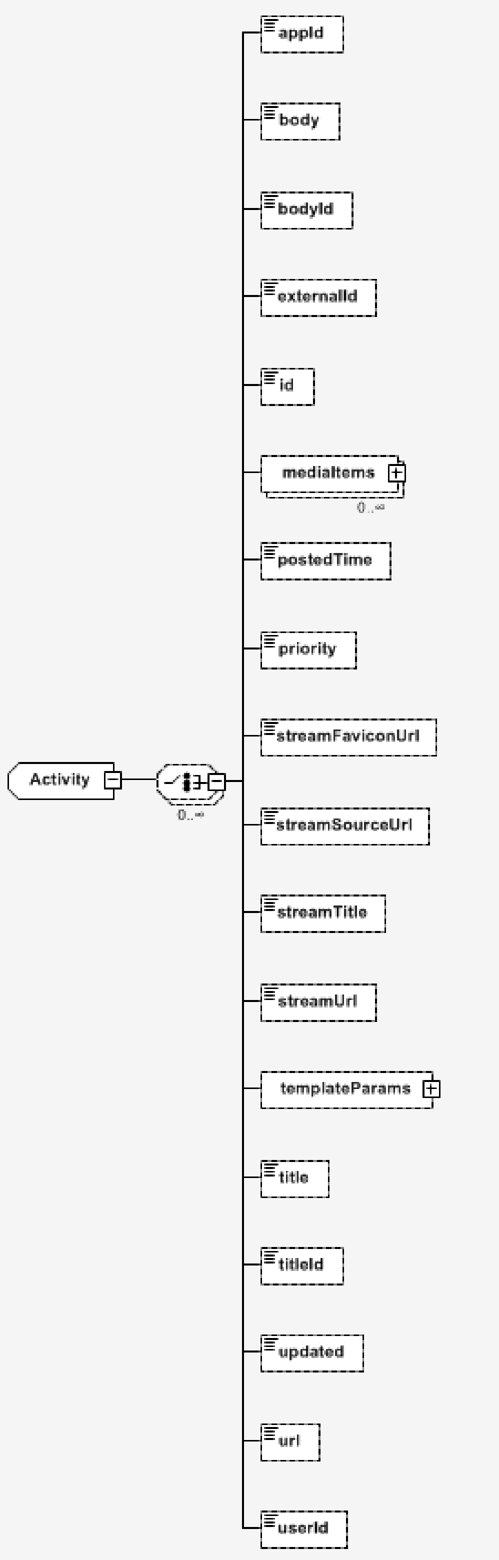
Complex Types

Activity
ActivityTemplateParams
Person
Group
AppdataEntry
Appdata
BodyType
Address
Account
Organization
Name
Url
MediaItem
Drinker
Presence
Smoker
LookingFor
NetworkPresence
PluralPersonField

Simple Types

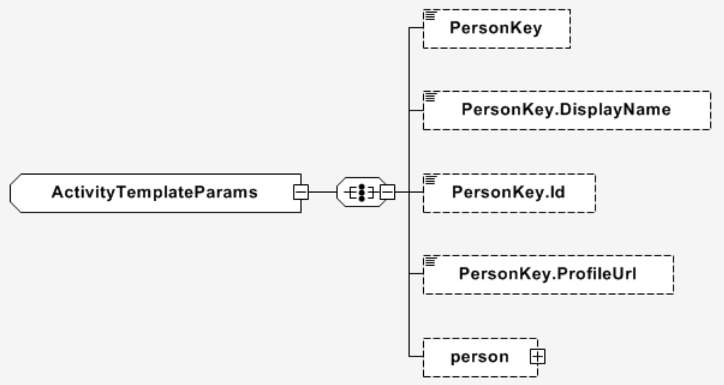
DrinkerType
PresenceType
LookingForType
SmokerType
NetworkPresenceType
MediaItemType

complexType **Activity**

diagram	 Activity 0..∞ appId body bodyId externalId id mediaItems 0..∞ postedTime priority streamFaviconUrl streamSourceUrl streamTitle streamUrl templateParams title titleId updated url userId
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
source	<xs:complexType name="Activity">

	`<xs:choice minOccurs="0" maxOccurs="unbounded">`
	`<xs:element minOccurs="0" name="appId" type="xs:string"/>`
	`<xs:element minOccurs="0" name="body" type="xs:string"/>`
	`<xs:element minOccurs="0" name="bodyId" type="xs:string"/>`
	`<xs:element minOccurs="0" name="externalId" type="xs:string"/>`
	`<xs:element minOccurs="0" name="id" type="xs:string"/>`
	`<xs:element minOccurs="0" maxOccurs="unbounded" name="mediaItems" type="tns:MediaItem"/>`
	`<xs:element minOccurs="0" name="postedTime" type="xs:long"/>`
	`<xs:element minOccurs="0" name="priority" type="xs:double"/>`
	`<xs:element minOccurs="0" name="streamFaviconUrl" type="xs:string"/>`
	`<xs:element minOccurs="0" name="streamSourceUrl" type="xs:string"/>`
	`<xs:element minOccurs="0" name="streamTitle" type="xs:string"/>`
	`<xs:element minOccurs="0" name="streamUrl" type="xs:string"/>`
	`<xs:element minOccurs="0" name="templateParams" type="tns:ActivityTemplateParams"/>`
	`<xs:element minOccurs="0" name="title" type="xs:string"/>`
	`<xs:element minOccurs="0" name="titleId" type="xs:string"/>`
	`<xs:element minOccurs="0" name="updated" type="xs:dateTime"/>`
	`<xs:element minOccurs="0" name="url" type="xs:string"/>`
	`<xs:element minOccurs="0" name="userId" type="xs:string"/>`
	`</xs:choice>`
	`</xs:complexType>`

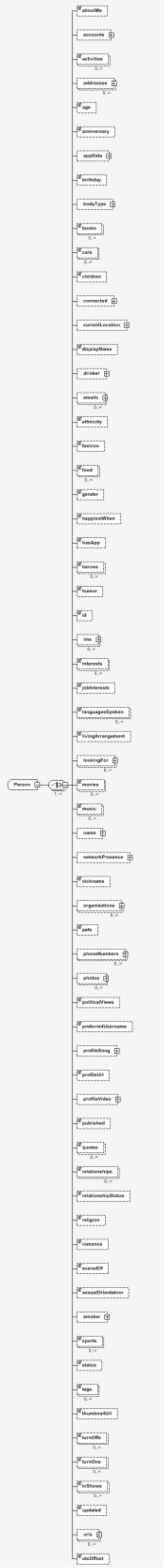
complexType **ActivityTemplateParams**

diagram	 ActivityTemplateParams PersonKey PersonKey.DisplayName PersonKey.Id PersonKey.ProfileUrl person
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

source	`<xs:complexType name="ActivityTemplateParams">`
	`<xs:all>`
	`<xs:element minOccurs="0" name="PersonKey" type="xs:string"/>`
	`<xs:element minOccurs="0" name="PersonKey.DisplayName" type="xs:string"/>`
	`<xs:element minOccurs="0" name="PersonKey.Id" type="xs:string"/>`
	`<xs:element minOccurs="0" name="PersonKey.ProfileUrl" type="xs:string"/>`
	`<xs:element minOccurs="0" name="person" type="tns:Person"/>`
	`</xs:all>`
	`</xs:complexType>`

complexType **Person**

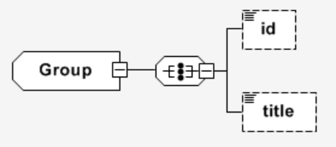
diagram



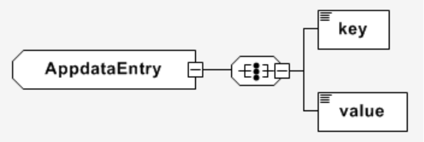
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
source	<xs:complexType name="Person"> <xs:choice minOccurs="1" maxOccurs="unbounded"> <xs:element minOccurs="0" name="aboutMe" type="xs:string"/> <xs:element minOccurs="0" name="accounts" type="tns:Account"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="activities" type="xs:string"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="addresses" type="tns:Address"/> <xs:element minOccurs="0" name="age" type="xs:string"/> <xs:element minOccurs="0" name="anniversary" type="xs:dateTime"/> <xs:element minOccurs="0" name="appData" type="tns:Appdata"/> <xs:element minOccurs="0" name="birthday" type="xs:dateTime"/> <xs:element minOccurs="0" name="bodyType" type="tns:BodyType"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="books" type="xs:string"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="cars" type="xs:string"/> <xs:element minOccurs="0" name="children" type="xs:string"/> <xs:element minOccurs="0" name="connected" type="tns:Presence"/> <xs:element minOccurs="0" name="currentLocation" type="tns:Address"/> <xs:element minOccurs="0" name="displayName" type="xs:string"/> <xs:element minOccurs="0" name="drinker" type="tns:Drinker"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="emails" type="tns:PluralPersonField"/> <xs:element minOccurs="0" name="ethnicity" type="xs:string"/> <xs:element minOccurs="0" name="fashion" type="xs:string"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="food" type="xs:string"/> <xs:element minOccurs="0" name="gender" type="xs:string"/> <xs:element minOccurs="0" name="happiestWhen" type="xs:string"/> <xs:element minOccurs="0" name="hasApp" type="xs:boolean"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="heroes" type="xs:string"/> <xs:element minOccurs="0" name="humor" type="xs:string"/> <xs:element minOccurs="0" name="id" type="xs:string"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="ims" type="tns:PluralPersonField"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="interests" type="xs:string"/> <xs:element minOccurs="0" name="jobInterests" type="xs:string"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="languagesSpoken" type="xs:string"/> <xs:element minOccurs="0" name="livingArrangement" type="xs:string"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="lookingFor" type="tns:LookingFor"/> <xs:element minOccurs="0" maxOccurs="unbounded" name="movies" type="xs:string"/>

```
    <xs:element minOccurs="0" maxOccurs="unbounded" name="music" type="xs:string"/>
    <xs:element minOccurs="0" name="name" type="tns:Name"/>
    <xs:element minOccurs="0" name="networkPresence" type="tns:NetworkPresence"/>
    <xs:element minOccurs="0" name="nickname" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="organizations" type="tns:Organization"/>
    <xs:element minOccurs="0" name="pets" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="phoneNumbers" type="tns:PluralPersonField"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="photos" type="tns:PluralPersonField"/>
    <xs:element minOccurs="0" name="politicalViews" type="xs:string"/>
    <xs:element minOccurs="0" name="preferredUsername" type="xs:string"/>
    <xs:element minOccurs="0" name="profileSong" type="tns:Url"/>
    <xs:element minOccurs="0" name="profileUrl" type="xs:string"/>
    <xs:element minOccurs="0" name="profileVideo" type="tns:Url"/>
    <xs:element minOccurs="0" name="published" type="xs:dateTime"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="quotes" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="relationships" type="xs:string"/>
    <xs:element minOccurs="0" name="relationshipStatus" type="xs:string"/>
    <xs:element minOccurs="0" name="religion" type="xs:string"/>
    <xs:element minOccurs="0" name="romance" type="xs:string"/>
    <xs:element minOccurs="0" name="scaredOf" type="xs:string"/>
    <xs:element minOccurs="0" name="sexualOrientation" type="xs:string"/>
    <xs:element minOccurs="0" name="smoker" type="tns:Smoker"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="sports" type="xs:string"/>
    <xs:element minOccurs="0" name="status" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="tags" type="xs:string"/>
    <xs:element minOccurs="0" name="thumbnailUrl" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="turnOffs" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="turnOns" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="tvShows" type="xs:string"/>
    <xs:element minOccurs="0" name="updated" type="xs:dateTime"/>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="urls" type="tns:Url"/>
    <xs:element minOccurs="0" name="utcOffset" type="xs:int"/>
  </xs:choice>
</xs:complexType>
```

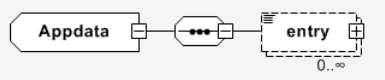
complexType **Group**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
source	<xs:complexType name="Group"> <xs:all> <xs:element minOccurs="0" name="id" type="xs:string"/> <xs:element minOccurs="0" name="title" type="xs:string"/> </xs:all> </xs:complexType>

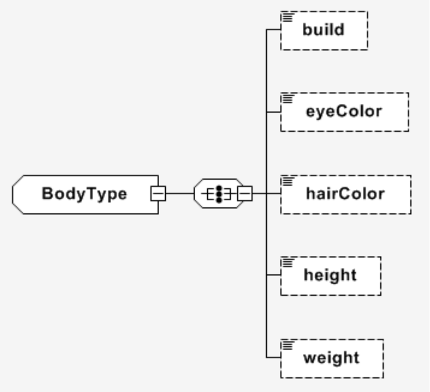
complexType **AppdataEntry**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
source	<xs:complexType name="AppdataEntry" mixed="true"> <xs:all> <xs:element minOccurs="1" name="key" type="xs:string"/> <xs:element minOccurs="1" name="value" type="xs:anyType"/> </xs:all> </xs:complexType>

complexType **Appdata**

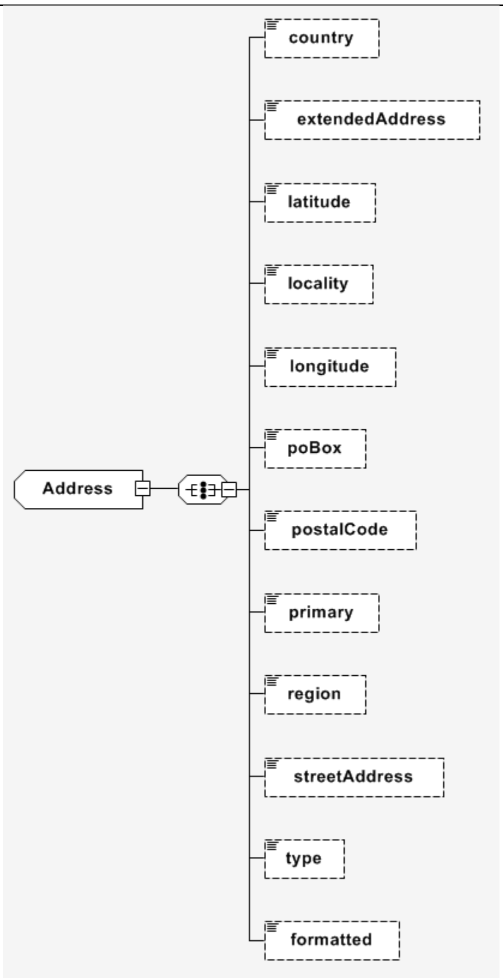
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
source	<xs:complexType name="Appdata"> <xs:sequence> <xs:element minOccurs="0" maxOccurs="unbounded" name="entry" type="tns:AppdataEntry"/> </xs:sequence> </xs:complexType>

complexType **BodyType**

diagram	 BodyType build eyeColor hairColor height weight
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

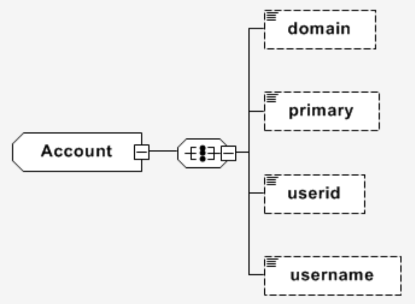
source	`<xs:complexType name="BodyType">`
	`  <xs:all>`
	`    <xs:element minOccurs="0" name="build" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="eyeColor" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="hairColor" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="height" type="xs:double"/>`
	`    <xs:element minOccurs="0" name="weight" type="xs:double"/>`
	`  </xs:all>`
	`</xs:complexType>`

complexType **Address**

diagram	 Address country extendedAddress latitude locality longitude poBox postalCode primary region streetAddress type formatted
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

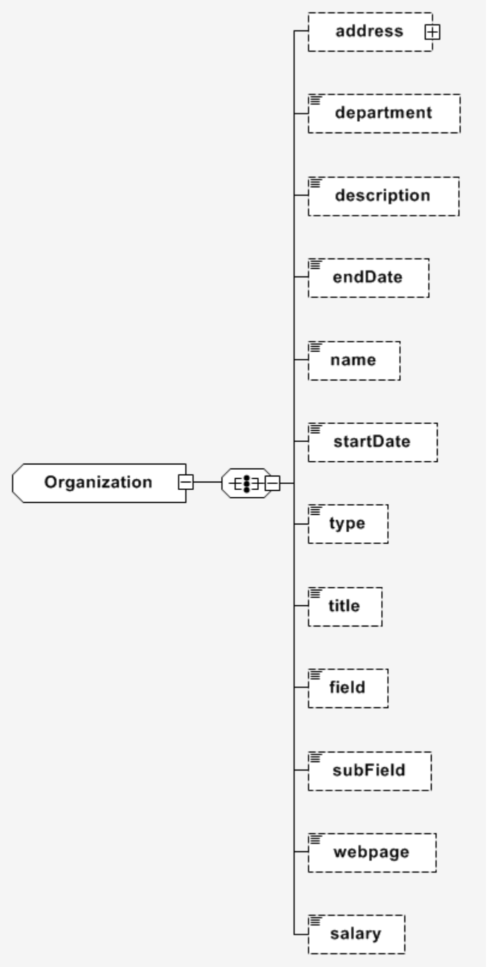
source	<xs:complexType name="Address">
	  <xs:all>
	    <xs:element minOccurs="0" name="country" type="xs:string"/>
	    <xs:element minOccurs="0" name="extendedAddress" type="xs:string"/>
	    <xs:element minOccurs="0" name="latitude" type="xs:double"/>
	    <xs:element minOccurs="0" name="locality" type="xs:string"/>
	    <xs:element minOccurs="0" name="longitude" type="xs:double"/>
	    <xs:element minOccurs="0" name="poBox" type="xs:string"/>
	    <xs:element minOccurs="0" name="postalCode" type="xs:string"/>
	    <xs:element minOccurs="0" name="primary" type="xs:boolean"/>
	    <xs:element minOccurs="0" name="region" type="xs:string"/>
	    <xs:element minOccurs="0" name="streetAddress" type="xs:string"/>
	    <xs:element minOccurs="0" name="type" type="xs:string"/>
	    <xs:element minOccurs="0" name="formatted" type="xs:string"/>
	  </xs:all>
	</xs:complexType>

complexType **Account**

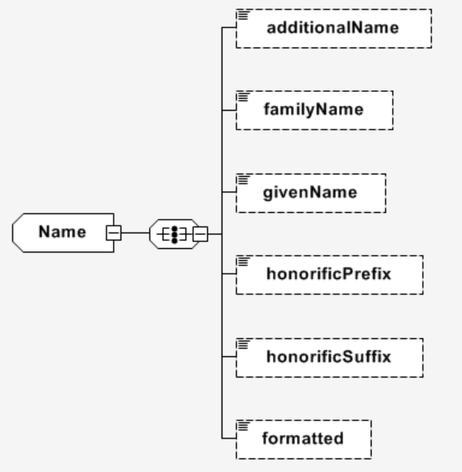
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

source	<xs:complexType name="Account">
	  <xs:all>
	    <xs:element minOccurs="0" name="domain" type="xs:string"/>
	    <xs:element minOccurs="0" name="primary" type="xs:boolean"/>
	    <xs:element minOccurs="0" name="userid" type="xs:string"/>
	    <xs:element minOccurs="0" name="username" type="xs:string"/>
	  </xs:all>
	</xs:complexType>

complexType **Organization**

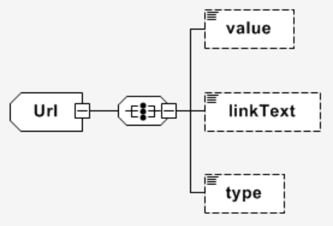
diagram	 Organization address department description endDate name startDate type title field subField webpage salary
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
source	<xs:complexType name="Organization"> <xs:all> <xs:element minOccurs="0" name="address" type="tns:Address"/> <xs:element minOccurs="0" name="department" type="xs:string"/> <xs:element minOccurs="0" name="description" type="xs:string"/> <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/> <xs:element minOccurs="0" name="name" type="xs:string"/> <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/> <xs:element minOccurs="0" name="type" type="xs:string"/> <xs:element minOccurs="0" name="title" type="xs:string"/> <xs:element minOccurs="0" name="field" type="xs:string"/> <xs:element minOccurs="0" name="subField" type="xs:string"/> <xs:element minOccurs="0" name="webpage" type="xs:string"/> <xs:element minOccurs="0" name="salary" type="xs:string"/> </xs:all> </xs:complexType>

complexType **Name**

diagram	 Name additionalName familyName givenName honorificPrefix honorificSuffix formatted
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

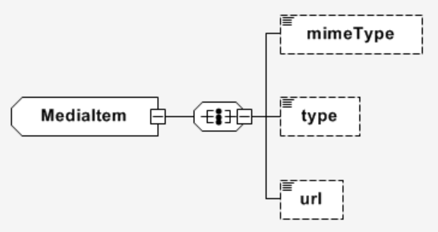
source	`<xs:complexType name="Name">`
	`  <xs:all>`
	`    <xs:element minOccurs="0" name="additionalName" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="familyName" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="givenName" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="honorificPrefix" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="honorificSuffix" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="formatted" type="xs:string"/>`
	`  </xs:all>`
	`</xs:complexType>`

complexType **Url**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

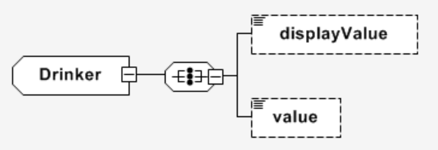
source	`<xs:complexType name="Url">`
	`  <xs:all>`
	`    <xs:element minOccurs="0" name="value" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="linkText" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="type" type="xs:string"/>`
	`  </xs:all>`
	`</xs:complexType>`

complexType **MediaItem**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

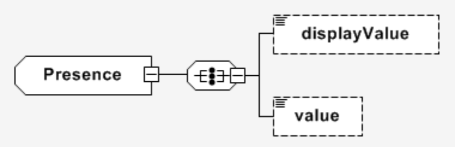
source	`<xs:complexType name="MediaItem">`
	`  <xs:all>`
	`    <xs:element minOccurs="0" name="mimeType" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="type" type="tns:MediaItemType"/>`
	`    <xs:element minOccurs="0" name="url" type="xs:string"/>`
	`  </xs:all>`
	`</xs:complexType>`

complexType **Drinker**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

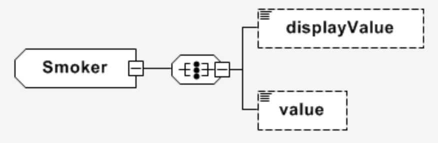
source	`<xs:complexType name="Drinker">`
	`  <xs:all>`
	`    <xs:element minOccurs="0" name="displayValue" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="value" type="tns:DrinkerType"/>`
	`  </xs:all>`
	`</xs:complexType>`

complexType **Presence**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

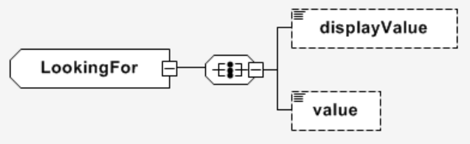
source	`<xs:complexType name="Presence">`
	`  <xs:all>`
	`    <xs:element minOccurs="0" name="displayValue" type="xs:string"/>`
	`    <xs:element minOccurs="0" name="value" type="tns:PresenceType"/>`
	`  </xs:all>`
	`</xs:complexType>`

complexType **Smoker**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

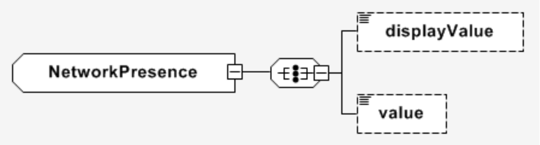
source	<xs:complexType name="Smoker">
	  <xs:all>
	    <xs:element minOccurs="0" name="displayValue" type="xs:string"/>
	    <xs:element minOccurs="0" name="value" type="tns:SmokerType"/>
	  </xs:all>
	</xs:complexType>

complexType **LookingFor**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

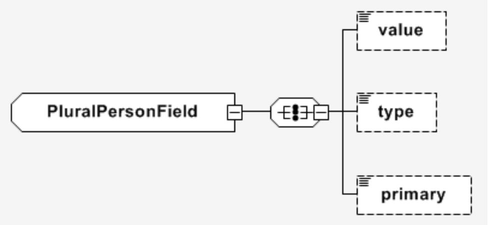
source	<xs:complexType name="LookingFor">
	  <xs:all>
	    <xs:element minOccurs="0" name="displayValue" type="xs:string"/>
	    <xs:element minOccurs="0" name="value" type="tns:LookingForType"/>
	  </xs:all>
	</xs:complexType>

complexType **NetworkPresence**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

source	<xs:complexType name="NetworkPresence">
	  <xs:all>
	    <xs:element minOccurs="0" name="displayValue" type="xs:string"/>
	    <xs:element minOccurs="0" name="value" type="tns:NetworkPresenceType"/>
	  </xs:all>
	</xs:complexType>

complexType **PluralPersonField**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial

source	<xs:complexType name="PluralPersonField">
	  <xs:all>
	    <xs:element minOccurs="0" name="value" type="xs:string"/>
	    <xs:element minOccurs="0" name="type" type="xs:string"/>
	    <xs:element minOccurs="0" name="primary" type="xs:boolean"/>
	  </xs:all>
	</xs:complexType>

simpleType **DrinkerType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
type	string

source	<xs:simpleType name="DrinkerType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="HEAVILY"/>
	    <xs:enumeration value="NO"/>
	    <xs:enumeration value="OCCASIONALLY"/>
	    <xs:enumeration value="QUIT"/>
	    <xs:enumeration value="QUITTING"/>
	    <xs:enumeration value="REGULARLY"/>
	    <xs:enumeration value="SOCIALLY"/>
	    <xs:enumeration value="YES"/>
	  </xs:restriction>
	</xs:simpleType>

simpleType **PresenceType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
type	string

source	<xs:simpleType name="PresenceType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="AWAY"/>
	    <xs:enumeration value="CHAT"/>
	    <xs:enumeration value="DND"/>
	    <xs:enumeration value="OFFLINE"/>
	    <xs:enumeration value="ONLINE"/>
	    <xs:enumeration value="XA"/>
	  </xs:restriction>
	</xs:simpleType>

simpleType **LookingForType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
type	string
source	<xs:simpleType name="LookingForType"> <xs:restriction base="xs:string"> <xs:enumeration value="ACTIVITY_PARTNERS"/> <xs:enumeration value="DATING"/> <xs:enumeration value="FRIENDS"/> <xs:enumeration value="NETWORKING"/> <xs:enumeration value="RANDOM"/> <xs:enumeration value="RELATIONSHIP"/> </xs:restriction> </xs:simpleType>

simpleType **SmokerType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
type	string
source	<xs:simpleType name="SmokerType"> <xs:restriction base="xs:string"> <xs:enumeration value="HEAVILY"/> <xs:enumeration value="NO"/> <xs:enumeration value="OCCASIONALLY"/> <xs:enumeration value="QUIT"/> <xs:enumeration value="QUITTING"/> <xs:enumeration value="REGULARLY"/> <xs:enumeration value="SOCIALLY"/> <xs:enumeration value="YES"/> </xs:restriction> </xs:simpleType>

simpleType **NetworkPresenceType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
type	string
source	<xs:simpleType name="NetworkPresenceType"> <xs:restriction base="xs:string"> <xs:enumeration value="AWAY"/> <xs:enumeration value="CHAT"/> <xs:enumeration value="DND"/> <xs:enumeration value="OFFLINE"/> <xs:enumeration value="ONLINE"/> <xs:enumeration value="XA"/> </xs:restriction> </xs:simpleType>

simpleType **MediaItemType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/OpenSocial
type	string

source	<xs:simpleType name="MediaItemType">
	  <xs:restriction base="xs:string">
	    <xs:enumeration value="AUDIO"/>
	    <xs:enumeration value="IMAGE"/>
	    <xs:enumeration value="VIDEO"/>
	  </xs:restriction>
	</xs:simpleType>

SCHEMA GAMING.XSD

Properties

attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

Elements

- Gamification

Complex Types

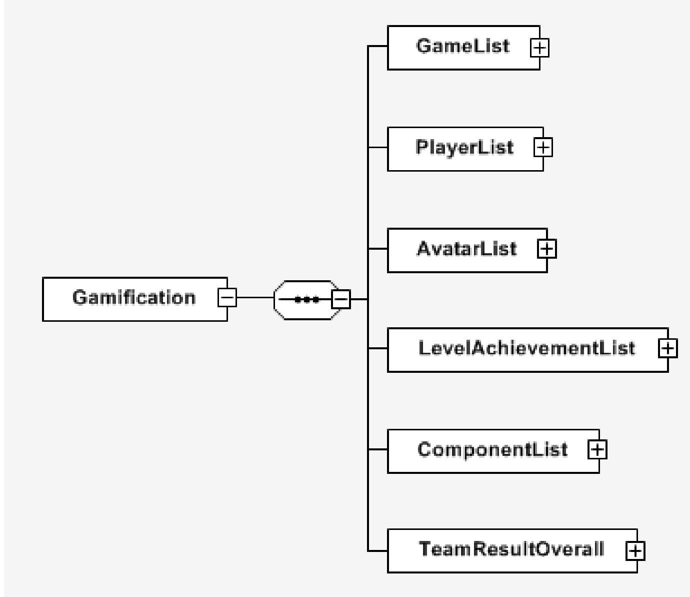
- AchievedBadgeType
- AppType
- AvatarType
- BadgeType
- ComponentType
- CustomizedTaskType
- LevelAchievementType
- PlayerType
- ScoredTaskType
- TaskType
- TeamPlayerType
- TeamType

Simple Types

- EmailAddressType

element **Gamification**

diagram



namespace http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

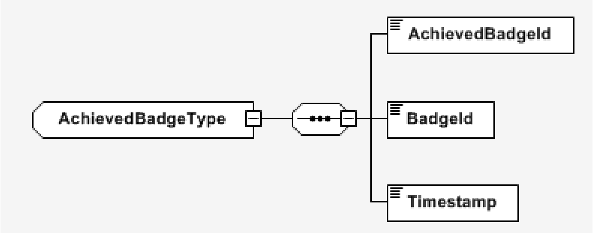
source

```
<xs:element name="Gamification">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="GameList">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Game">
                            <xs:annotation>
                                <xs:documentation>A game for the gamification framework</xs:documentation>
                            </xs:annotation>
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="GameId"/>
                                    <xs:element name="Name"/>
                                    <xs:element name="Description"/>
                                    <xs:element name="ComponentId"/>
                                    <xs:element name="ExpiryDate"/>
                                    <xs:element name="MaximumScoreAchievable" type="xs:int"/>
                                    <xs:element name="CustomizedTaskList">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="CustomizedTask" type="GAMIFICATION:CustomizedTaskType"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
```

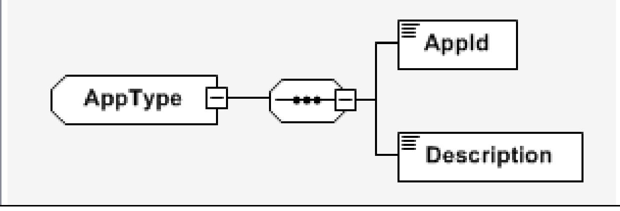
```
                                      <xs:element name="TeamList">
                                                <xs:complexType>
                                                          <xs:sequence>
                                                                    <xs:element      name="Team"
type="GAMIFICATION:TeamType" maxOccurs="unbounded"/>
                                                          </xs:sequence>
                                                </xs:complexType>
                                      </xs:element>
                                      <xs:element name="BadgeList">
                                                <xs:complexType>
                                                          <xs:sequence>
                                                                    <xs:element     name="Badge"
type="GAMIFICATION:BadgeType" maxOccurs="unbounded"/>
                                                          </xs:sequence>
                                                </xs:complexType>
                                      </xs:element>
                                      <xs:element name="Status"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="PlayerList">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Player" type="GAMIFICATION:PlayerType"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="AvatarList">
                <xs:annotation>
                    <xs:documentation>List of global Avatars</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Avatar" type="GAMIFICATION:AvatarType"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="LevelAchievementList">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element                              name="LevelAchievement"
type="GAMIFICATION:LevelAchievementType"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
```

	<xs:element name="ComponentList"> <xs:complexType> <xs:complexContent> <xs:extension base="GAMIFICATION:ComponentListType"> <xs:sequence> <xs:element name="Component" type="GAMIFICATION:ComponentType"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element> <xs:element name="TeamResultOverall"> <xs:complexType> <xs:sequence> <xs:element name="YesterdaysScore"/> <xs:element name="TodaysScore"/> <xs:element name="HighestScore"/> <xs:element name="Timestamp"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element>

complexType **AchievedBadgeType**

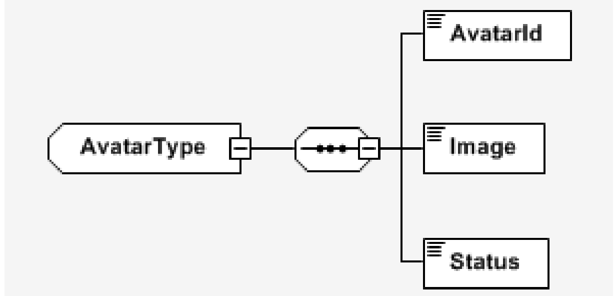
diagram	 AchievedBadgeType AchievedBadgeId BadgeId Timestamp
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming
source	<xs:complexType name="AchievedBadgeType"> <xs:sequence> <xs:element name="AchievedBadgeId"/> <xs:element name="BadgeId"/> <xs:element name="Timestamp"/> </xs:sequence> </xs:complexType>

complexType **AppType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

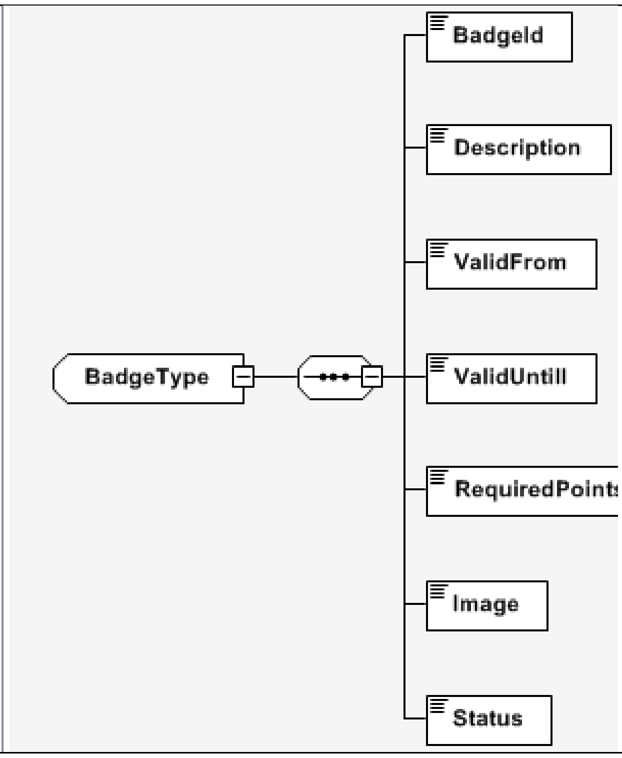
source	<xs:complexType name="AppType">
	    <xs:sequence>
	        <xs:element name="AppId"/>
	        <xs:element name="Description"/>
	    </xs:sequence>
	</xs:complexType>

complexType **AvatarType**

diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

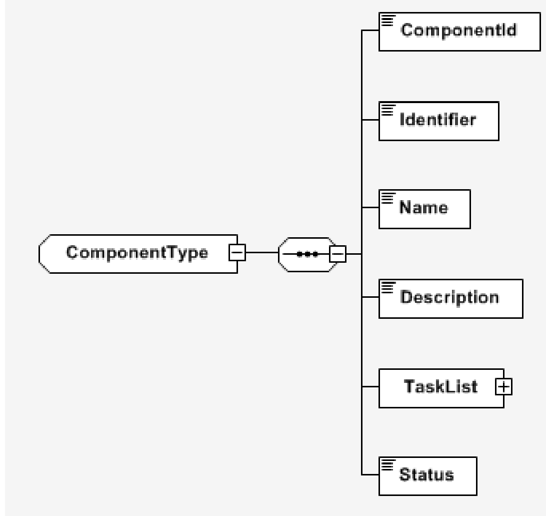
source	<xs:complexType name="AvatarType">
	    <xs:sequence>
	        <xs:element name="AvatarId"/>
	        <xs:element name="Image"/>
	        <xs:element name="Status"/>
	    </xs:sequence>
	</xs:complexType>

complexType **BadgeType**

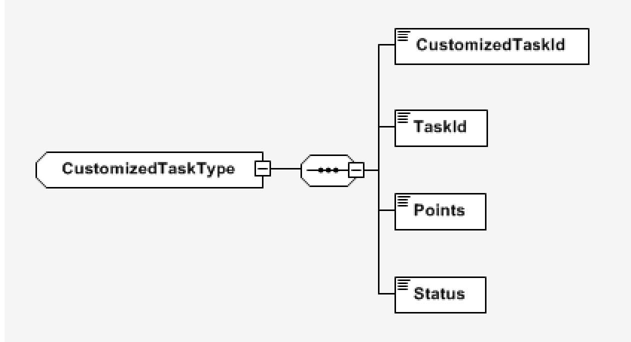
diagram	 BadgeId Description ValidFrom BadgeType ValidUntill RequiredPoint: Image Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

source	<xs:complexType name="BadgeType">
	    <xs:sequence>
	        <xs:element name="BadgeId"/>
	        <xs:element name="Description"/>
	        <xs:element name="ValidFrom"/>
	        <xs:element name="ValidUntill"/>
	        <xs:element name="RequiredPoints"/>
	        <xs:element name="Image"/>
	        <xs:element name="Status" type="xs:boolean" default="false"/>
	    </xs:sequence>
	</xs:complexType>

complexType **ComponentType**

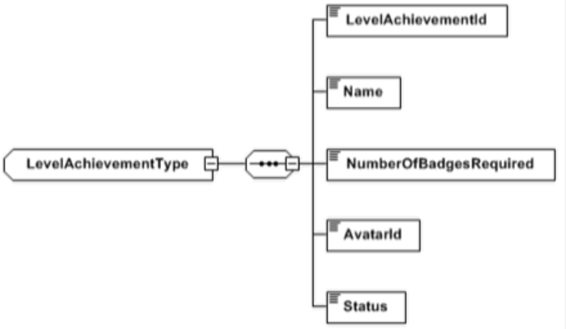
diagram	 ComponentType ComponentId Identifier Name Description TaskList Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming
source	<xs:complexType name="ComponentType"> <xs:sequence> <xs:element name="ComponentId"/> <xs:element name="Identifier"/> <xs:element name="Name"/> <xs:element name="Description"/> <xs:element name="TaskList"> <xs:complexType> <xs:sequence> <xs:element name="Task" type="GAMIFICATION:TaskType"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="Status"/> </xs:sequence> </xs:complexType>

complexType **CustomizedTaskType**

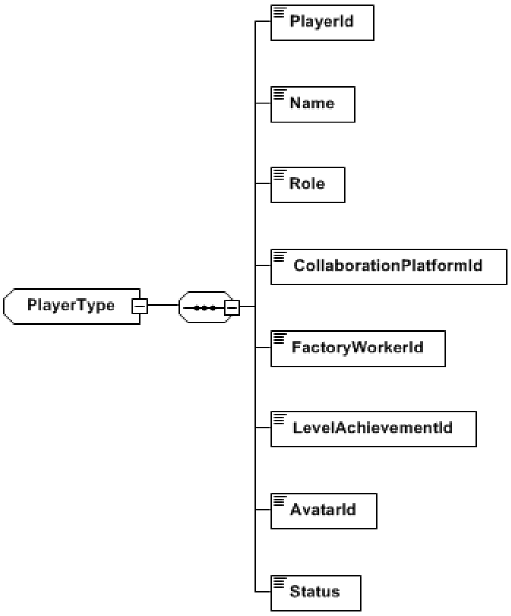
diagram	 CustomizedTaskType CustomizedTaskId TaskId Points Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

source	<xs:complexType name="CustomizedTaskType"> <xs:sequence> <xs:element name="CustomizedTaskId"/> <xs:element name="TaskId"/> <xs:element name="Points"/> <xs:element name="Status"/> </xs:sequence> </xs:complexType>

complexType **LevelAchievementType**

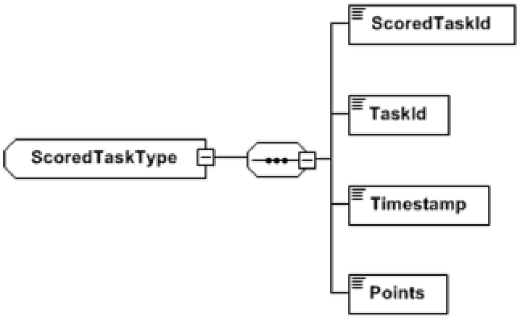
diagram	 LevelAchievementType LevelAchievementId Name NumberOfBadgesRequired AvatarId Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming
source	<xs:complexType name="LevelAchievementType"> <xs:sequence> <xs:element name="LevelAchievementId"/> <xs:element name="Name"/> <xs:element name="NumberOfBadgesRequired"/> <xs:element name="AvatarId"/> <xs:element name="Status"/> </xs:sequence> </xs:complexType>

complexType **PlayerType**

diagram	 PlayerType PlayerId Name Role CollaborationPlatformId FactoryWorkerId LevelAchievementId AvatarId Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

source	`<xs:complexType name="PlayerType">`
	`  <xs:sequence>`
	`    <xs:element name="PlayerId"/>`
	`    <xs:element name="Name"/>`
	`    <xs:element name="Role"/>`
	`    <xs:element name="CollaborationPlatformId"/>`
	`    <xs:element name="FactoryWorkerId"/>`
	`    <xs:element name="LevelAchievementId" type="xs:int"/>`
	`    <xs:element name="AvatarId"/>`
	`    <xs:element name="Status"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

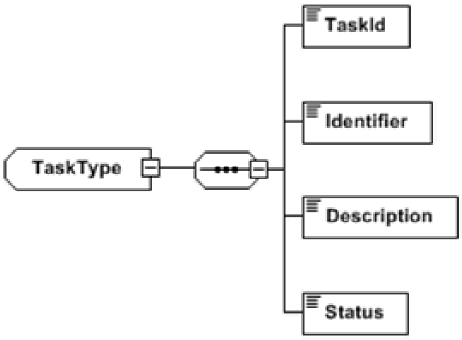
complexType **ScoredTaskType**

diagram	 ScoredTaskType ScoredTaskId TaskId Timestamp Points
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

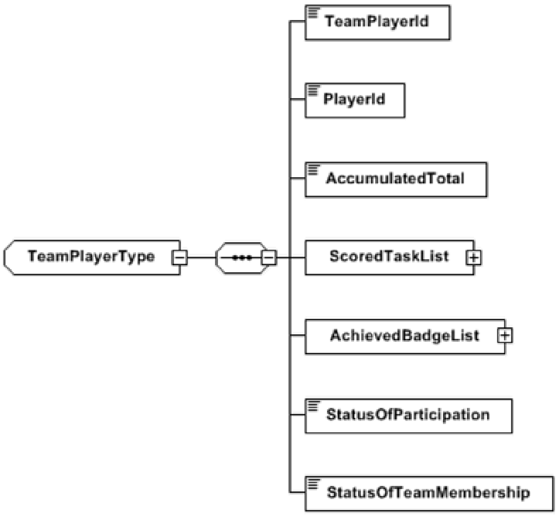
source	`<xs:complexType name="ScoredTaskType">`
	`  <xs:sequence>`

	<xs:element name="ScoredTaskId"/> <xs:element name="TaskId"/> <xs:element name="Timestamp"/> <xs:element name="Points"/> </xs:sequence> </xs:complexType>

complexType **TaskType**

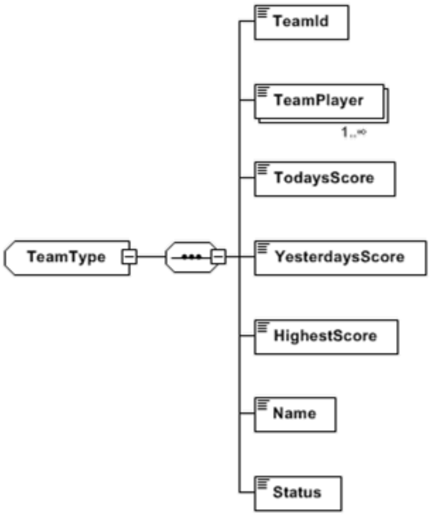
diagram	
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming
source	<xs:complexType name="TaskType"> <xs:sequence> <xs:element name="TaskId"/> <xs:element name="Identifier"/> <xs:element name="Description"/> <xs:element name="Status"/> </xs:sequence> </xs:complexType>

complexType **TeamPlayerType**

diagram	 TeamPlayerType TeamPlayerId PlayerId AccumulatedTotal ScoredTaskList AchievedBadgeList StatusOfParticipation StatusOfTeamMembership
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming

source	<xs:complexType name="TeamPlayerType"> <xs:sequence> <xs:element name="TeamPlayerId"/> <xs:element name="PlayerId" type="xs:string"/> <xs:element name="AccumulatedTotal"/> <xs:element name="ScoredTaskList"> <xs:complexType> <xs:sequence> <xs:element name="ScoredTask" type="GAMIFICATION:ScoredTaskType"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="AchievedBadgeList"> <xs:complexType> <xs:sequence> <xs:element name="AchievedBadge" type="GAMIFICATION:AchievedBadgeType"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="StatusOfParticipation"/> <xs:element name="StatusOfTeamMembership"/> </xs:sequence> </xs:complexType>

complexType **TeamType**

diagram	 TeamId TeamPlayer 1..∞ TodaysScore TeamType YesterdaysScore HighestScore Name Status
namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming
source	<xs:complexType name="TeamType"> <xs:sequence> <xs:element name="TeamId" type="xs:string"/> <xs:element name="TeamPlayer" type="GAMIFICATION:TeamPlayerType"

	maxOccurs="unbounded"/> <xs:element name="TodaysScore"/> <xs:element name="YesterdaysScore"/> <xs:element name="HighestScore"/> <xs:element name="Name"/> <xs:element name="Status"/> </xs:sequence> </xs:complexType>

simpleType **EmailAddressType**

namespace	http://www.satisfactory-project.eu/XMLSchema/v1.0/Gaming
type	string
source	<xs:simpleType name="EmailAddressType"> <xs:restriction base="xs:string"> <xs:pattern value="[^@]+@[^\.]+\..+"/> </xs:restriction> </xs:simpleType>