Office of Government Commerce

Towards e - government

UK Online – Information Architecture – CECA Property Data Structures Fragment

Document Type: Specification

Version: 2.1
Status: Draft
Document Reference: [ ]

07/11/2001

Document Control

Contributors		
Name/Position	**Organisation**	**Contact Details**
Geoff Parkin	Property and Construction Directorate within The Office of Government Commerce	geoff.parkin@ogc.gsi.gov.uk geoff@murphysoft.com
Steve Rooney	Property and Construction Directorate within The Office of Government Commerce	steve.rooney@ogc.gsi.gov.uk
Chris Statham	Property and Construction Directorate within The Office of Government Commerce	chris.statham@ogc.gsi.gov.uk
Sue Hamer-Moss	Property and Construction Directorate within The Office of Government Commerce	sue.hamer-moss@ogc.gsi.gov.uk shamermoss@hedra.co.uk
Gerry Bartlett	Property and Construction Directorate within The Office of Government Commerce	gerry.bartlett@ogc.gsi.gov.uk
Paul Spencer	alphaXML	paul.spencer@alphaxml.com
Steven Benbow	alphaXML	steven.benbow@alphaxml.com
Matthew Hasler	alphaXML	matthew.hasler@alphaxml.com

Current Version				
Date	**Version**	**Status**	**Editor/ Author**	**Comment**
28th February 2002	2.1	RFC	Author: Geoff Parkin	Submission to Govtalk for 3rd RFC cycle following implementation of 2nd phase comments and schema changes required for CECA Bulk update specification.

Authorisation	
Name	
Organisation	
Date	

Change History				
Date	**Version**	**Status**	**Editor/ Author**	**Comment**
28/02/2002	2.1	RFC	Geoff Parkn	Submission to Govtalk for 3rd RFC cycle following implementation of 2nd phase comments and schema changes required for CECA Bulk update specification.
04/11/2001	2.0	Archive	Geoff Parkn	Submission to Govtalk for 2nd RFC cycle following implementation of 1st phase comments
17/10/2001	1.1	Archive	Geoff Parkin	Version 1.1 draft following initial RFC comments (see Appendix A : version control)
25/07/2001	1.0	Archive	Geoff Parkin	Submission to Govtalk for RFC
25/07/2001	0.7	Archive	Geoff Parkin	Minor changes to LeaseStructure schema.

22/06/2001	0.6	Archive	Steve Benbow	Added UML diagrams and related documentation
15/06/2001	0.5	Archive	Geoff Parkin	Final draft prior submission to AlphaXML for inclusion of UML
15/06/2001	0.4	Archive	Geoff Parkin	See Appendix A for list of changes
05/06/2001	0.3	Archive	Geoff Parkin	The messaging schemas have been simplified. Each CECA data object now has only 1 messaging schema to use for both Add New and Amend actions
24/04/2001	0.2	Archive	Geoff Parkin	Amended relational model
20/04/2001	0.1a	Archive	Geoff Parkin	Initial draft

Contents

Figures

1 Requirement Definition

1.1 Source of Requirement

This document defines the data required to be submitted by Central Civil Government departments to the Office of Government Commerce (OGC) about their Civil Estate property occupations.

DAO Gen 1/96[1], which incorporates the Civil Estate Co-ordination Agreement (CECA), placed a requirement on departments to update core data about their occupations. Most departments have had difficulty in delivering this information electronically because of problems with the linkage between their database systems and OGC's database. The information exchange that takes place is achieved largely as a result of local, face to face liaison meetings – often a laborious and time consuming exercise.

In response to a Modernising Government White Paper initiative, OGC have developed an electronic Property Information Mapping System (e-PIMS) of government's civil property occupations, which will accept and deliver bulk data according to the schemae defined in this document.

The Property and Construction Directorate of OGC manages a central database of property details. The CECA agreement specifies a framework for the update and maintenance of the core property data held in the OGC database, based on data held in other departments estate management systems.

An initiative is underway to facilitate the transfer of CORE CECA data between the database managed by OGC and other Estate Management databases managed by individual departments. The main use of the schema will be to support initial bulk data upload and subsequent synchronisation of information.
This initiative is developed in line with the government's own e-Gif standards, but is essentially designed for use within the public sector, to facilitate the improvement in quality and accuracy of Civil Estate data.

XML is being used extensively within the e-Government initiative to facilitate communication, with this in mind CECA is utilising XML as the key data format for bulk data transfer.

The transition from current processes to electronic processes is expected to realise an annual £5 million savings to UK Government by 2006.

1.2 Definition Statement

The aim of this document is to outline a proposed XML schema architecture for representing CECA Property data structures.

The architecture has been developed with modular structure and has 2 clearly defined parts. The actual data structures referred to as the *architecture* and a set of schemas to be used to transfer the data, referred to as *messaging* schemas.

[1] HM Treasury Dear Accounting Officer letter of 1996

2 CECA Data Structures

2.1 Overview of CECA

The CECA agreement was issued under the Dear Accounting Officer letter DAO GEN 1/96 and set out the agreed procedures to be followed by OGC (formerly PACE) and departments in co-ordinating activity in the property market and in co-ordinating rationalisations of Civil Estate property. It also sets out the obligations placed on departments to provide OGC with the information necessary to successfully undertake these co-ordination functions, but responsibility for property decisions moved to the owning departments. OGC, in addition to the co-ordination and estate rationalisation functions, also held and maintained all Core Estate data on a central database.

The Civil Estate is estate owned or leased by a government department (including non-Ministerial departments; and Crown NDBP's). Property is usually held in the name of the Secretary of State for the Environment, but may be held in the name of another Government Minister, or in the name of a Government Agency.

The Agreement extends to all Civil estate property in Great Britain with the exception of the NHS Estate, the Prison Estate, the Foreign Office Overseas Estate and the Defence Estate.

OGC and Departments share a common interest in the timely exchange of key information. The CECA agreement stipulates that departments will provide OGC with updated information relating to the update and maintenance of Civil Estate property and occupation data.

This XML initiative is directed toward the support of the Core element of the CECA agreement and the implementation of linkages between government departments for the sharing of Civil Estate data. Core data is defined as specific information about individual properties (eg address, number of floors etc) which departments are required to provide to OGC.

The schemas detailed in this document relate to the Core element of the CECA agreement. Core data is defined as specific information about individual properties (eg address, number of floors etc) which departments are required to provide to OGC.

2.2 Core CECA structures defined

The Core CECA data structures are comprised of 9 entities which represent a Civil Estate occupancy. These are listed below:

Property - A separately identifiable unit of real estate, comprising an area of land with or without building, within a continuous boundary. This may range from a site comprising land and any number of buildings to a single building or even (under exceptional circumstances) part of a building.

Building - An identifier and basic details within the property of each building. A holding may fall entirely within the building or it may consist of buildings on a multi-building site. Against each Holding are recorded the numbers of all buildings within that holding.

Holding - The basic unit associated with a legal interest in a property. A property may comprise a single or many Holdings.

Occupation - Details of the occupation of a holding or a sub-let / minor occupation of any kind.

Lease - Leases come in 2 forms

Main Lease - Details of a lease or other legal transaction for a Holding
Occupation based lease - Details of a lease or other legal transaction for a sublet minor occupation of any kind. These may be either commercial (Sublets) or internal (MOT's)

Landlord – An identifier, description and contact details of a landlord. A landlord is associated with a lease.

Tenant - An identifier, description and contact details of a tenant. A tenant is associated with an occupation.

Vacant Space – Details relating to vacant areas of Civil Estate property associated with holdings or occupations.

BreakReview – Details relating to breaks or reviews in the conditions of a lease. BreakReviews are associated with leases.

The data entities listed above are relational. These relationships combine to give a complete view of a Civil Estate Property entity (to be referred to as a CECA entity).

The figure below (Figure 1) represents the relational architecture used to generate the XML schemas detailed in section 4.

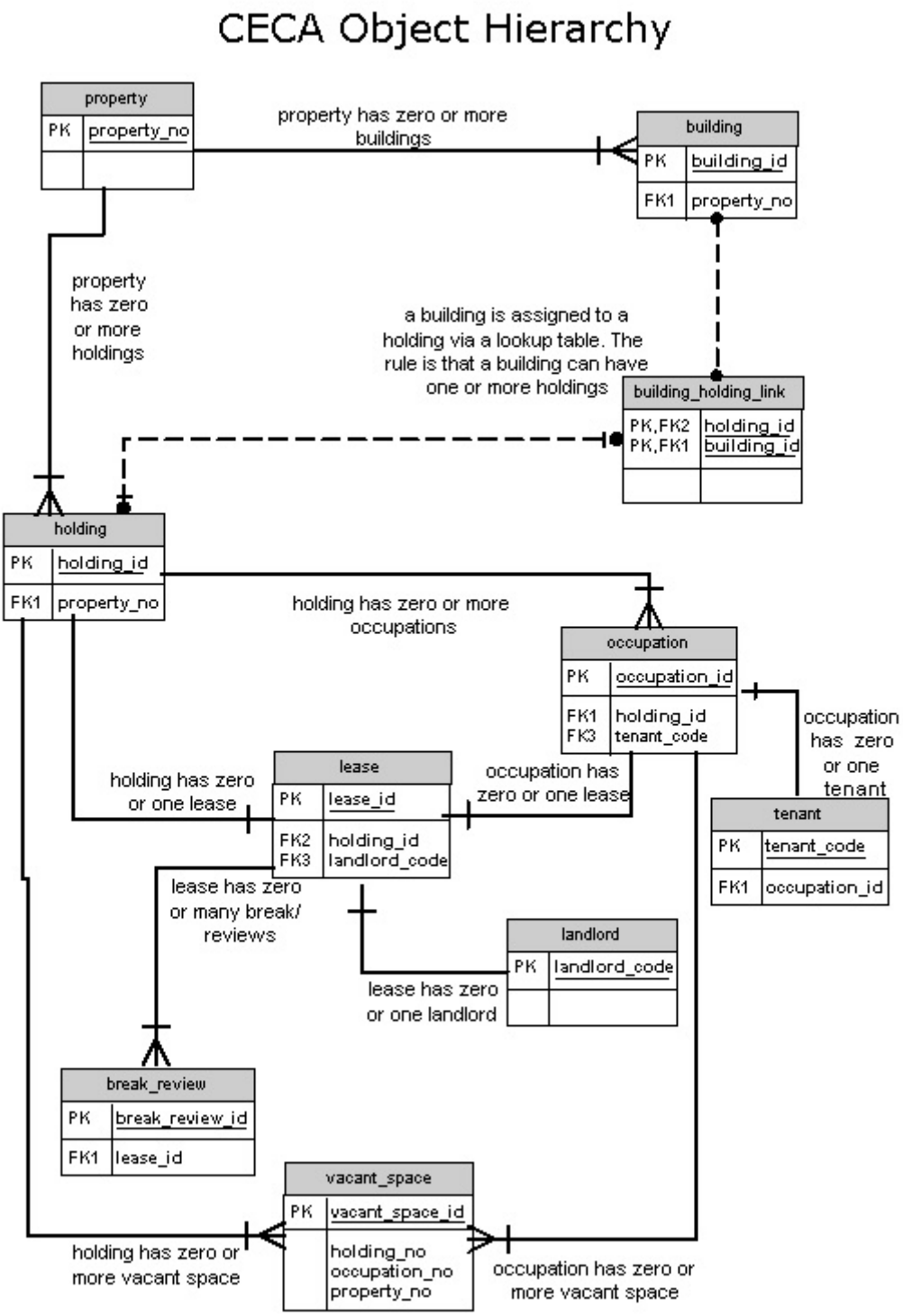
The following rules apply regarding the relationships between the data entities.

- A property entity is a single element representing key data relating to the entire CECA entity. This data includes the address, total site area, OS reference details etc.
- A property can have zero or many buildings.
- A property can have zero or many holdings.
- A building can have one or many holdings. There is a many to many relationship between holding and building which is managed by an intermediary data structure (see relationship diagram below). Actual implementation will give focus to building as the main element in the relationship. So the statement a building can have

many holdings is true (although a holding can be part of multiple building relationships)

- A holding is unique to a specific government department and is identified as such.
- A holding can optionally have an associated lease known as the main lease.
- A holding can have zero or many occupations.
- A holding can have zero or many vacant space records.
- An occupation is unique to a specific government department and is identified as such,
- An occupation can optionally have an associated lease known as the sub lease.
- An occupation can have zero or many vacant space records.
- An occupation can optionally be associated with a tenant.
- A lease can have zero or many break/review records.
- A lease can optionally be associated with a landlord.

Figure 1: CECA data structure relational schema



2.3 Data catalogue

Please refer to the CECA agreement for the data structures.

The contents of this agreement have been submitted to the Government process group for consideration for inclusion in the Government Data Standards catalogue (GDS).

3 UML notation for XML Schema

The job of modelling XML Schema in UML is at an early stage, and alternative approaches are emerging. The approach taken here is based on an amalgamation of the methods suggested in [7,8], with slight extensions to identify types and namespaces in a clearer fashion. The interested reader is referred to the original documents, in particular [7], for background reading. Extensions of the UML notation of [7,8] specific to this document are described in the following section.

3.1 Extensions of the existing UML notations

To motivate a description of the UML modelling approach that has been followed, consider the following small XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.alphaxml.com/stringSchema"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
targetNamespace="http://www.alphaxml.com/stringSchema"
elementFormDefault="qualified">
  <xsd:simpleType name="myStringType">
     <xsd:restriction base="xsd:string">
       <xsd:maxLength value="100"/>
       <xsd:whiteSpace value="preserve"/>
     </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="myBagOfStringTypes">
     <xsd:sequence>
       <xsd:element name="myString" type="myStringType" minOccurs="1"
       maxOccurs="unbounded"/>
     </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Figure 2: the dummyString schema

Following the approaches in [7,8], the dummyString schema would be represented as in Figure 3 and Figure 4 respectively. Clearly, the notation of [7] is far more verbose than that of [8], but conveys a lot more information. In particular it is clear from [7], which objects are elements, simpleTypes and complexTypes. This information is of potential use to conforming XML message creators, and is hence retained in this modelling approach. The concise approach of [8] has led to a slight simplification of the notation of [7], we allow the type of an element to be contained in its attribute list (as is also the case in XML Schema). This eliminates the need for the central object in Figure 3.

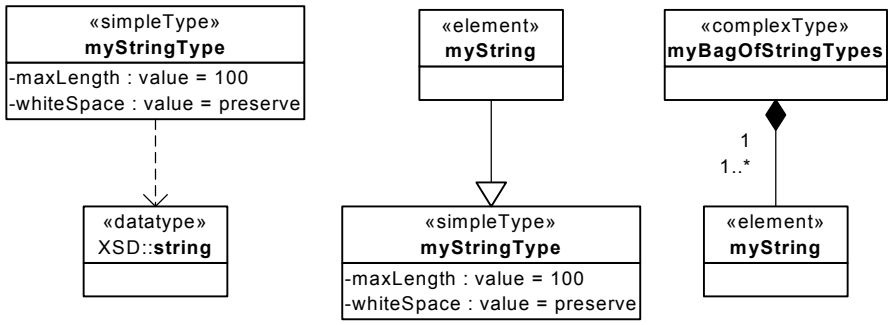


Figure 3: representation of the dummyString schema following [7].

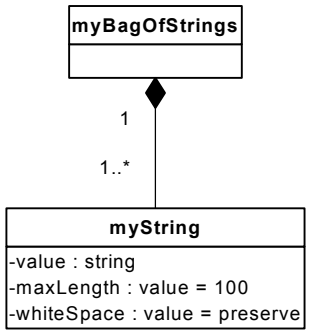


Figure 4: representation of the dummyString schema following [8].

A further diagrammatical enhancement has been made in an attempt to clearly indicate which types and elements defined in the XML Schema are global types. A bold box surrounding the object will distinguish such types and elements. Thus the representation of the dummyString schema in our UML notation would appear as in Figure 5. Furthermore, we insist that all simpleType and complexType objects are defined in a standalone UML diagram. The fact that all types are defined in standalone UML diagrams, and that the types of all objects are clearly indicated is useful documentation for the developers of XML messages to be validated against the schema, and for developers of a second XML Schema that uses the namespace of this schema. Each global simpleType is defined once; hence the developer can clearly see what making use of a specific type implies.

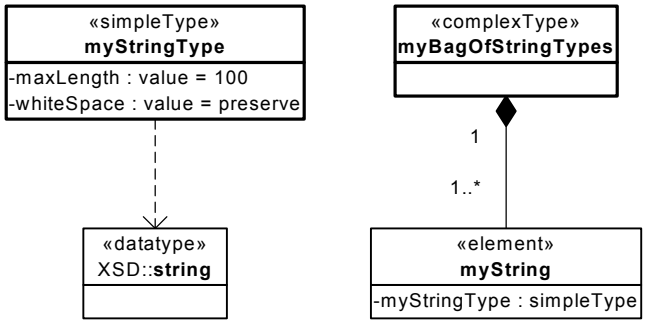


Figure 5: approach taken to modelling the dummyString schema in this document.

Following [7], namespace diagrams are included in the overall model of the XML Schema. However, we diverge slightly from the approach in [7] by also allowing included schema (through `<xsd:include schemaLocation="…"/>`) in the namespace diagram. Included schemas are highlighted, and in subsequent UML diagrams objects from the included schema are highlighted in the appropriate colour. The convention adopted has been to indicate inclusion of a schema with the UML generalisation notation, and declaration of a namespace with the UML dependency notation (with the exception of the target namespace). This convention is adopted throughout further UML diagrams of the schema, so that any objects from a declared namespace (other than the target namespace) use dependency notation, whereas objects from an included schema use generalisation notation (and are highlighted). See for example Figure 6 and Figure 7.

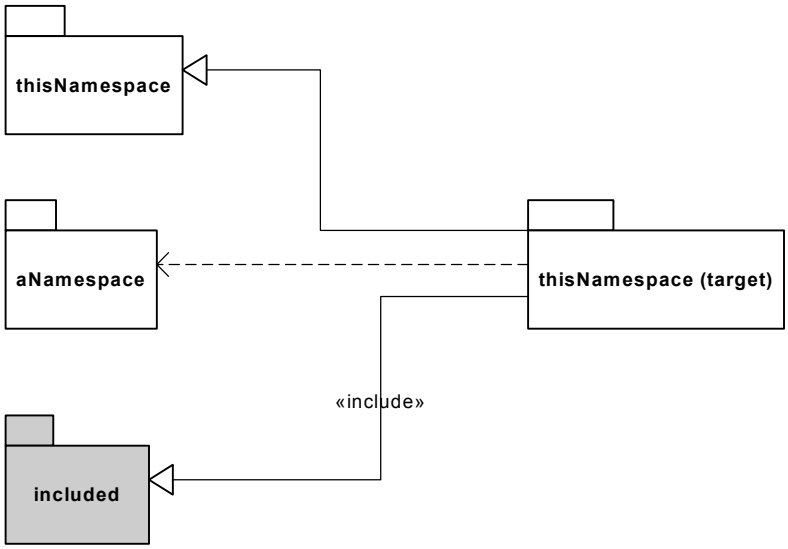


Figure 6: namespace diagram. The diagram has been extended to show included schema. Declared namespaces are highlighted.

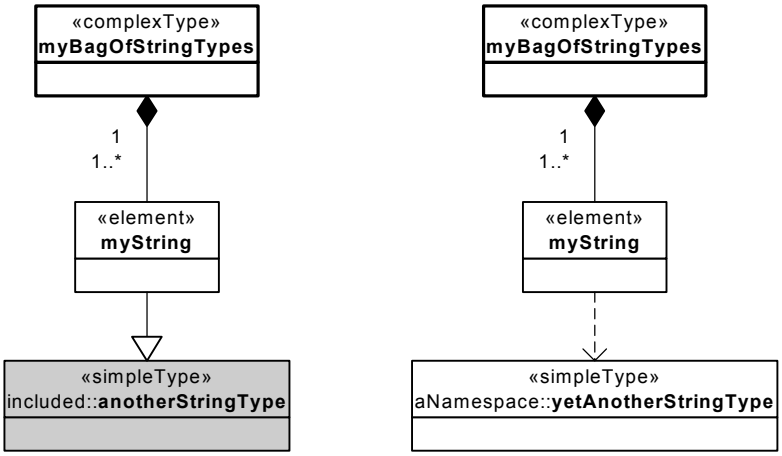


Figure 7: approach taken for referencing types defined in included schema (left) and types from declared namespaces (right).

3.2 Other types of XML Schema elements

We generally follow the modelling approach outlined in [7] to model XML Schema element that have not already been explicitly mentioned, for example enumerations, groups, choices, etc.

4 XML Script

4.1 Proposed schema architecture

The architecture has been developed with an object based structure and has 2 clearly defined parts. The actual data structures, referred to as the *architecture* and a set of schemas to be used to transfer the data, referred to as *messaging* schemas.

The architecture schemas provide the structure to reflect the pre-defined CECA data schema and provide a set of specific types used by other architecture and the messaging schemas. The architecture schemas implement the CECA object model structure at the specific position of the data type within the model. This involves the defining of the individual data elements of the data type and providing a structured reference to "child" elements (eg PropertyStructure defines each property element and also the ability to specify a number of Building and Holding elements).
NOTE: the architecture schemas are NOT to be used by XML document "authors". They are specified to provide structure and validation rules to the messaging schemas only.

The architecture schemas are described briefly below:

- CECADataTypes – declares common simple and complex Types used by the other schemas
- CECAAddressStructure – defines the structure(s) to be used to specify a property address.
- PropertyStructure – one of 5 architecture schemas which represent the 5 main elements of the CECA data schema. This represents the structure of an element of property data and defines by reference the relationships in the CECA object hierarchy with building and holding.
- BuildingStructure – represents an element of Building data and defines the building/holding link relationships defined in the CECA object model.
- HoldingStructure – represents an element of Holding data and defines by reference the relationships in the CECA object hierarchy with occupation, the main lease and vacant space.
- LeaseStructure – represents an element of Lease data and defines by reference the relationships in the CECA object hierarchy with Landlord and break/review.
- OccupationStructure – represents an element of Occupation data defines by reference the relationships in the CECA object hierarchy with tenant, lease and vacant space.
- LandlordStructure – represents an element of Landlord data.
- TenantStructure – represents an element of Tenant data.

In addition other CECA objects have been defined as complexType structures within existing schemas. These are defined as follows:

- VacantSpaceType – defined within CECADataTypes and used by HoldingStructure and OccupationStructure to define a number of vacant space data elements associated with the specific instance.
- BreakReviewType – this is defined within LeaseStructure and used to define a number of break/review data elements associated with the actual lease.

The messaging schemas are to be used by XML document "authors" who want to represent their property data in the CECA format. These are listed briefly below:

- CECAEnvelope – This is defined as the main messaging schema used by CECA XML document authors to transmit data. The CECAEnvelope specifies "document signing" details and provides the structure to detail one or more "messages". These messages are XML documents based on the other messages defined below.

The following 5 messages define each individual main CECA data type as defined in the CECA object hierarchy. These have been defined as messages to provide flexibility to this specification, to enable further widened use outside the scope of the CECAEnvelope which has been defined for data transmission. These schemas allow messages to be generated for data at any part of the CECA object hierarchy.

- CECAProperty – this represents the structure for sending updates to a main property data record. In addition this can be used to send a complex XML document containing an entire property entities data. This includes the main property details, all associated buildings, holdings, leases and occupations. This schema will be used when the requirement is to transfer entire property related data (eg during bulk update procedures between separate government departments).
- CECABuilding - this represents the structure for sending updates to a main building data record and optionally any building/holing link details.
- CECAHolding - this represents the structure for sending updates to a main holding data record and optionally any main lease, occupation or vacant space details.
- CECALease - this represents the structure for sending updates to a main lease data record and optionally any landlord or break/review details.
- CECAOccupation - this represents the structure for sending updates to a main occupation data record and optionally any tenant, lease or vacant space details.

4.1.1 CECA Data element unique key definition

The 7 architecture schemas defined to support the CECA schema structures each represent a specific type of data as defined in section 2.2. Each element of data requires a unique identifier to be applied and this is established by the inclusion in each schema of an element which represents a unique identifier for the data.
The scope of CECA means that property related data can be sourced from 2 disparate management systems, the central OGC database which supports the ePIMS application and a specific departments own internal property management system. This raises the likelihood that a property or its related data could have 2 identifiers (ie OGC and departmental). The elements in each schema relating to the unique identifier have to be able to support either an OGC ID or a departmental ID or both, with the rule that at least one identifier must exist. An example is detailed below relating to a holding element

```
<xsd:element name="HoldingReference" type="HoldingReferenceDetailsStructure"/>
```

This is the main element specifies the actual unique identifier. This references the complexType HoldingReferenceDetailsStructure.

```
<xsd:complexType name="HoldingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptHoldingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
```

```
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCHoldingRef" type="OGCHoldingKeyStructure"/>
            <xsd:element name="DeptHoldingRef" type="DeptHoldingKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

This complexType provides a structure indicating that any element derived from it must choose between the elements OGCDeptHoldingRef, OGCHoldingRef or DeptHoldingRef. The element OGCDeptHoldingRef is implemented as a complex type and ensures that all references must be written based on the base types defined in the CECADataStructures schema.
The other 2 elements defined are derived from the complexTypes OGCHoldingKeyStructure and DeptHoldingKeyStructure.

```
<xsd:complexType name="OGCHoldingKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeptHoldingKeyStructure">
    <xsd:sequence>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
```

The complexTypes defined above are derived from simpleTypes defined in the main CECADataStructures schema. These complexTypes define the actual structure required to define the unique holding identifier. In this example a Holding identifier is a complex key comprising a unique property number and a holding number. The 2 complex types defined relate to either a complex key structure for OGC references or a key structure for departmental references.

4.2 CECA Architecture schemas

The schemas detailed below provide an underlying set of data types and data structures which are used by the messaging schemas detailed in section 3.3. These schemas provide the data structures detailed in the original CECA agreement and supply data types and data validation, enumeration and selection structure to support CECA.
Each architecture schema is detailed below.
NOTE: these schemas are not intended to be used to generate XML documents. They are there to provide common data structures to the messaging schemas set up to provide a mechanism for data transfer.

4.2.1 CECA Data Types

This schema provides common CECA related data types which can be used by all other architecture schemas. A list of each type is detailed below:

- **OGCPropertyReferenceType**: this is the unique property identifier held by OGC within the pisces database
- **OGCBuildingReferenceType**: this is the unique building identifier held by OGC within the pisces database
- **OGCHoldingReferenceType**: this is the unique holding identifier held by OGC within the pisces database
- **OGCLeaseReferenceType**: this is the unique lease identifier held by OGC within the pisces database
- **OGCOccupationReferenceType**: this is the unique occupation identifier held by OGC within the pisces database
- **OGCLandlordReferenceType**: this is the unique landlord identifier held by OGC within the pisces database
- **OGCOccupationReferenceType**: this is the unique tenant identifier held by OGC within the pisces database
- **DepartmentPropertyReferenceType**: this is the unique property identifier held by the department within their own estate management systems
- **DepartmentBuildingReferenceType**: this is the unique building identifier held by the department within their own estate management systems
- **DepartmentHoldingReferenceType**: this is the unique holding identifier held by the department within their own estate management systems
- **DepartmentLeaseReferenceType**: this is the unique lease identifier held by the department within their own estate management systems
- **DepartmentOccupationReferenceType**: this is the unique occupation identifier held by the department within their own estate management systems
- **DepartmentLandlordReferenceType**: this is the unique landlord identifier held by the department within their own estate management systems
- **DepartmentTenantReferenceType**: this is the unique tenant identifier held by the department within their own estate management systems
- **StatusIndicatorType**: this is a basic type used to define a boolean state. The values Y/N are specified via enumeration. Use of this type is intended to implement a boolean standard for CECA data.
- **DataDescriptionType**: this is a basic type used by data structure schemas to allow textual descriptions of the data to be applied.
- **DepartmentCodeType**: used to represent a short textual code representing a department or a departments location. This is based on codes held within the master OGC Civil Estate database.

- **PropertyCentreCodeType**: used to represent a short textual code representing a property centre or a property centre. This is based on codes held within the master OGC Civil Estate database.
- **CECAFlagType**: the CECA format has some data fields mapped as flags. This is in essence another boolean type structure but specifically for CECA flag fields which do not conform to the standard in StatusIndicator. Validity to be discussed.
- **DepartmentalDataReferenceType**: OGC pisces database CECA data contains references to departmental data indicators. These are not the same as the Departmental data ID's specified above and apply to property, building and holding. Validity to be discussed … not used in architecture schemas.
- **FloorAreaType**: the type of floor area related to a holding or occupation
- **TermPeriodType**: represents different period indicators (eg year, month, week, day etc)
- **ConsentType**: where consent (whether legal or not) is required. This is a list of available responses.
- **PaymentFrequencyType**: relating to payment of fees, rents etc and supplies a list of the different forms of payment duration.
- **VatBandType**: represents the different types of VAT expense appluied to a monetary data item
- **VacancyCodeType**: represents the codes relating to the types of vacant space.
- **VacantStatusType**: represents the codes relating to vacant space status.
- **RefurbReqdType**: represents the refurbishment codes relating to a vacant space.
- **OSGridReferenceEastType**: The easterly Ordnance Survey grid reference relating to the property's geographical location.
- **OSGridReferenceNorthType**: The northerly Ordnance Survey grid reference relating to the property's geographical location.
- **DepartmentalNotesTypeType**: This type has been defined to allow remarks/comments to be applied to a schema. It is to be applied to freeform string based elements for within the CECA Data Structure messaging schemas.
- **CECADateFormatType**: Standard date format as defined in W3C specification pattern is CCYY-MM-DD
- **CECATimeFormatType**: Standard time format as defined in W3C specification pattern is hh:mm:ss
- **EmailAddressType**: common type representing an Email address.
- **TelephoneNumberType**: common type representing a telehone number
- **ePIMSUserIDType** : common type representing structure of an Epims user ID
- **CECADateStructure**: This is a complex type derived from the date and time types defined above. This gives the XML document generator the option to specify a date as either date only or date and time.
- **CECAMessageSignature**: complexType defining elements used to "sign" an individual message
- **VacantSpaceStructure**: complexType defining the structure of a vacant space data element.

4.2.1.1 CECADataTypes UML and XML Schema

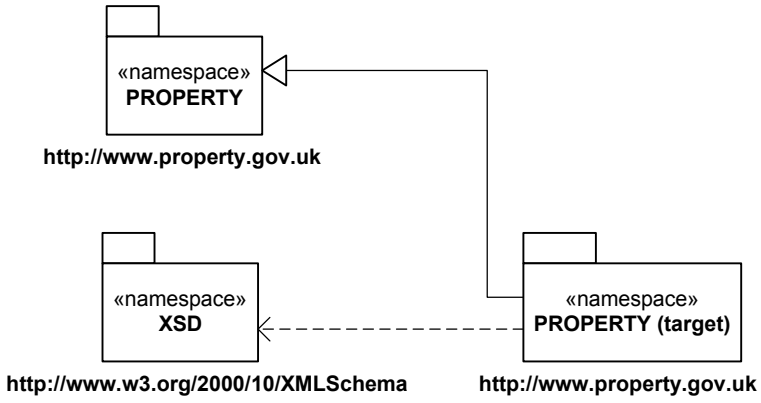


Figure 8: CECADataTypes - namespace

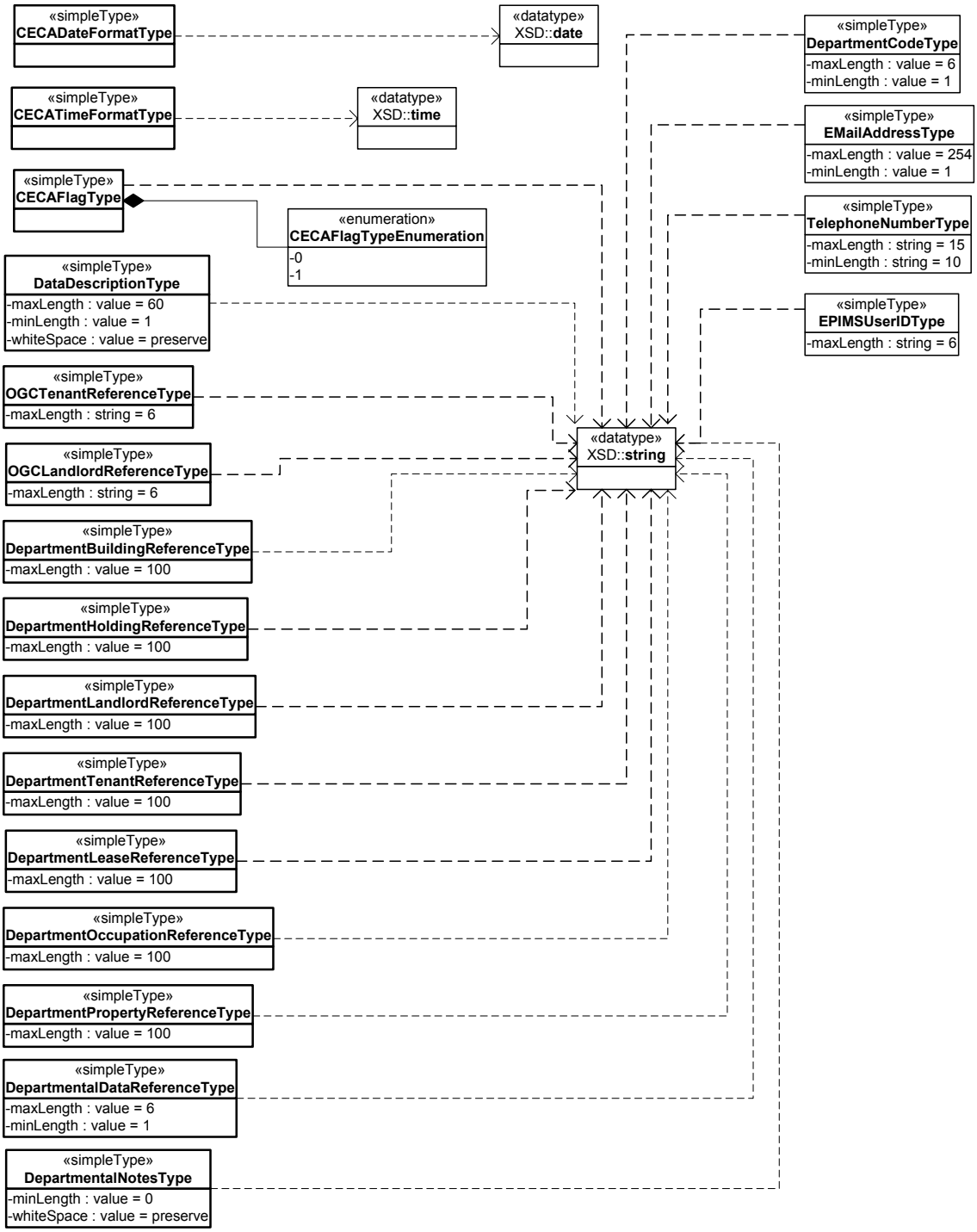


Figure 9: CECADataTypes - simpleTypes 1

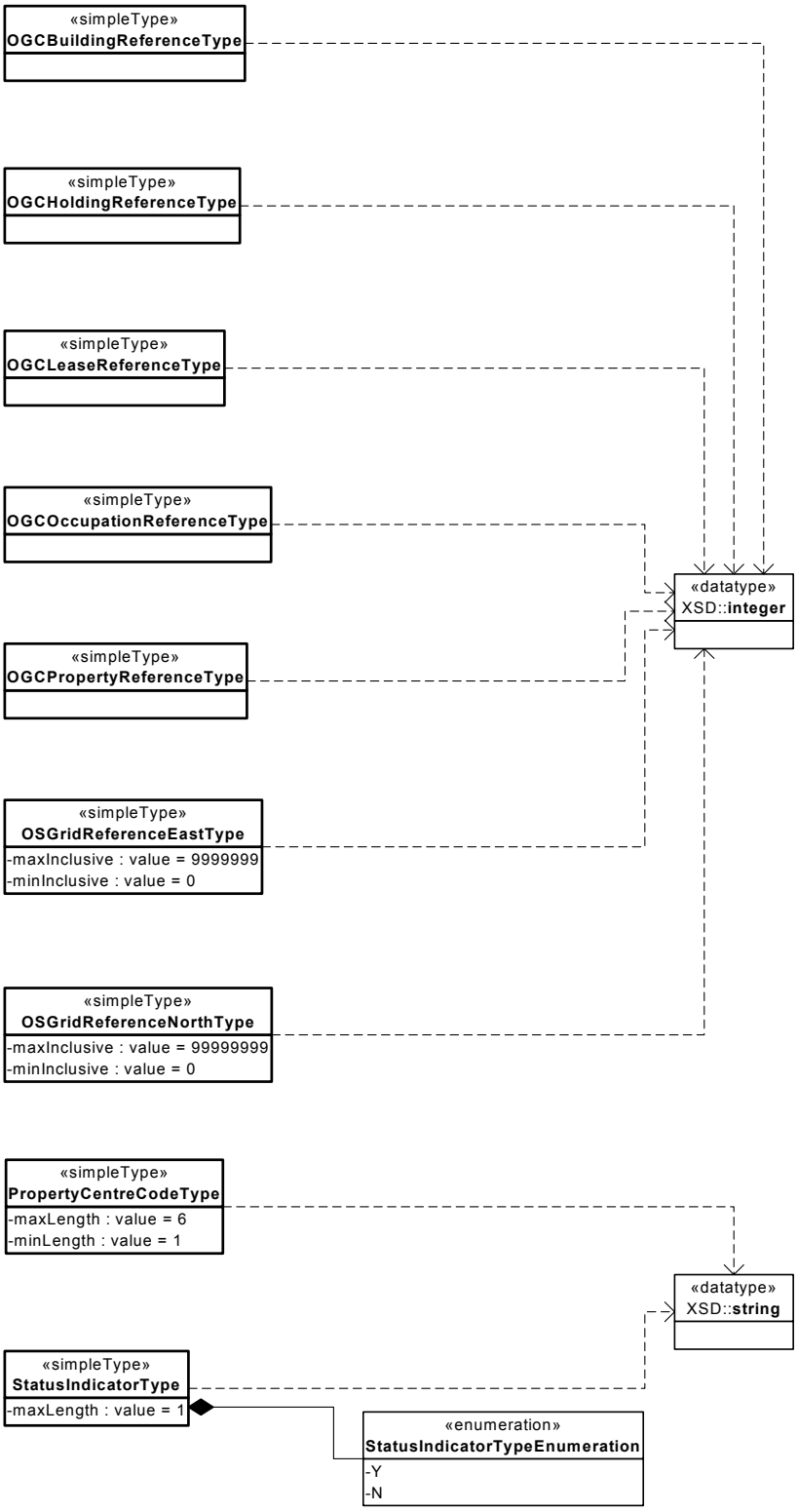


Figure 10: CECADataTypes - simpleTypes 2

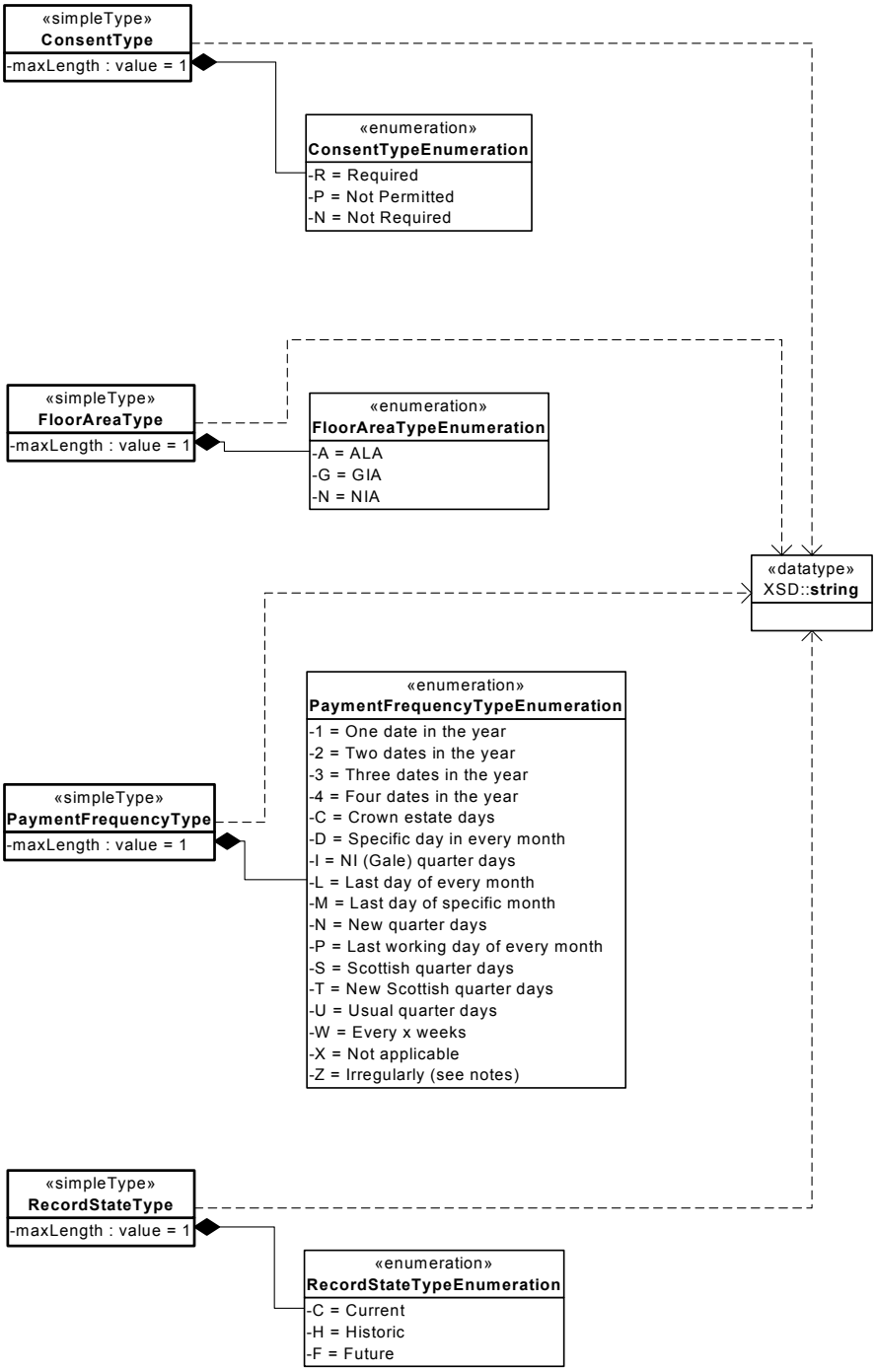


Figure 11: CECADataTypes - simpleTypes 3

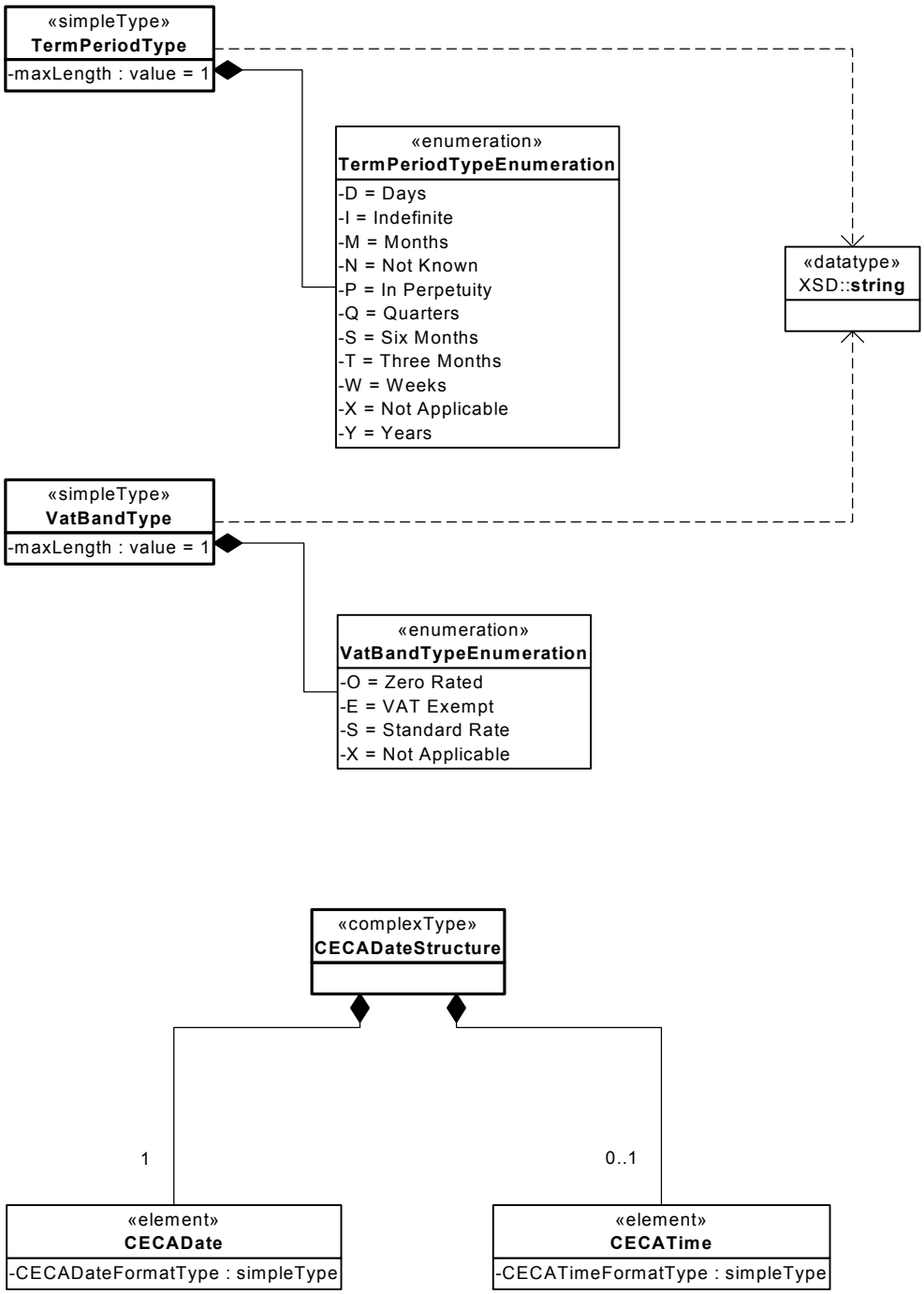


Figure 12: CECADataTypes - CECADataTypes simpleTypes 4 (top), CECADateStructure complexType (bottom)

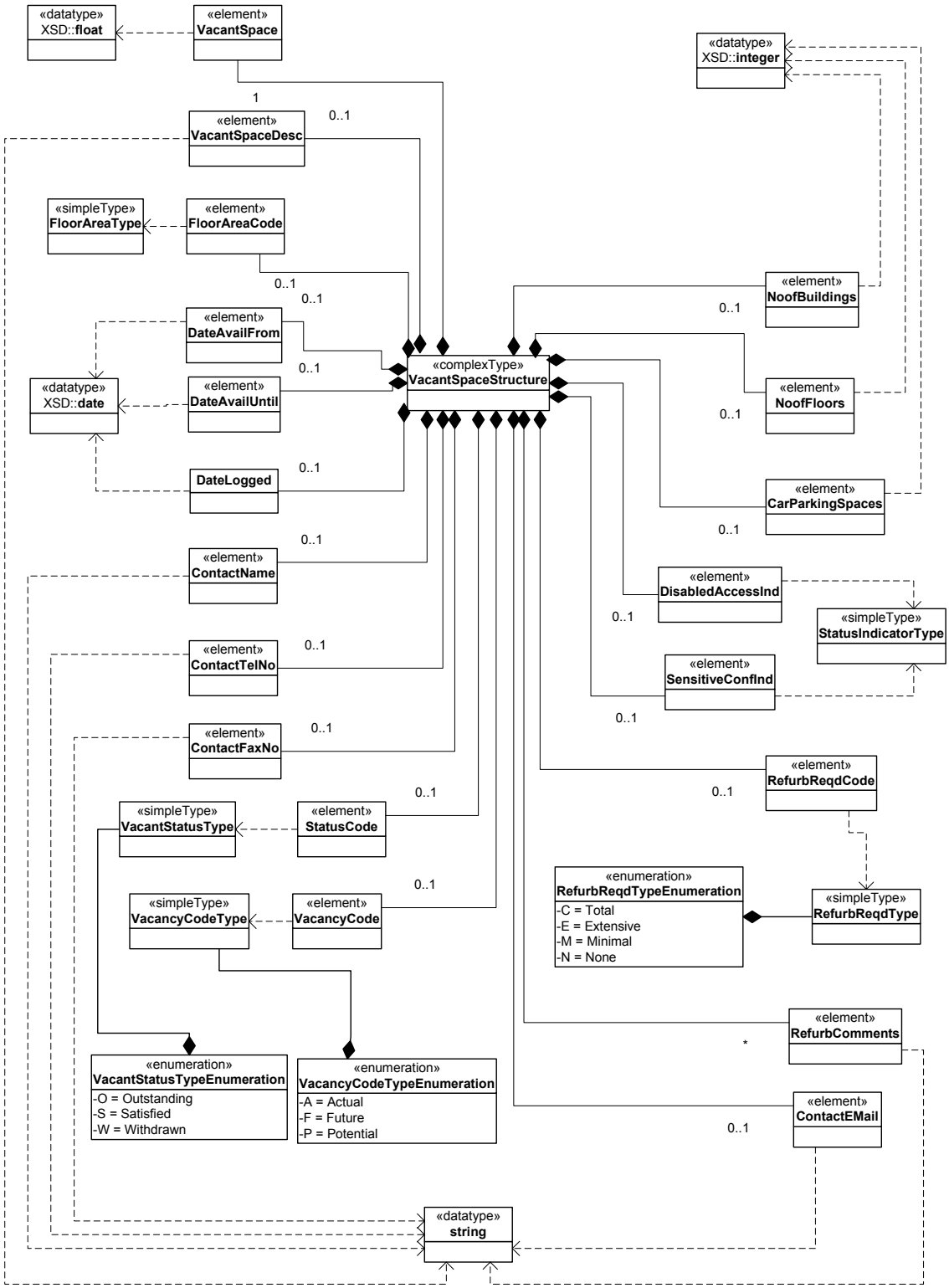


Figure 13 : CECADataTypes – complexType VacantSpace

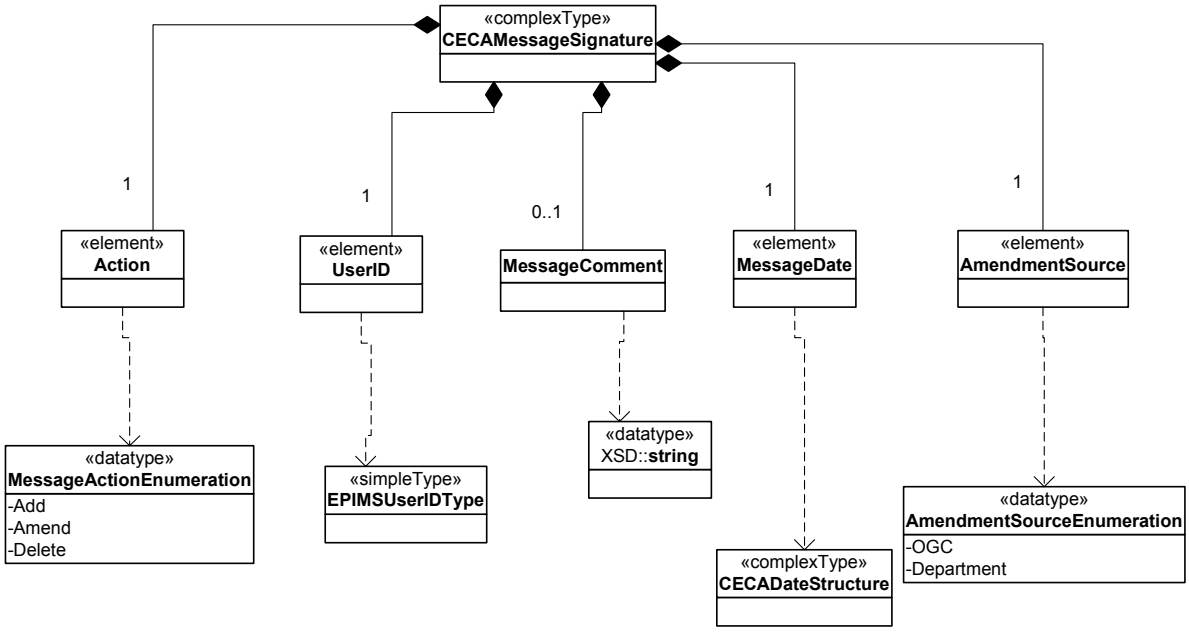


Figure 14 : CECADataTypes – complexType CECAMessageSignature

Figure 15: the XML schema definition for CECADataTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="CECADataTypes">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Common CECA datatypes

    Purpose: This schema is used to supply common data types to architecture and message schemas used throughout
the entire CECA schema structures

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides common data types to be used throughtout the CECA XML schemas
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAAddressStructure.xsd"/>
    <xsd:simpleType name="OGCPropertyReferenceType">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="OGCBuildingReferenceType">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="OGCHoldingReferenceType">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="OGCLeaseReferenceType">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="OGCOccupationReferenceType">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="OGCLandlordReferenceType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="6"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="OGCTenantReferenceType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="6"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="DepartmentPropertyReferenceType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="100"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="DepartmentBuildingReferenceType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="100"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="DepartmentHoldingReferenceType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="100"/>
        </xsd:restriction>
```

```
</xsd:simpleType>
<xsd:simpleType name="DepartmentLeaseReferenceType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="100"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DepartmentOccupationReferenceType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="100"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DepartmentLandlordReferenceType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="100"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DepartmentTenantReferenceType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="100"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StatusIndicatorType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="1"/>
        <xsd:enumeration value="Y"/>
        <xsd:enumeration value="N"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DataDescriptionType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="255"/>
        <xsd:minLength value="1"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DepartmentCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="6"/>
        <xsd:minLength value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="PropertyCentreCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="6"/>
        <xsd:minLength value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="CECAFlagType">
    <xsd:restriction base="xsd:integer">
        <xsd:enumeration value="0"/>
        <xsd:enumeration value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DepartmentalDataReferenceType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="6"/>
        <xsd:minLength value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="FloorAreaType">
    <xsd:annotation>
        <xsd:documentation>
            The following descriptions apply to the enumerated types listed below
                A:ALA
                G:GIA
                N:NIA
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="1"/>
        <xsd:enumeration value="A"/>
```

```
            <xsd:enumeration value="G"/>
            <xsd:enumeration value="N"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="TermPeriodType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    D: Days
                    I: Indefinite
                    M: Months
                    N: Not Known
                    P: In Perpetuity
                    Q: Quarters
                    S: Six Months
                    T: Three Months
                    W: Weeks
                    X: Not Applicable
                    Y: Years
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="I"/>
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="N"/>
            <xsd:enumeration value="P"/>
            <xsd:enumeration value="Q"/>
            <xsd:enumeration value="S"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="W"/>
            <xsd:enumeration value="X"/>
            <xsd:enumeration value="Y"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ConsentType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        R: Required
                        P: Not Permitted
                        N: Not required
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="R"/>
            <xsd:enumeration value="P"/>
            <xsd:enumeration value="N"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="PaymentFrequencyType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        1: One date in the year
                        2: Two date in the year
                        3: Three dates in the year
                        4: Four dates in the year
                        C: Crown estate days
                        D: Specific day in every month
                        I: NI (Gale) quarter days
                        L: Last day of every month
                        M: Last day of specific month
                        N: New quarter days
                        P: Last working day of every month
                        S: Scottish quarter days
                        T: New Scottish quarter days
                        U: Usual quarter days
                        W: Every x weeks
```

```
                    X: Not applicable
                    Z: Irregularly (see notes)
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
            <xsd:enumeration value="4"/>
            <xsd:enumeration value="C"/>
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="I"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="N"/>
            <xsd:enumeration value="P"/>
            <xsd:enumeration value="S"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="U"/>
            <xsd:enumeration value="W"/>
            <xsd:enumeration value="X"/>
            <xsd:enumeration value="Z"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="VatBandType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    O: Zero Rated
                    E: VAT Exempt
                    S: Standard Rate
                    X: Not Applicable
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="O"/>
            <xsd:enumeration value="E"/>
            <xsd:enumeration value="S"/>
            <xsd:enumeration value="X"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="VacancyCodeType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    A   Actual
                    F   Future
                    P   Potential
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="A"/>
            <xsd:enumeration value="F"/>
            <xsd:enumeration value="P"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="VacantStatusType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    O   Outstanding
                    S   Satisfied
                    W   Withdrawn
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="O"/>
```

```
            <xsd:enumeration value="S"/>
            <xsd:enumeration value="W"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="RefurbReqdType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    C   Total
                    E   Extensive
                    M   Minimal
                    N   None
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="C"/>
            <xsd:enumeration value="E"/>
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="N"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="OSGridReferenceEastType">
        <xsd:restriction base="xsd:integer">
            <xsd:maxInclusive value="9999999"/>
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="OSGridReferenceNorthType">
        <xsd:restriction base="xsd:integer">
            <xsd:maxInclusive value="99999999"/>
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="DepartmentalNotesType">
        <xsd:annotation>
            <xsd:documentation>
                This type has been defined to allow remarks/comments to be applied to a schema. It is to be applied to
freeform string based elements for within the CECA Data Structure messaging schemas.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:minLength value="0"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="CECADateFormatType">
        <xsd:annotation>
            <xsd:documentation>
                Standard date format as defined in W3C specification pattern is CCYY-MM-DD
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:date"/>
    </xsd:simpleType>
    <xsd:simpleType name="CECATimeFormatType">
        <xsd:annotation>
            <xsd:documentation>
                Standard time format as defined in W3C specification pattern is hh:mm:ss
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:time"/>
    </xsd:simpleType>
    <xsd:simpleType name="EmailAddressType">
        <xsd:restriction base="xsd:string">
            <xsd:minLength value="1"/>
            <xsd:maxLength value="254"/>
            <xsd:pattern value="[0-9A-Za-z\.\-_]{1,127}@[0-9A-Za-z\.\-_]{1,127}"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="TelephoneNumberType">
        <xsd:restriction base="xsd:string">
```

```
            <xsd:minLength value="10"/>
            <xsd:maxLength value="15"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ePIMSUserIDType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="6"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CECADateStructure">
        <xsd:annotation>
            <xsd:documentation>
                This is a complex type derived from the date and time types defined above. This gives the XML
                document generator the option to specify a date as either date only or date and time.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="CECADate" type="CECADateFormatType"/>
            <xsd:element name="CECATime" type="CECATimeFormatType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CECAMessageSignature">
        <xsd:sequence>
            <xsd:element name="Action">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="Add"/>
                        <xsd:enumeration value="Amend"/>
                        <xsd:enumeration value="Delete"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="UserID" type="ePIMSUserIDType"/>
            <xsd:element name="MessageDate" type="CECADateStructure"/>
            <xsd:element name="AmendmentSource">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="OGC"/>
                        <xsd:enumeration value="DEPARTMENT"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="MessageComment" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="VacantSpaceStructure">
        <xsd:annotation>
            <xsd:documentation>
                This complex type defines the structure to represent a vacant space data entity. Vacant Space is
                declared as part of Holding and Occupation data structures.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="VacantSpace" type="xsd:float"/>
            <xsd:element name="VacantSpaceDesc" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="2000"/>
                        <xsd:whiteSpace value="preserve"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="VacancyCode" type="VacancyCodeType" minOccurs="0"/>
            <xsd:element name="FloorAreaCode" type="FloorAreaType" minOccurs="0"/>
            <xsd:element name="DateAvailFrom" type="CECADateFormatType"/>
            <xsd:element name="DateAvailUntil" type="CECADateFormatType" minOccurs="0"/>
            <xsd:element name="DateLogged" type="CECADateFormatType"/>
            <xsd:element name="StatusCode" type="VacantStatusType" minOccurs="0"/>
            <xsd:element name="ContactName">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
```

```
                    <xsd:maxLength value="35"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="ContactTelNo" type="TelephoneNumberType" minOccurs="0"/>
        <xsd:element name="ContactFaxNo" type="TelephoneNumberType" minOccurs="0"/>
        <xsd:element name="ContactEMail" type="EmailAddressType" minOccurs="0"/>
        <xsd:element name="NoofBuildings" type="xsd:integer"/>
        <xsd:element name="NoofFloors" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="CarParkingSpaces" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="DisabledAccessInd" type="StatusIndicatorType"/>
        <xsd:element name="RefurbReqdCode" type="RefurbReqdType"/>
        <xsd:element name="RefurbComments" type="xsd:string" minOccurs="0"/>
        <xsd:element name="SensitiveConfInd" type="StatusIndicatorType"/>
        <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

4.2.2 CECAAddressStructure Type and BS7666 relationship

Property addresses within OGC pisces have a simple structure, which is specific to the internal data structures within OGC pisces. This data structure is defined in the schema CECAAddressStructure as a complexType CECAAddressType.
However a British Standard for property addressing BS7666 has been implemented as another XML initiative by Office of eEnvoy (OeE).
A property requires an address, so the property structure schema (detailed below 3.2.3) has the flexibility to supply either the CECA OGC pisces structured address or a BS7666 compliant address. This is done via a complexType PropertyAddressType.

There are a number of simple Types defined within the schema. These are intended to support the OGC pisces address format and are detailed below

- **PropertyDescriptorType**: a textual description of the property
- **AddressPostcodeAreaType**: the area part of the postcode (ie the first part)
- **AddressPostcodeStreetType**: the street part of a postcode
- **CECAAddressStreetNumberType**: street number
- **CECAAddressRoadType**: name of road
- **CECAAddressNeighbourhoodType**: name of neighbourhood or district
- **CECAAddressTownType**: name of town
- **CECAAddressCountyType**: name of county (this is not controlled by an set of enumerations restricting selection. This is open for discussion as to validity)
- **UPRNType**: Unique Property Reference Numbe UK initiative for unique identification of property.
- **CECAAddressStructure**: the complete CECA address format
- **PropertyAddressStructure**: a choice between CECA or BS7666 address formats

4.2.2.1 CECAAddressStructure UML and XML Schema

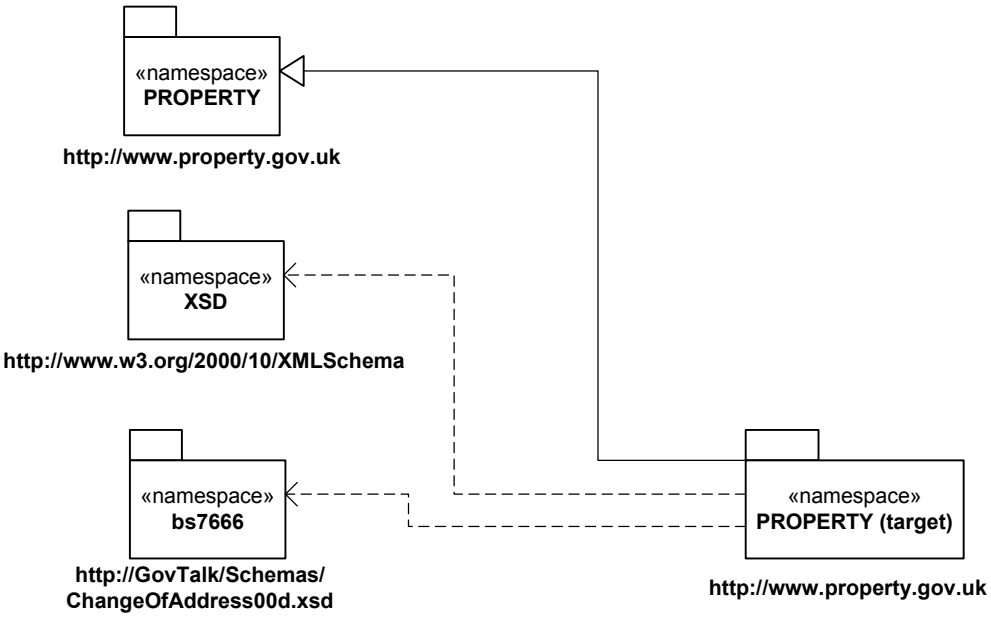


Figure 16: CECAAddressStructure - namespace

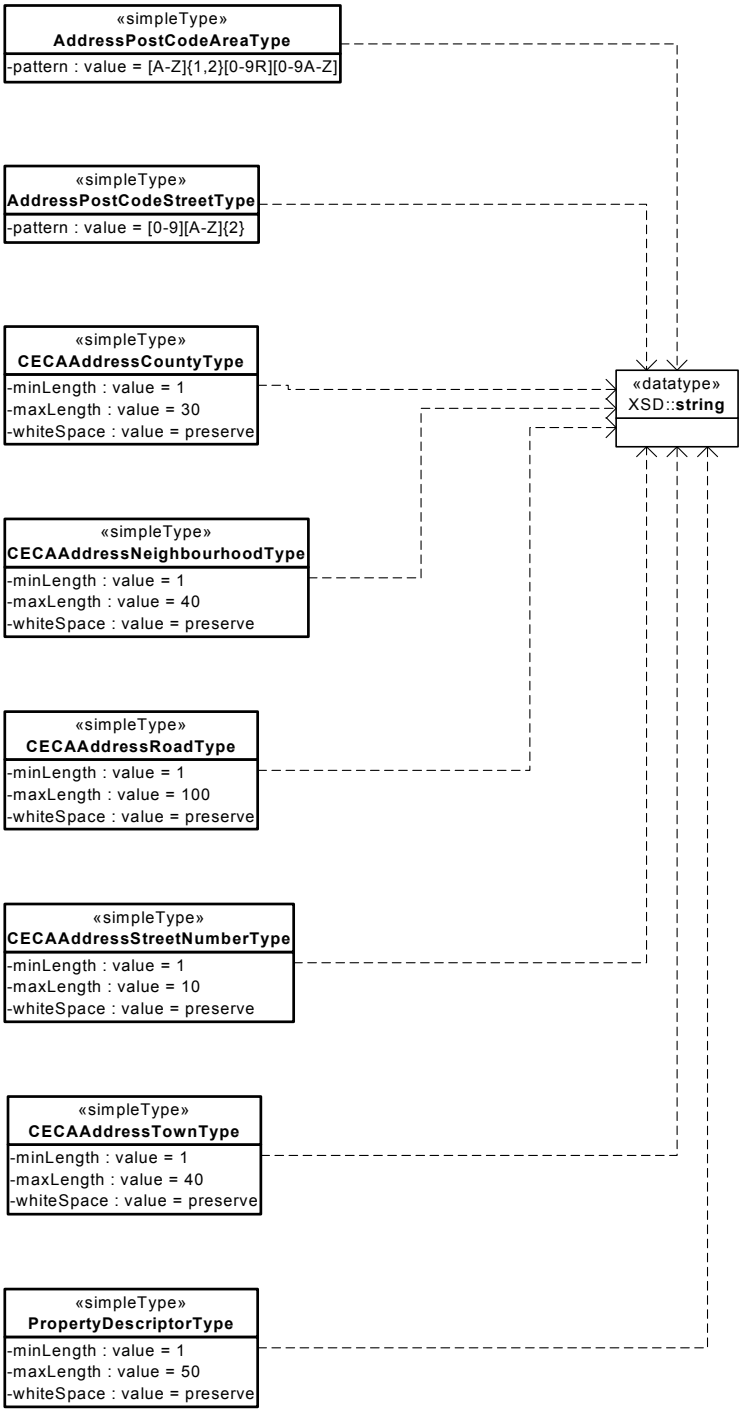


Figure 17: CECAAddressStructure - simpleTypes1

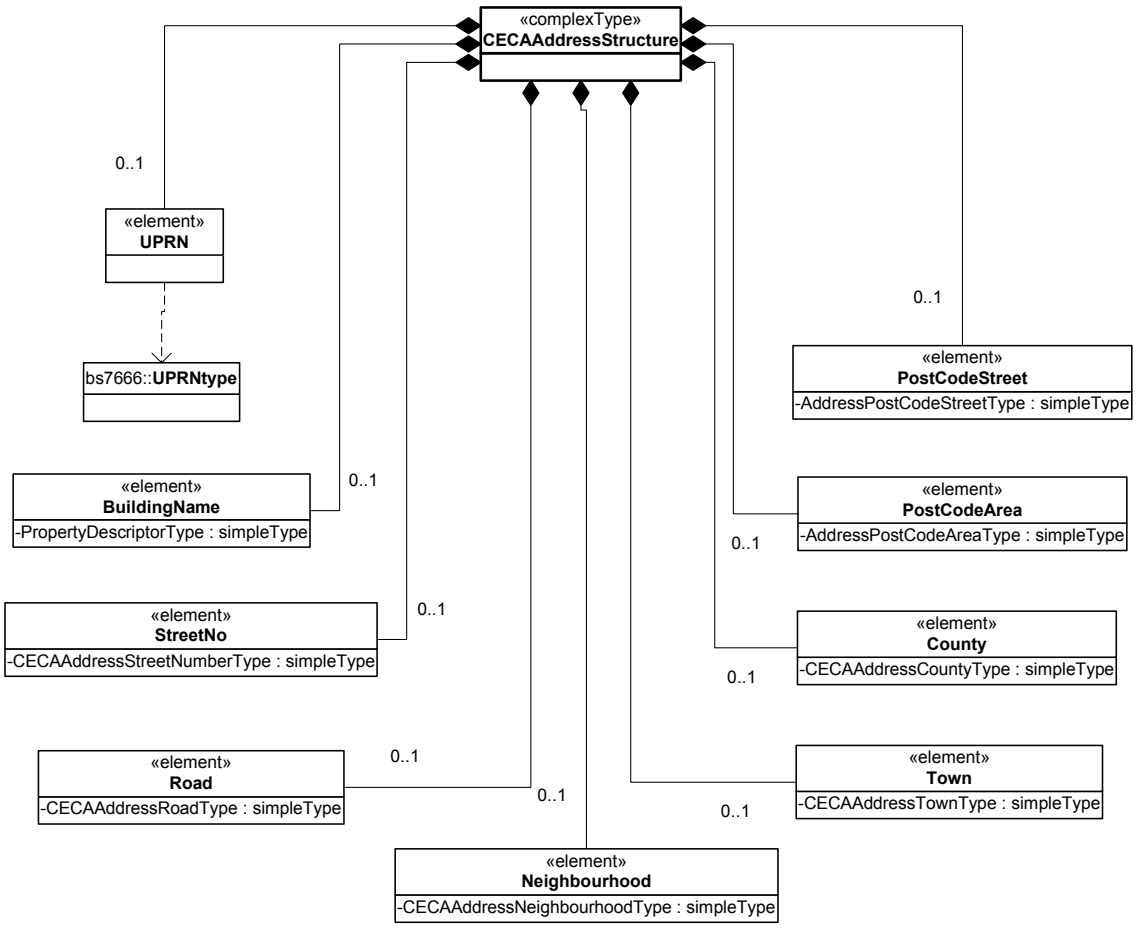


Figure 18: CECAAddressStructure – CECAAddressStructure complexType

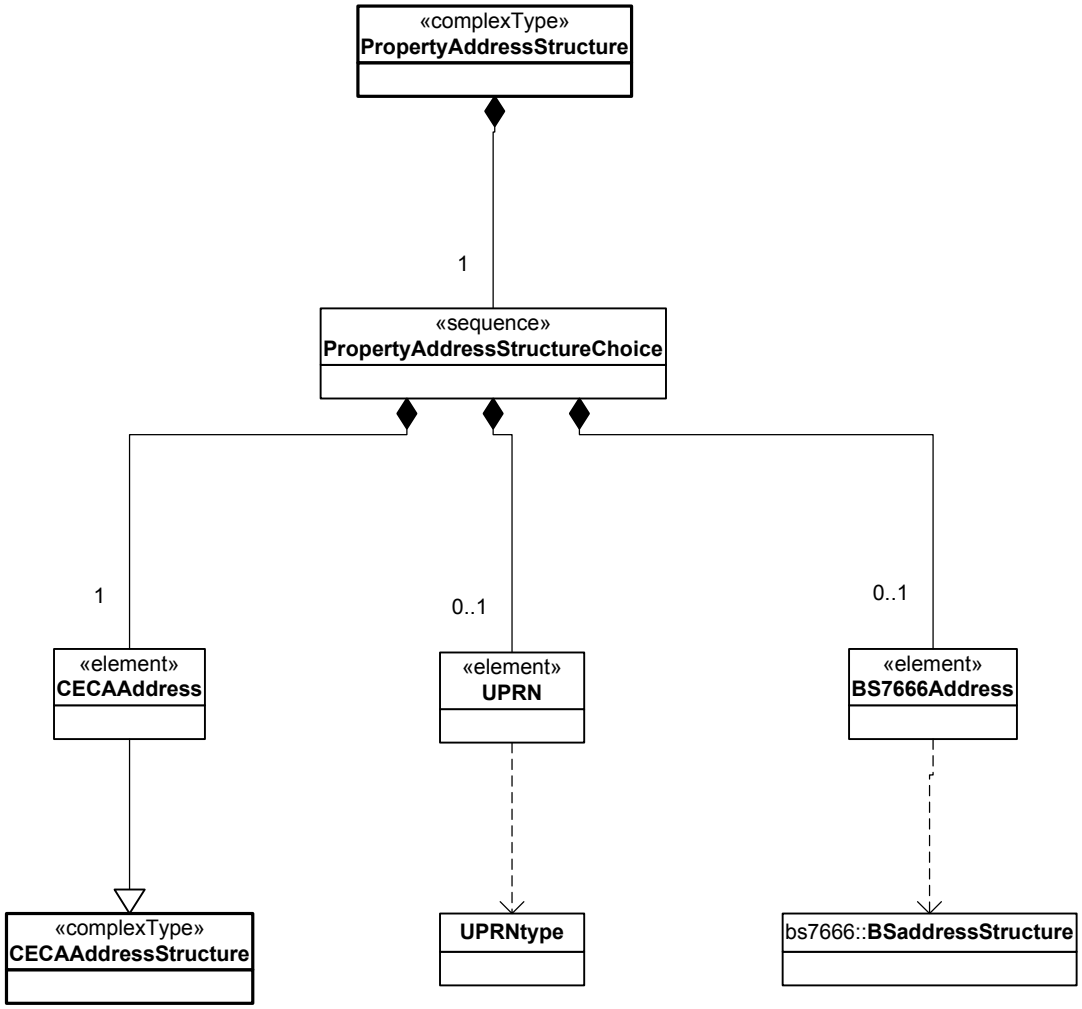


Figure 19: CECAAddressStructure – PropertyAddressStructure complexType

Figure 20: the XML schema definition for CECAAddressStructure.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="CECAAddressStructure">
    <!--xmlns:bs7666="http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd"-->
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Common CECA address structure

    Purpose: This schema is used to supply the CECA property addressing structures to architecture and message
schemas    used throughout the entire CECA schema structures

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
    </xsd:annotation>
    <!--
    <xsd:import namespace="http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd"
schemaLocation="http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd"/>
    -->
    <xsd:simpleType name="PropertyDescriptorType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="50"/>
            <xsd:minLength value="1"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="AddressPostCodeAreaType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[A-Z]{1,2}[0-9R]?[0-9A-Z]"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="AddressPostCodeStreetType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[0-9][A-Z]{2}"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="CECAAddressStreetNumberType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="10"/>
            <xsd:minLength value="1"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="CECAAddressRoadType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="100"/>
            <xsd:minLength value="1"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="CECAAddressNeighbourhoodType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40"/>
            <xsd:minLength value="1"/>
```

```
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="CECAAddressTownType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40"/>
            <xsd:minLength value="1"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="CECAAddressCountyType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="30"/>
            <xsd:minLength value="1"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- UPRN Definition -->
    <xsd:simpleType name="UPRNtype">
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="1"/>
            <xsd:maxInclusive value="999999999999"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CECAAddressStructure">
        <xsd:sequence>
            <xsd:element name="BuildingName" type="PropertyDescriptorType" minOccurs="0"/>
            <xsd:element name="StreetNo" type="CECAAddressStreetNumberType" minOccurs="0"/>
            <xsd:element name="Road" type="CECAAddressRoadType" minOccurs="0"/>
            <xsd:element name="Neighbourhood" type="CECAAddressNeighbourhoodType" minOccurs="0"/>
            <xsd:element name="Town" type="CECAAddressTownType"/>
            <xsd:element name="County" type="CECAAddressCountyType" minOccurs="0"/>
            <xsd:element name="PostCodeArea" type="AddressPostCodeAreaType" minOccurs="0"/>
            <xsd:element name="PostCodeStreet" type="AddressPostCodeStreetType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="PropertyAddressStructure">
        <xsd:sequence>
            <xsd:element name="CECAAddress" type="CECAAddressStructure"/>
            <xsd:element name="UPRN" type="UPRNtype" minOccurs="0"/>
            <!--
            <xsd:element name="UPRN" type="bs7666:UPRNtype" minOccurs="0"/>
            <xsd:element name="BS7666Address" type="bs7666:BSaddressStructure" minOccurs="0"/>
            -->
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.3 Property Structure

This schema supports the architecture required for a CECA property data entity. It provides a complexType *PropertyStructure* that is used by the messaging schemas.

PropertyStructure defines the data elements of a CECA property object and also defines the object relationships for Building and Holding as defined in the CECA objectc hierarchy model. These relational elements as described below

Building

The buildings element defines a sequence of zero to many Building elements based on the structure implemented in the schema BuildingStructure.

```
<xsd:element name="Buildings" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Building" type="BuildingStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Holding

The holdings element defines a sequence of zero to many Holding elements based on the structure implemented in the schema HoldingStructure.

```
<xsd:element name="Holdings" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Holding" type="HoldingStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

In addition it also supplied a number of simple types which provide an enumerated list of valid values for specific data items. These are listed below.

- **ListedBuildingsType**: this contains a list of all valid listed building codes

Other Complex Type definitions for PropertyStructure are

- **OSGridRefsStructure**: defining elements related to Northerly and Easterly OS grid references

Owner Property Centre

An element is defined in the structure which relates to the OGC property centre code of the property "owner". The "owner" is defined as the property centre who has update permissions to the data held against a specific property record.

```
<xsd:element name="OwnerPropertyCentre" type="PropertyCentreCodeType"/>
```

4.2.3.1 Unique Identifier

There is a complex Type defined *PropertyReferenceDetails*, this provide the XML document author with the ability to supply unique property identifiers based on the types *OGCPropertyReference* or *DepartmentPropertyReference* OR BOTH (these types are sourced from CECADataTypes.xsd).

A property can be uniquely identified by a single reference (to be known as the property number). An XML document "author" can therefore specify the property identifier as either a unique OGC property number (defined as simpleType *OGCPropertyReferenceType*) OR a unique departmental property number (defined as simpleType *DepartmentPropertyReferenceType*) OR BOTH.
The structure for this is defined in the complexType *PropertyReferenceDetailsStructure.*

```
<xsd:element name="PropertyReference" type="PropertyReferenceDetailsStructure"/>

<xsd:complexType name="PropertyReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptPropertyRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

4.2.3.2 PropertyStructure UML and XML Schema

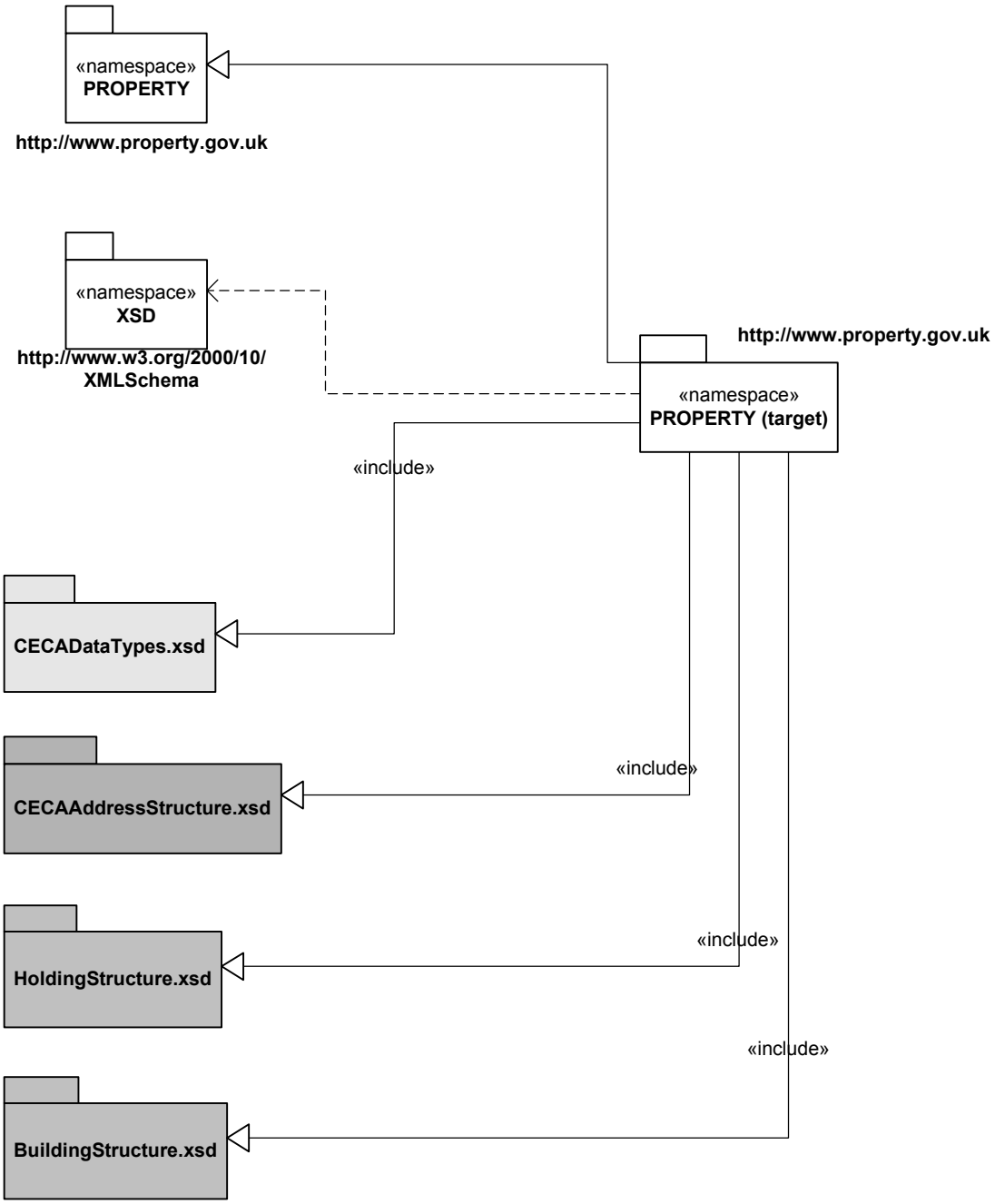


Figure 21: PropertyStructure - namespace

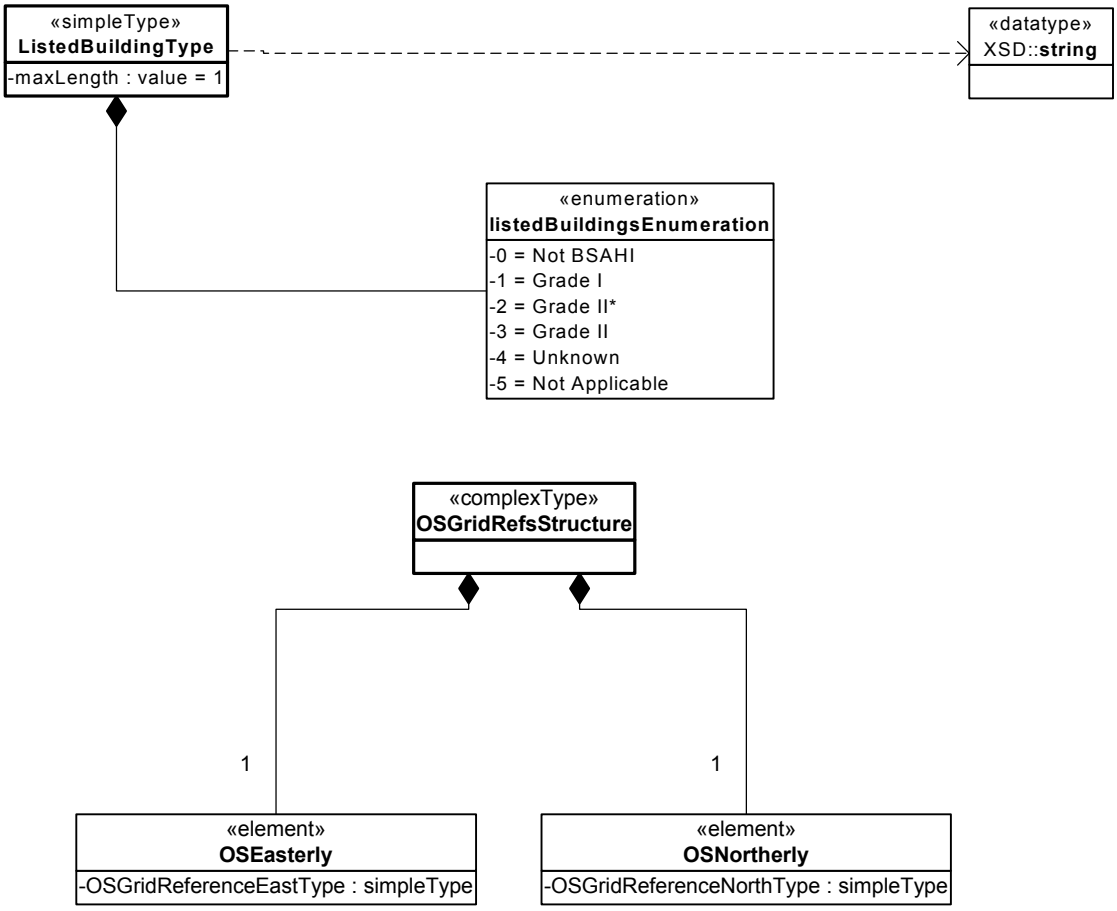
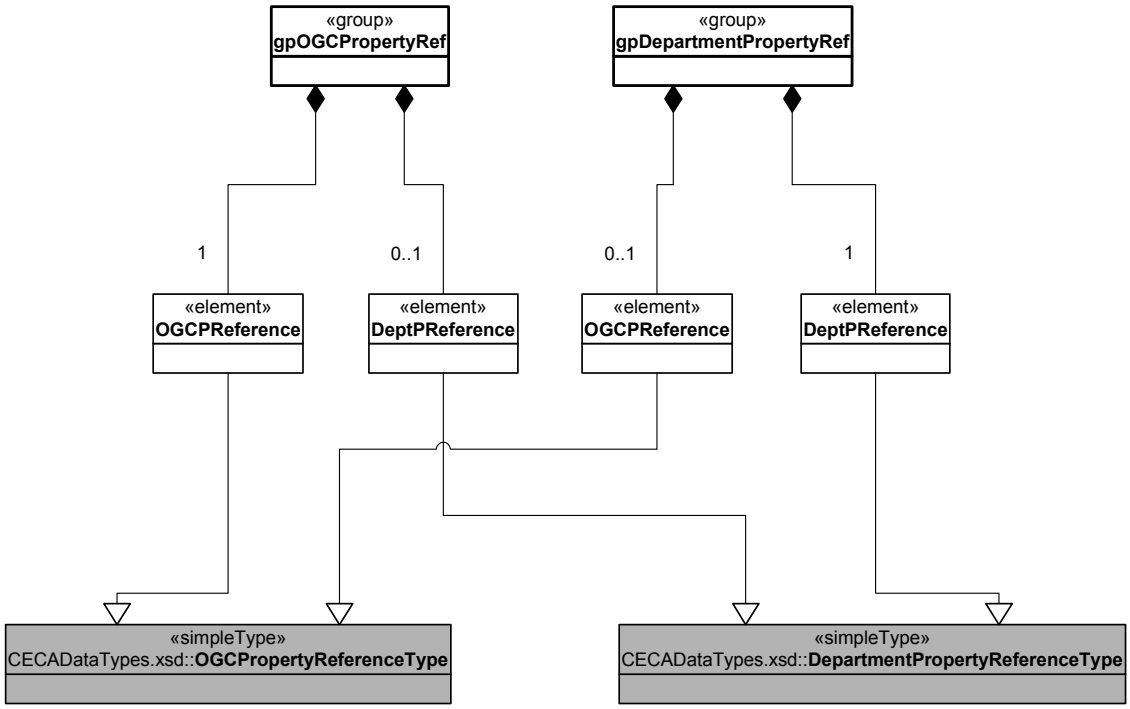


Figure 22: PropertyStructure – ListedBuildingType simpleType (top), OSGridRefsStructure complexType (bottom)



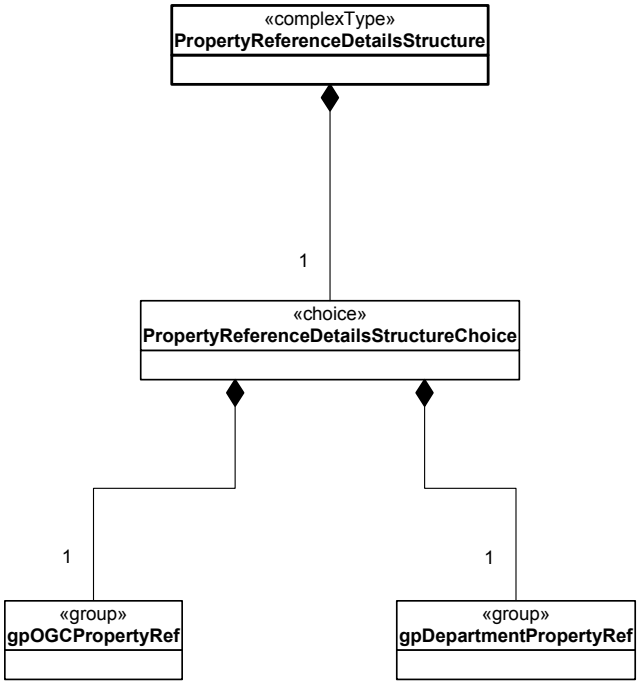


Figure 23: PropertyStructure - groups (top), PropertyReferenceDetailsStructure complexType (bottom)

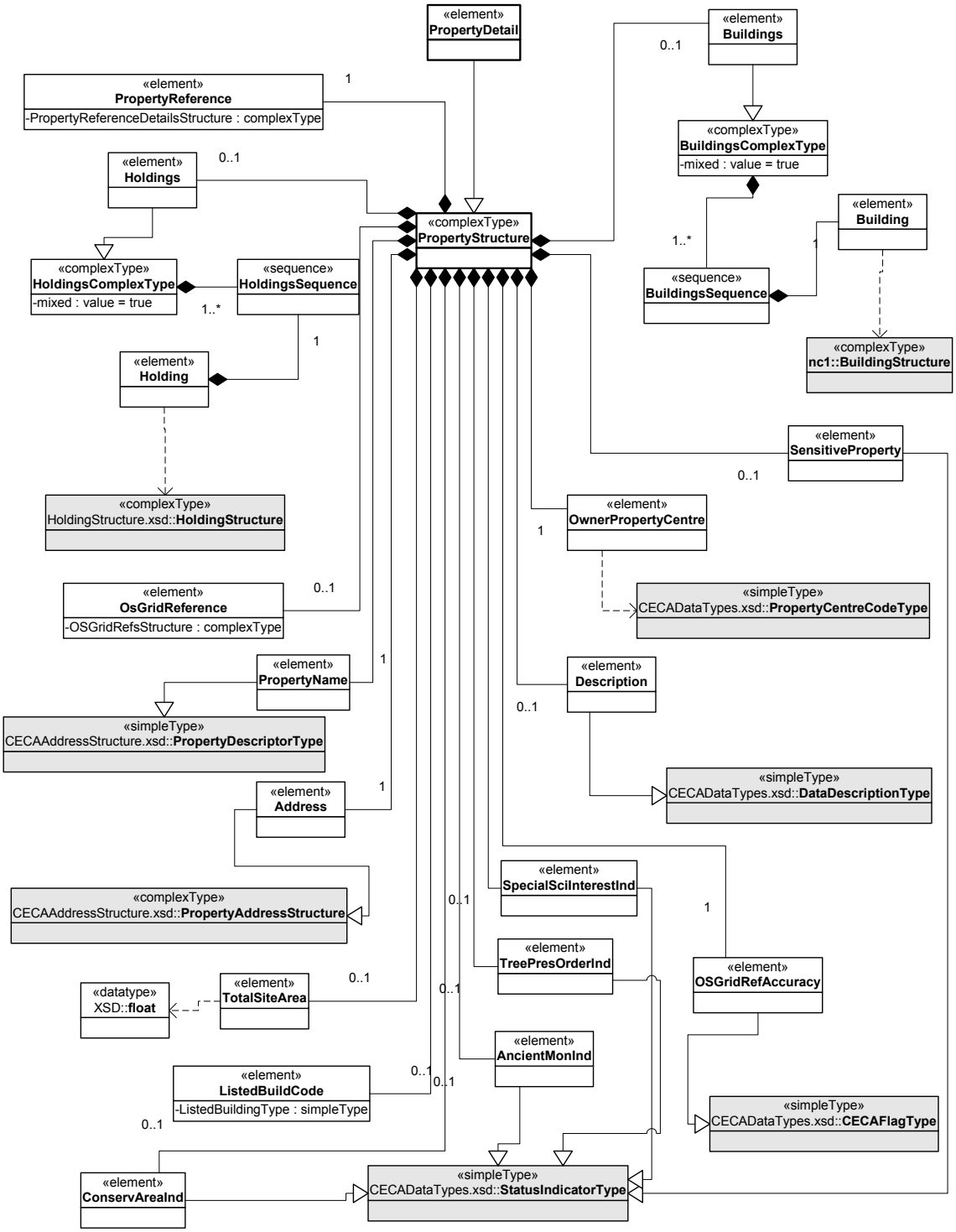


Figure 24: PropertyStructure – PropertyDetail element and PropertyStructure complexType

Figure 25: the XML schema definition for PropertyStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="PropertyStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Property data

    Purpose: This schema is used to supply the base property data structures for use with property messaging
schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core property details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAAddressStructure.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/HoldingStructure.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/BuildingStructure.xsd"/>
    <xsd:simpleType name="ListedBuildingType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        0: Not BSAHI
                        1: Grade I
                        2: Grade II*
                        3: Grade II
                        4: Unknown
                        5: Not Applicable
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="0"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
            <xsd:enumeration value="4"/>
            <xsd:enumeration value="5"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--The main element feeding off the PropertyStructure Complex Type-->
    <xsd:element name="PropertyDetail" type="PropertyStructure"/>
    <!--The main property schema structure-->
    <xsd:complexType name="PropertyStructure">
        <xsd:sequence>
            <xsd:element name="PropertyReference" type="PropertyReferenceDetailsStructure">
                <xsd:annotation>
                    <xsd:documentation>
                        This is the unique property reference which can be expressed in the following 3 formats
                        1. Using OGC property reference indicators. These are ID's used to reference properties within
the  OGC property reference database. Uses the simpleType OGCPropertyReference as a base.
```

2. Using departmental property reference ID's. These are the ID's used within a departments property management system (either in digital or manual format). Uses the simple type DepartmentPropertyReference as a base.

3. An option to specify both. The xml document generator can specify both refrerence numbers for completeness.

```
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="OwnerPropertyCentre" type="PropertyCentreCodeType"/>
    <xsd:element name="OsGridReference" type="OSGridRefsStructure" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
            This represents the OS grid ref data for property location. The XML document author must provide easterly and northerly together or not at all.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="PropertyName" type="PropertyDescriptorType">
        <xsd:annotation>
            <xsd:documentation>The OGC/Departmental name given to the property</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="Address" type="PropertyAddressStructure">
        <xsd:annotation>
            <xsd:documentation>
                The xml document generator has an option as to which address format to use against the property. There are 2 choices
                1. An address based on the OGC internal address format. This references the type CECAAddressType located in the schema  CECAAddressStructure.xsd
                2. An address which is compliant to the bs7666 address format. The xml document uses the type BSaddressStructure which is defined in the main bs7666 schema located in govtalk.

                NOTE: Only 1 address format is acceptable. The preferred option for departments being bs7666.

            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="TotalSiteArea" type="xsd:float" minOccurs="0"/>
    <xsd:element name="ListedBuildCode" type="ListedBuildingType" minOccurs="0"/>
    <xsd:element name="ConservAreaInd" type="StatusIndicatorType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Is property located in a Conservation Area</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="AncientMonInd" type="StatusIndicatorType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Does property have anchient monument status</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="TreePresOrderInd" type="StatusIndicatorType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Is the property subject to a tree preservation order related to one or more trees located within the boundaries of the property</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="SpecialSciInterestInd" type="StatusIndicatorType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Is the property of special scientific interest</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="OSGridRefAccuracy" type="xsd:integer"/>
    <xsd:element name="Description" type="DataDescriptionType" minOccurs="0"/>
    <xsd:element name="SensitiveProperty" type="StatusIndicatorType" minOccurs="0"/>
    <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
    <xsd:element name="Buildings" minOccurs="0">
        <xsd:complexType mixed="true">
            <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="Building" type="BuildingStructure"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
```

```
            <xsd:element name="Holdings" minOccurs="0">
                <xsd:complexType mixed="true">
                    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                        <xsd:element name="Holding" type="HoldingStructure"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="PropertyReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="OGCDeptPropertyRef">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="OSGridRefsStructure">
        <xsd:sequence>
            <xsd:element name="OSEasterly" type="OSGridReferenceEastType"/>
            <xsd:element name="OSNortherly" type="OSGridReferenceNorthType"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.4 Building Structure

This schema supports the architecture required for a CECA property data entity. It provides a complexType *BuildingStructure* that is used by the messaging schemas.

This architecture schema also has part of the CECA relational structure implemented within it. Part of the CECA relational definition states

"a building can have many associated holdings and a holding can have many associated buildings"

This many to many relationship has been simplified and agreed for this schema design. The building entity is seen as master so we use the statement

"a building has zero or many associated holdings"

This relationship has been implemented in the BuildingStructure schema by allowing an XML document generator to simply provide a list of associated holding ID's. Validation of the holding ID's is then the responsibility of the target application.

```
<xsd:element name="Holdings" type="AssociatedHoldingsStructure" minOccurs="0"/>

<xsd:complexType name="AssociatedHoldingsStructure">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="linkedHolding" type="FGNHoldingReferenceDetailsStructure"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FGNHoldingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptHoldingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
                        <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
            <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

In addition it also supplied a number of simple types which provide an enumerated list of valid values for specific data items. These are listed below.

- **BuildingType**: type of building based on structure
- **BuildingUsagesType**: The commercial usage type of the building
- **ConstructionBandsType**: A list of year based periods relating to the date of construction.
- **FloorType**: a list of material used in the floor construction.
- **BuildingElementDescriptorShortType:** a short description of the building

4.2.4.1 Unique Identifier

There is a complex Type defined *BuildingReferenceDetails*, this provide the XML document author with the ability to supply unique building identifiers, which is a combination of a unique property and building ID, based on the types *OGCPropertyReference/OGCBuildingReference* or *DepartmentPropertyReference/DepartmentBuildingReference* OR BOTH (these types are sourced from CECADataTypes.xsd).

A building can be uniquely identified by a complex key containing the associated unique property reference and building reference (known as a related building number). The building reference number does not have to be unique within the scope of the entire database. It must however be unique in the context of its relationship with a specific property.
The following property no/building no combinations are allowed:

58106/1
58106/2
33303/1

The following property no/building no combinations are not allowed as they relate to the same property number

58106/1
58106/1

An XML document "author" can therefore specify the building identifier as either a unique OGC property number/OGC building number combination (defined as complexType *OGCBuildingKeyStructure*) OR a unique departmental property number/departmental building number combination (defined as complexType *DeptBuildingKeyStructure*) OR BOTH.
The structure for this is defined in the complexType *BuildingReferenceDetailsStructure***.**

The building structure also contains an unbounded list of associated holdings (element *holdings)* derived from the complexType AssociatedHoldingsStructure (which in turn is derived from the complexType *FGNHoldingReferenceDetailsStructure)*. The structure sepcified for a unique holding reference is detailed in section 4.2.5.1 below.

```
<xsd:element name="BuildingReference" type="BuildingReferenceDetailsStructure"/>

<xsd:complexType name="BuildingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptBuildingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCBReference" type="OGCBuildingReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptBReference" type="DepartmentBuildingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCBuildingRef" type="OGCBuildingKeyStructure"/>
            <xsd:element name="DeptBuildingRef" type="DeptBuildingKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="OGCBuildingKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCBReference" type="OGCBuildingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeptBuildingKeyStructure">
    <xsd:sequence>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        <xsd:element name="DeptBReference" type="DepartmentBuildingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
```

4.2.4.2 BuildingStructure UML and XML Schema

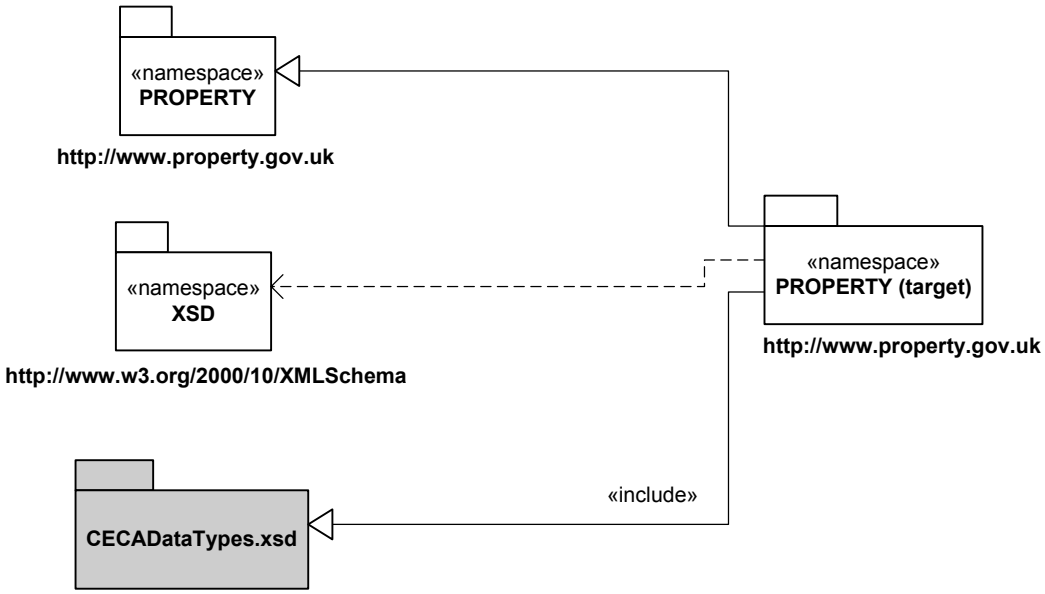


Figure 26: BuildingStructure - namespace

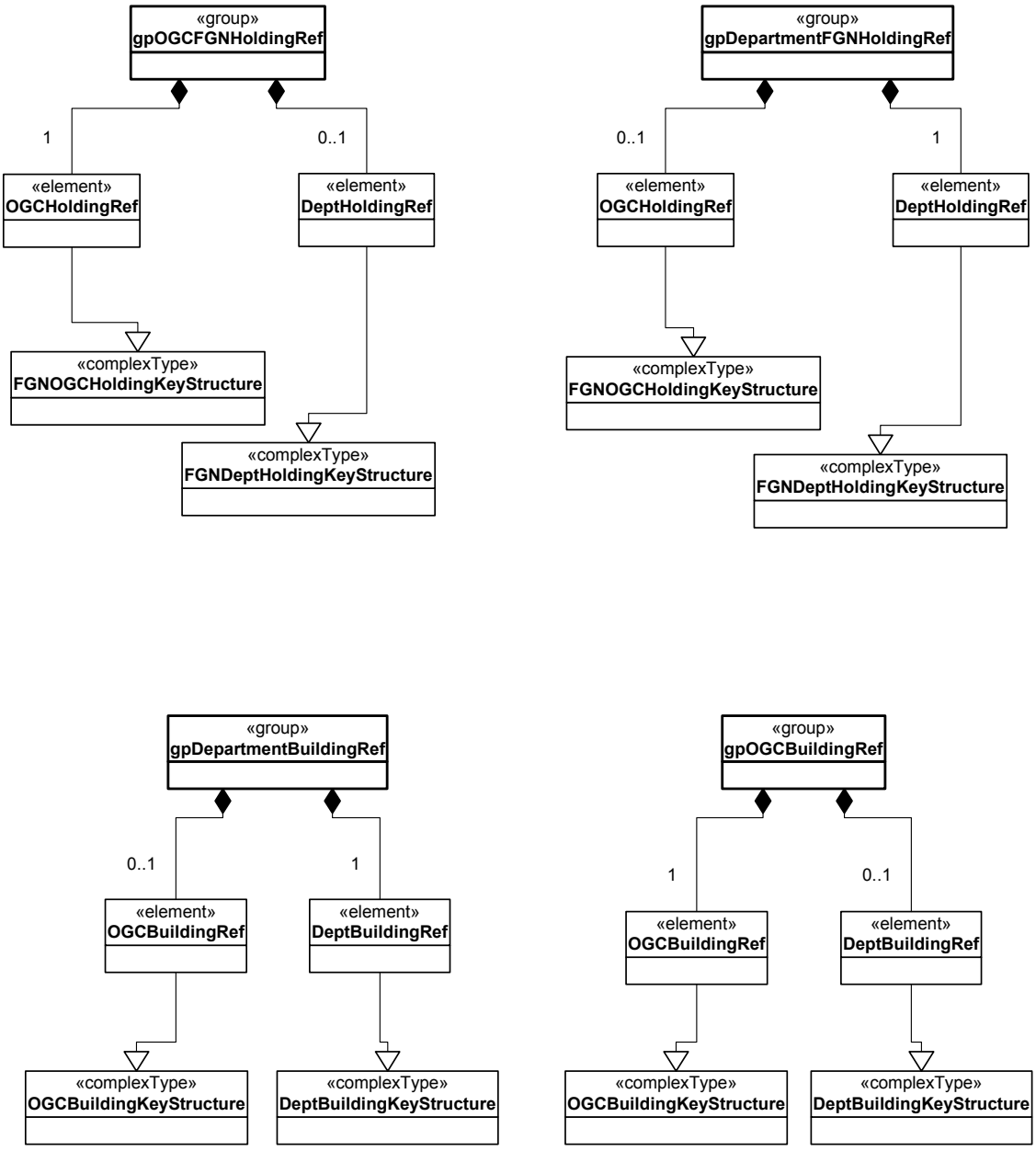


Figure 27: BuildingStructure - groups

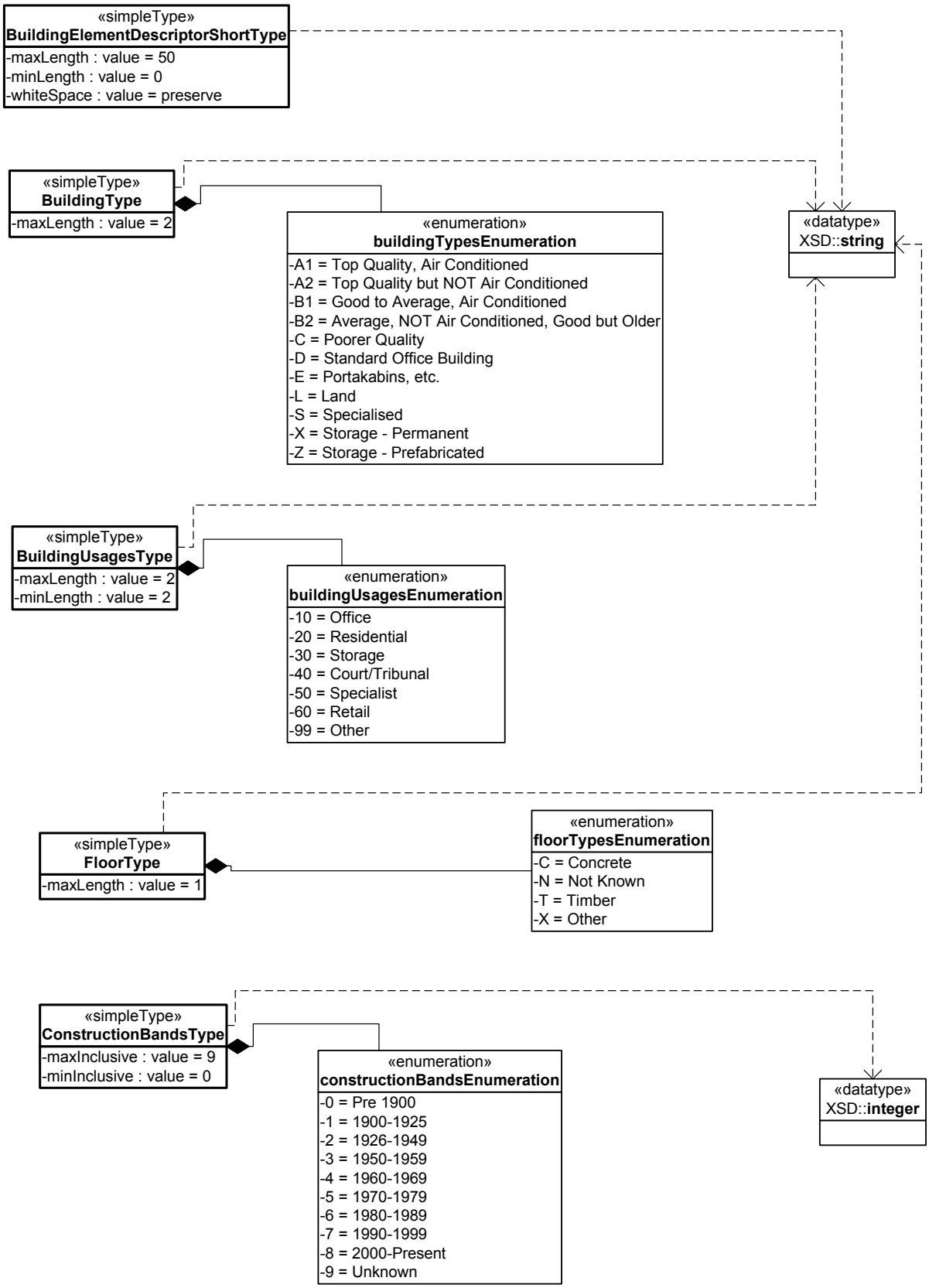


Figure 28: BuildingStructure - simpleTypes 1

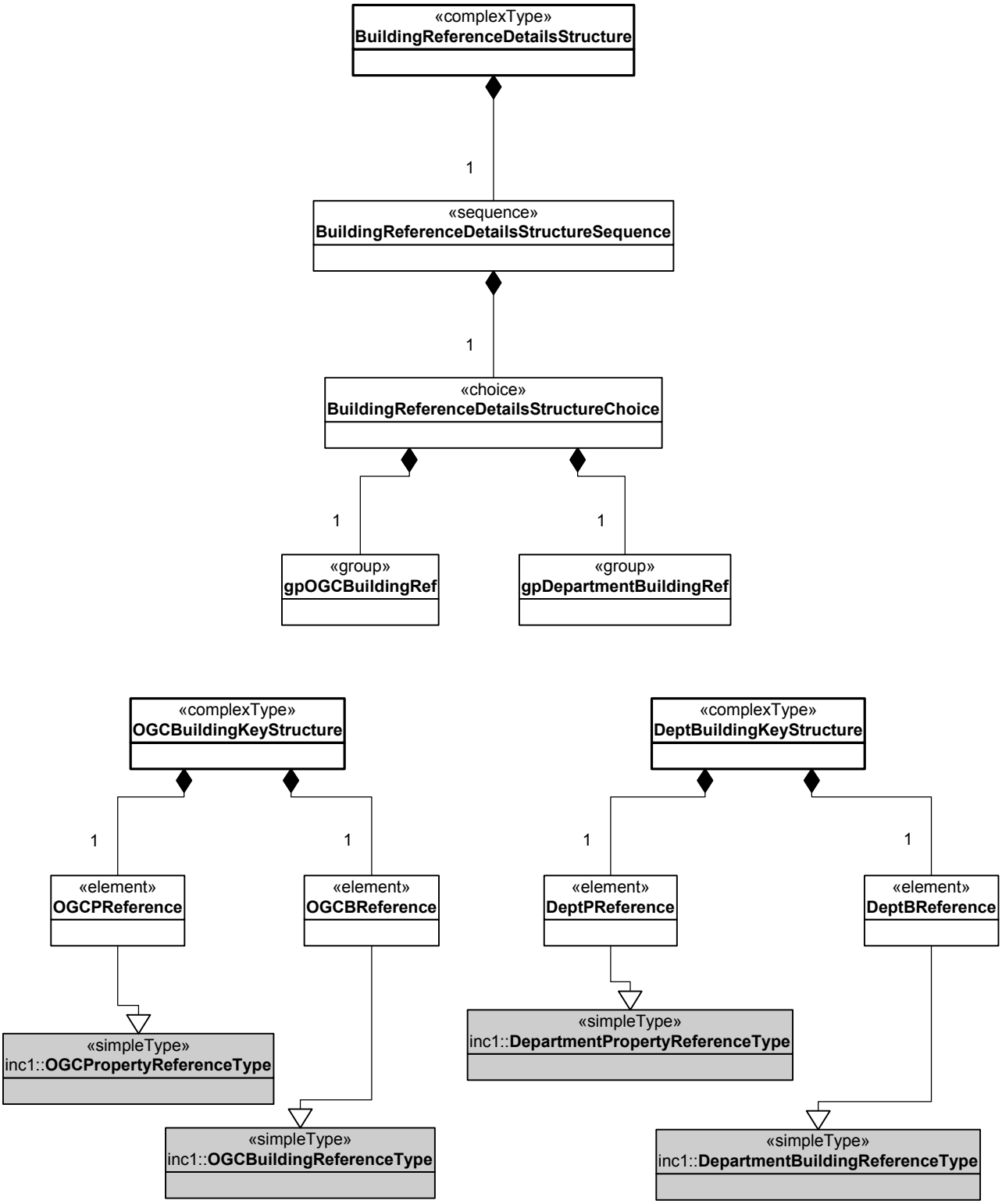


Figure 29: BuildingStructure - complexTypes Building Key

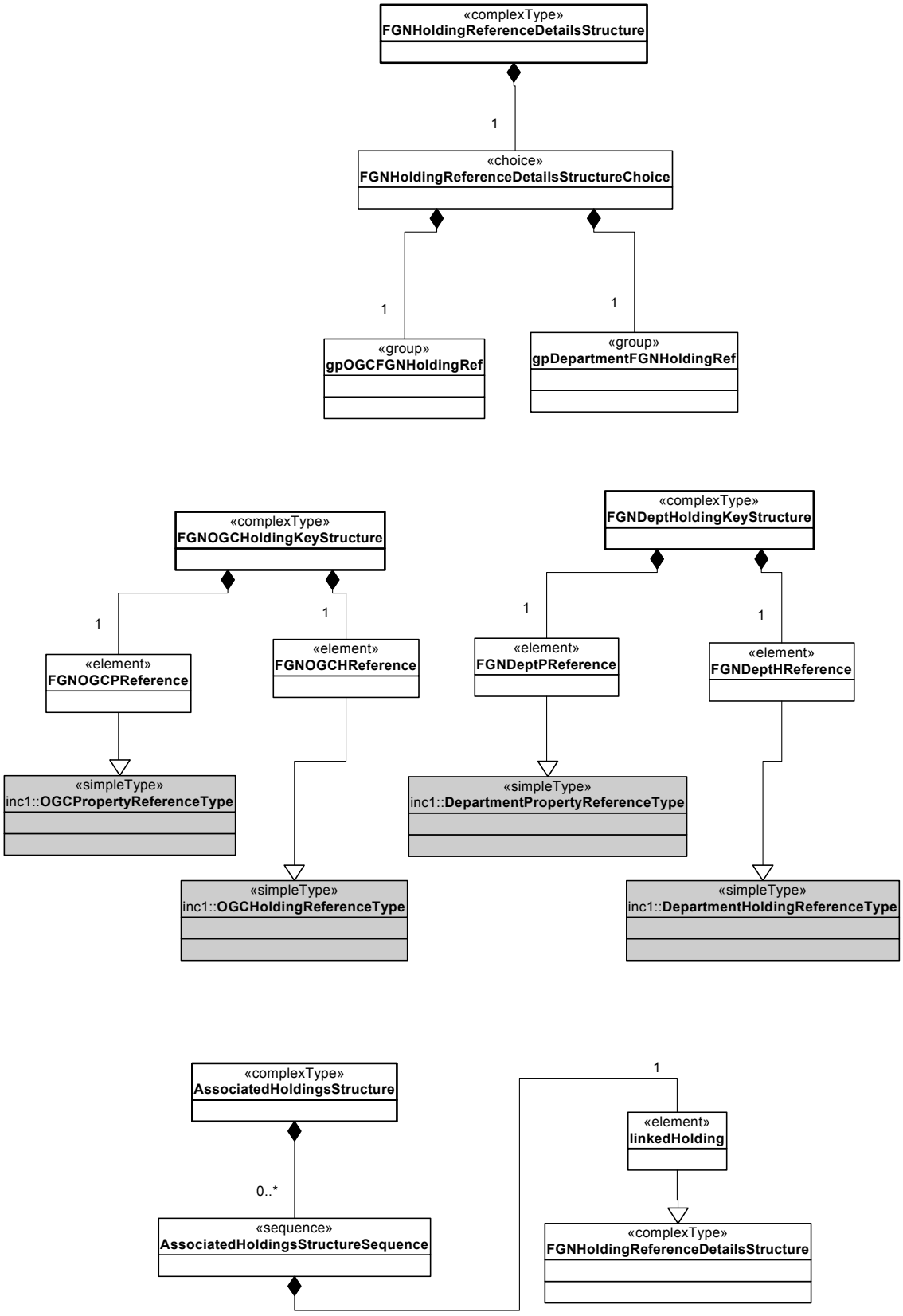


Figure 30: BuildingStructure - complexTypes Associated Holding Key

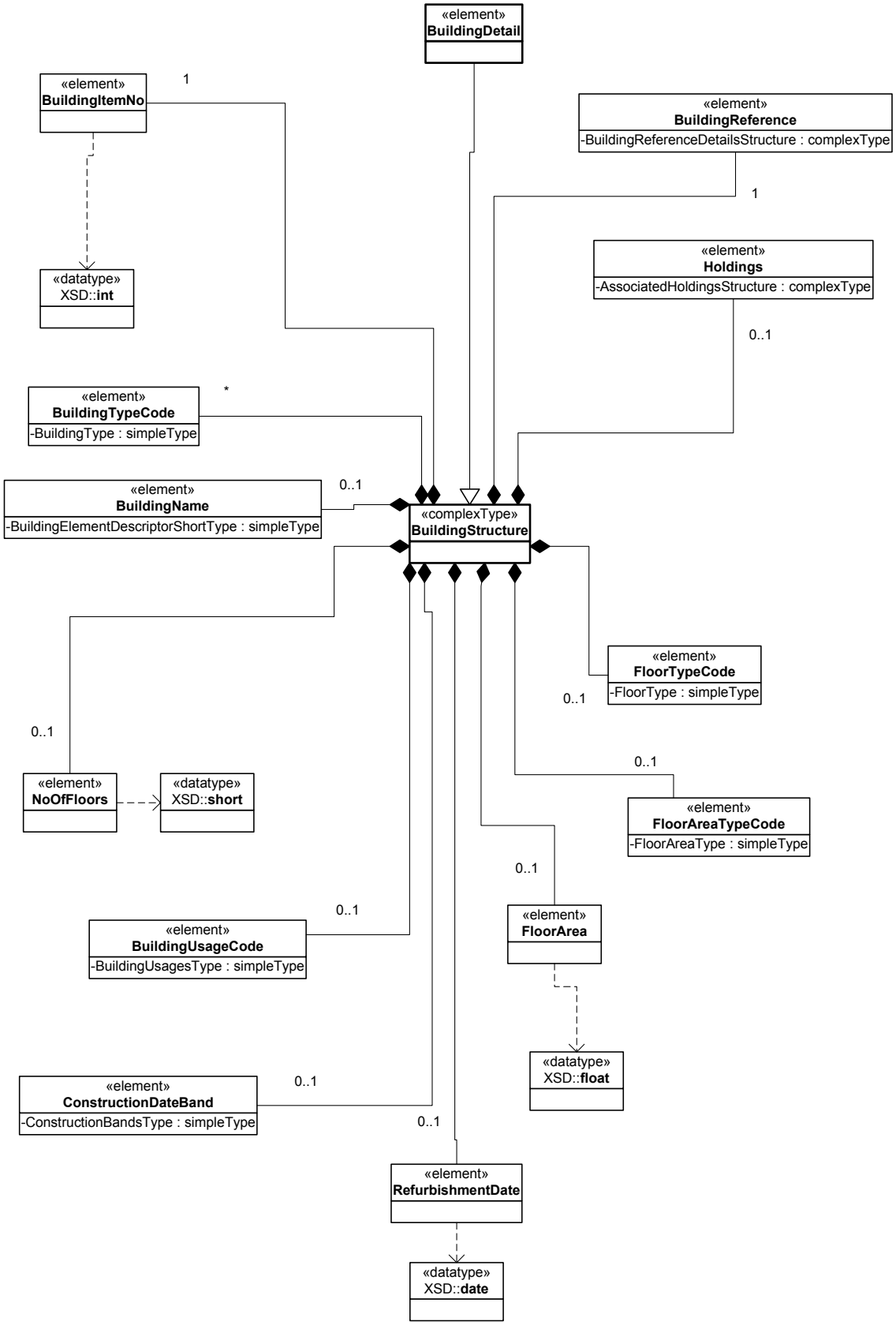


Figure 31: BuildingStructure - BuildingDetail element and BuildingStructure complexType

Figure 32: the XML schema definition for BuildingStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="BuildingStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Property related building reference data

    Purpose: This schema is used to supply the base building data structures for use with building and property
messaging schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core property building details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:simpleType name="BuildingType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        A1:Top Quality, Air Conditioned
                        A2:Top Quality but NOT Air Conditioned
                        B1:Good to Average, Air Conditioned
                        B2:Average, NOT Air Conditioned, Good but Older
                        C:Poorer Quality
                        D:Standard Office Building
                        E:Portakabins, etc.
                        L:Land
                        S:Specialised
                        X:Storage - Permanent
                        Z:Storage - Prefabricated
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="2"/>
            <xsd:enumeration value="A1"/>
            <xsd:enumeration value="A2"/>
            <xsd:enumeration value="B1"/>
            <xsd:enumeration value="B2"/>
            <xsd:enumeration value="C"/>
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="E"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="S"/>
            <xsd:enumeration value="X"/>
            <xsd:enumeration value="Z"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="BuildingUsagesType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        10:Office
```

```
                    20:Residential
                    30:Storage
                    40:Court/Tribunal
                    50:Specialist
                    60:Retail
                    99:Other
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="2"/>
            <xsd:minLength value="2"/>
            <xsd:enumeration value="10"/>
            <xsd:enumeration value="20"/>
            <xsd:enumeration value="30"/>
            <xsd:enumeration value="40"/>
            <xsd:enumeration value="50"/>
            <xsd:enumeration value="60"/>
            <xsd:enumeration value="99"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ConstructionBandsType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    0:Pre 1900
                    1:1900-1925
                    2:1926-1949
                    3:1950-1959
                    4:1960-1969
                    5:1970-1979
                    6:1980-1989
                    7:1990-1999
                    8:2000-Present
                    9:Unknown
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:integer">
            <xsd:maxInclusive value="9"/>
            <xsd:minInclusive value="0"/>
            <xsd:enumeration value="0"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
            <xsd:enumeration value="4"/>
            <xsd:enumeration value="5"/>
            <xsd:enumeration value="6"/>
            <xsd:enumeration value="7"/>
            <xsd:enumeration value="8"/>
            <xsd:enumeration value="9"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="FloorType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    C:Concrete
                    N:Not Known
                    T:Timber
                    X:Other
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="C"/>
            <xsd:enumeration value="N"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="X"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="BuildingElementDescriptorShortType">
        <xsd:restriction base="xsd:string">
```

```
            <xsd:maxLength value="50"/>
            <xsd:minLength value="0"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--The main element feeding off the BuildingStructure Complex Type-->
    <xsd:element name="BuildingDetail" type="BuildingStructure"/>
    <!--The main property schema structure-->
    <xsd:complexType name="BuildingStructure">
        <xsd:sequence>
            <xsd:element name="BuildingReference" type="BuildingReferenceDetailsStructure"/>
            <xsd:element name="Holdings" type="AssociatedHoldingsStructure" minOccurs="0"/>
            <xsd:element name="BuildingTypeCode" type="BuildingType" minOccurs="0"/>
            <xsd:element name="BuildingName" type="BuildingElementDescriptorShortType"/>
            <xsd:element name="NoOfFloors" type="xsd:short" minOccurs="0"/>
            <xsd:element name="BuildingUsageCode" type="BuildingUsagesType" minOccurs="0"/>
            <xsd:element name="ConstructionDateBand" type="ConstructionBandsType" minOccurs="0"/>
            <xsd:element name="RefurbishmentDate" type="xsd:date" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Date of last building refurbishment</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="FloorArea" type="xsd:float" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Total floor area of the building expressed in square
meters</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="FloorAreaTypeCode" type="FloorAreaType" minOccurs="0"/>
            <xsd:element name="FloorTypeCode" type="FloorType" minOccurs="0"/>
            <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="FGNHoldingReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="OGCDeptHoldingRef">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
                            <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
                <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="BuildingReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="OGCDeptBuildingRef">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                            <xsd:element name="OGCBReference" type="OGCBuildingReferenceType"/>
                            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                            <xsd:element name="DeptBReference" type="DepartmentBuildingReferenceType"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="OGCBuildingRef" type="OGCBuildingKeyStructure"/>
                <xsd:element name="DeptBuildingRef" type="DeptBuildingKeyStructure"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="OGCBuildingKeyStructure">
        <xsd:sequence>
            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
            <xsd:element name="OGCBReference" type="OGCBuildingReferenceType"/>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="DeptBuildingKeyStructure">
        <xsd:sequence>
            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
            <xsd:element name="DeptBReference" type="DepartmentBuildingReferenceType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="AssociatedHoldingsStructure">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="linkedHolding" type="FGNHoldingReferenceDetailsStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.5 Holding Structure

This schema supports the architecture required for a CECA property data entity. It provides a complexType *HoldingStructure* that is used by the messaging schemas.

HoldingStructure defines the data elements of a CECA holding object and also defines the object relationships for MainLease, Occupations and Vacant Space as defined in the CECA objectc hierarchy model. These relational elements as described below

Main Lease

The MainLease element is based on the structure implemented in the schema LeaseStructure.

```
<xsd:element name="MainLease" type="LeaseStructure" minOccurs="0"/>
```

Occupations

The Occupations element defines a sequence of zero to many Occupation elements based on the structure implemented in the schema OccupationStructure.

```
<xsd:element name="Occupations" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Occupation" type="OccupationStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Vacant Space

The VacantSpace element defines a sequence of zero to many Vacant elements based on the complexType structure implemented in the schema CECADataTypes.

```
<xsd:element name="VacantSpace" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Vacant" type="VacantSpaceStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

In addition it also supplied a number of simple types which provide an enumerated list of valid values for specific data items. These are listed below.

- **TenureType**: this contains a list of the legal occupancy/tenure
- **HoldingNameType**: a descriptor relating to the reference name given to the holding

4.2.5.1 Unique Identifier

There is a complex Type defined *HoldingReferenceDetails*, this provide the XML document author with the ability to supply unique holding identifiers, which is a combination of a unique property and holding ID, based on the types *OGCPropertyReference/OGCHoldingReference* or *DepartmentPropertyReference/DepartmentHoldingReference* OR BOTH (these types are sourced from CECADataTypes.xsd).

A holding can be uniquely identified by a complex key containing the associated unique property reference and holding reference (known as a related holding number). The holding reference number does not have to be unique within the scope of the entire database. It must however be unique in the context of its relationship with a specific property.
The following property no/holding no combinations are allowed:

58106/1
58106/99999
33303/1

An XML document "author" can therefore specify the holding identifier as either a unique OGC property number/OGC holding number combination (defined as complexType *OGCHoldingKeyStructure*) OR a unique departmental property number/departmental holding number combination (defined as complexType *DeptHoldingKeyStructure*) OR BOTH.
The structure for this is defined in the complexType *HoldingReferenceDetailsStructure.*

```
<xsd:element name="HoldingReference" type="HoldingReferenceDetailsStructure"/>

<xsd:complexType name="HoldingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptHoldingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCHoldingRef" type="OGCHoldingKeyStructure"/>
            <xsd:element name="DeptHoldingRef" type="DeptHoldingKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OGCHoldingKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DeptHoldingKeyStructure">
    <xsd:sequence>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
```

4.2.5.2 HoldingStructure UML and XML Schema

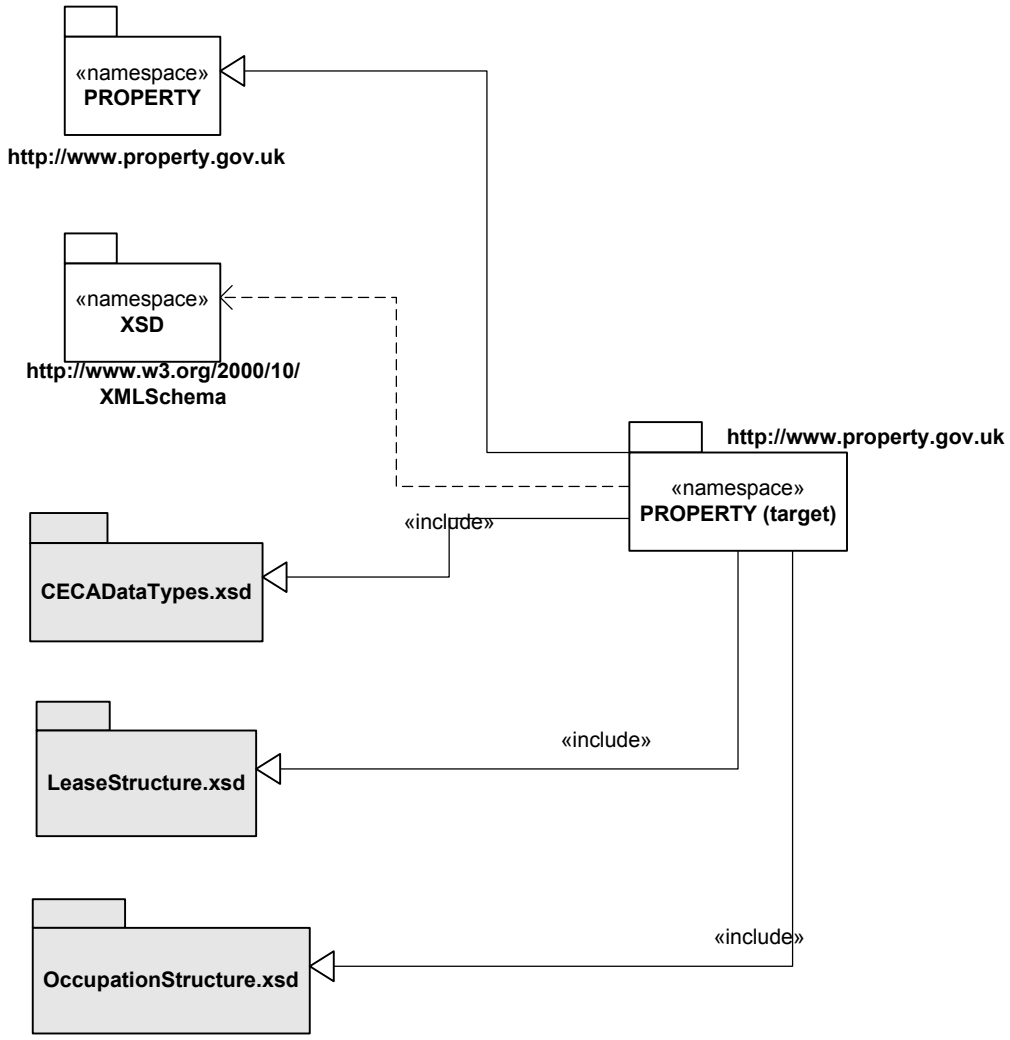


Figure 33: HoldingStructure - Namespace

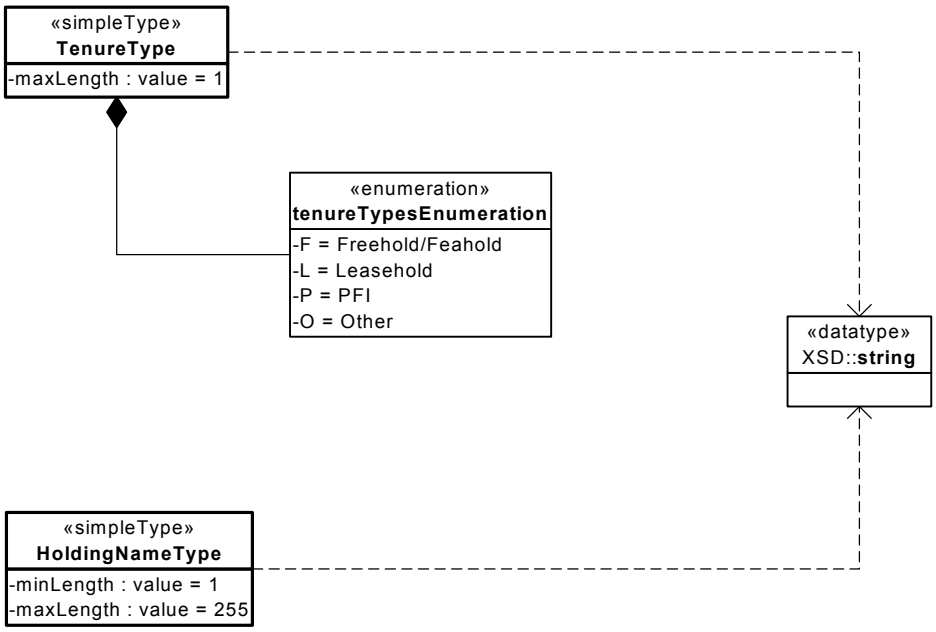


Figure 34: HoldingStructure - SimpleTypes

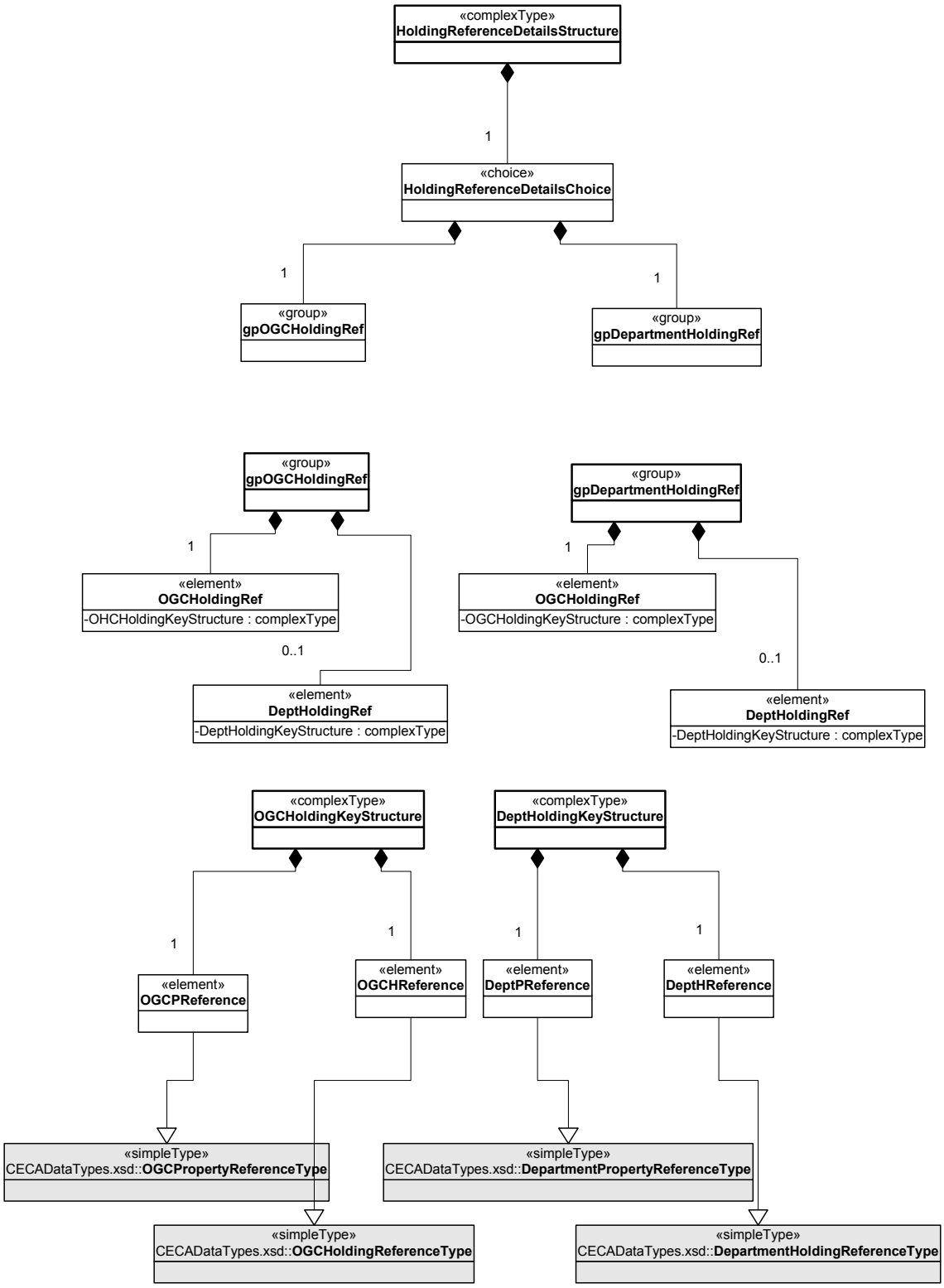


Figure 35: HoldingStructure - complexTypes and groups

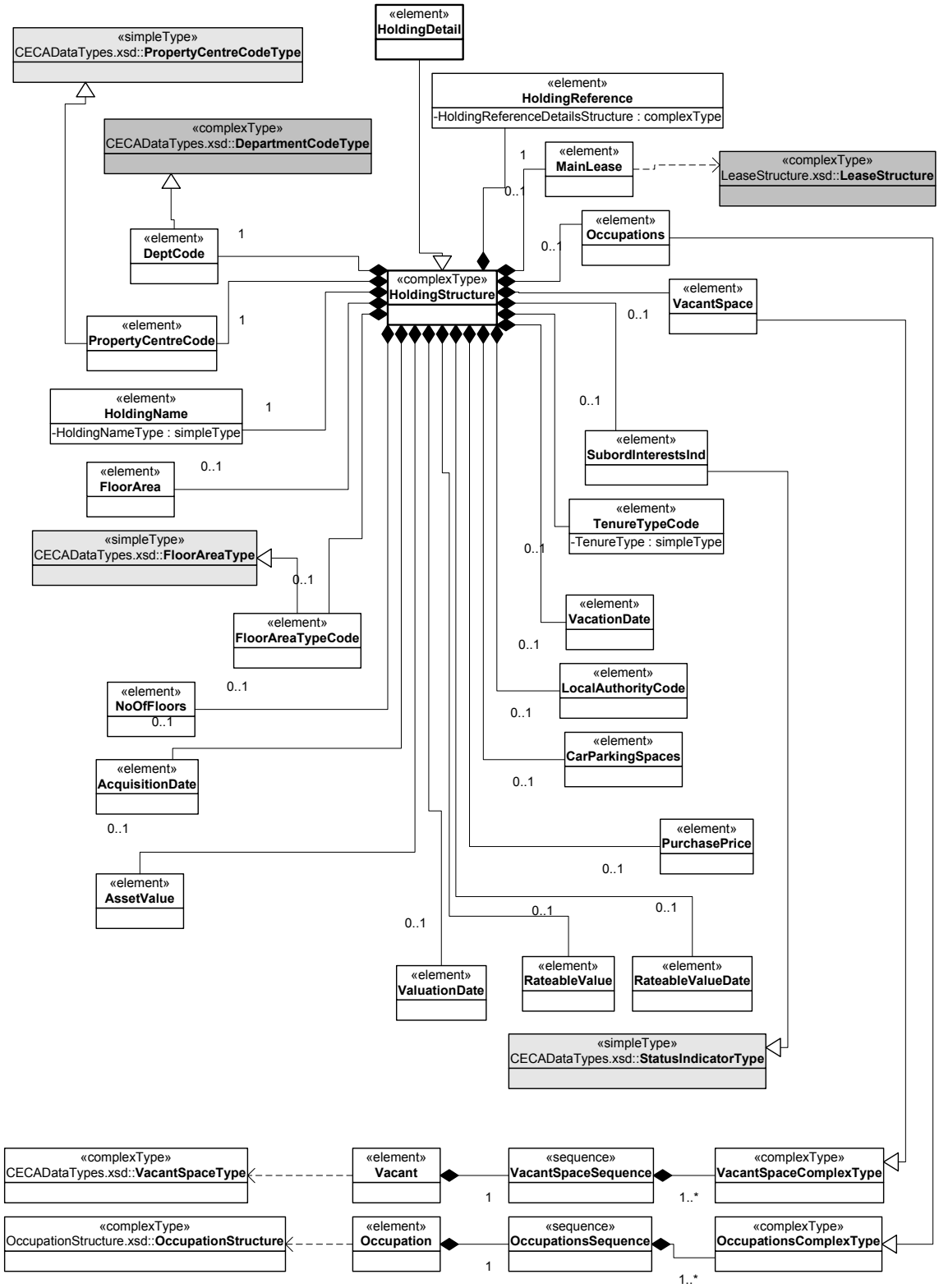


Figure 36: HoldingStructure – HoldingDetail element and HoldingStructure complexType

Figure 37: the XML schema definition for HoldingStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="HoldingStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Property related holding reference data

    Purpose: This schema is used to supply the base holding data structures for use with holding  messaging schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core property building details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/LeaseStructure.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/OccupationStructure.xsd"/>
    <xsd:simpleType name="TenureType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        F:Freehold/Feuhold
                        L:Leasehold
                        P:PFI
                        O:Other
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="F"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="P"/>
            <xsd:enumeration value="O"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="HoldingNameType">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="255"/>
            <xsd:minLength value="1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--The main element feeding off the HoldingStructure Complex Type-->
    <xsd:element name="HoldingDetail" type="HoldingStructure"/>
    <!--The main holding schema structure-->
    <xsd:complexType name="HoldingStructure">
        <xsd:sequence>
            <xsd:element name="HoldingReference" type="HoldingReferenceDetailsStructure"/>
            <xsd:element name="DeptCode" type="PropertyCentreCodeType">
                <xsd:annotation>
                    <xsd:documentation>The OGC reference to the owning department</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="PropertyCentreCode" type="PropertyCentreCodeType">
```

```
            <xsd:annotation>
                <xsd:documentation>The OGC reference to the owning property centre</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="HoldingName" type="HoldingNameType">
            <xsd:annotation>
                <xsd:documentation>A description of the holding</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="FloorArea" type="xsd:float" minOccurs="0"/>
        <xsd:element name="FloorAreaTypeCode" type="FloorAreaType" minOccurs="0"/>
        <xsd:element name="NoOfFloors" type="xsd:short" minOccurs="0"/>
        <xsd:element name="AcquisitionDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="AssetValue" type="xsd:float" minOccurs="0"/>
        <xsd:element name="ValuationDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="RateableValue" type="xsd:float" minOccurs="0"/>
        <xsd:element name="RateableValueDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="PurchasePrice" type="xsd:float" minOccurs="0"/>
        <xsd:element name="CarParkingSpaces" type="xsd:short" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>OGC internal data to be decided</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="LocalAuthorityCode" type="xsd:short" minOccurs="0"/>
        <xsd:element name="VacationDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="TenureTypeCode" type="TenureType"/>
        <xsd:element name="SubordInterestsInd" type="StatusIndicatorType" minOccurs="0"/>
        <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
        <xsd:element name="MainLease" type="LeaseStructure" minOccurs="0"/>
        <xsd:element name="Occupations" minOccurs="0">
            <xsd:complexType mixed="true">
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="Occupation" type="OccupationStructure"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="VacantSpace" minOccurs="0">
            <xsd:complexType mixed="true">
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="Vacant" type="VacantSpaceStructure"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="HoldingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptHoldingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCHoldingRef" type="OGCHoldingKeyStructure"/>
            <xsd:element name="DeptHoldingRef" type="DeptHoldingKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="OGCHoldingKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DeptHoldingKeyStructure">
    <xsd:sequence>
```

```
            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
            <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.6 Lease Structure

This schema supports the architecture required for a CECA property data entity. It provides a complexType *LeaseStructure* that is used by the messaging schemas.

LeaseStructure defines the data elements of a CECA lease object and also defines the object relationships for Landlord and Break/Review as defined in the CECA object hierarchy model. These relational elements as described below

Landlord

The Landlord element is based on the structure implemented in the schema LandlordStructure.

```
<xsd:element name="Landlord" type="LandlordStructure" minOccurs="0"/>
```

Break/Review

The BreakReviews element defines a sequence of zero to many BreakReview elements based on the type BreakReviewStructure.

```
<xsd:element name="BreakReviews" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="BreakReview" type="BreakReviewStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

In addition it also supplied a number of simple types which provide an enumerated list of valid values for specific data items. These are listed below.

- **InsuranceLiabilityType**: this contains a list of all valid type of insurance liability
- **RentType**: valid types of rent payable
- **RentReviewType**: valid methods of rent review negotiation methods
- **RentPaymentPeriodType**: the payment cycle duration for a lease rent
- **RentPaymentTermsType**: the terms of the lease regarding the time of rent payment
- **BreakReviewIndType**: the type of the BreakReview element
- **BreakOptPartyType**: the types of Break Options available.

4.2.6.1 Unique Identifier

There is a complex Type defined Lease*ReferenceDetails*, this provide the XML document author with the ability to supply unique lease identifiers. This is a combination of a unique property, holding, lease and optionally occupation ID (the occupation ID is optional as it is not required for a main lease associated with a holding), based on the types *OGCPropertyReference/OGCHoldingReference/ OGCLeaseReference/ OGCOccupationReference* or *DepartmentPropertyReference/DepartmentHoldingReference/ DepartmentLeaseReference/ DepartmentOccupationReference* OR BOTH (these types are sourced from CECADataTypes.xsd).

A lease can be uniquely identified by a complex key containing the associated unique property reference, related holding and optionally an occupation reference (known as a

related lease number). The occupation number is optional in the following circumstances. Where the lease is a main lease associated directly with a holding then the occupation number is not set. When the lease is an occupation based lease the occupation number must be specified. The following are examples of valid key combinations

Main Lease (propertyno/holding no)

58106/1
58106/2
33303/1

Occupation Based Lease (property no/holding no/occupation no)

58106/1/1
58102/1/2
58106/2/1
33303/1/1

NOTE: these rules are strict. Failure to specify an occupation number will identify the lease as a main lease. Inclusion of an occupation number will always identify the lease as occupation based. Within Epims there are specific rules related to the handling of each lease type.

In addition the key structure defined for lease also allows the inclusion of a unique lease ID. This ID will be unique to the entire database. This element is optional because a lease can be idetified using the element listed above and it cannot be assumed that all property management systems can support an identifier which is unique (and also meaningless in the context of the data it relates to).

The overall structure of the lease key can therefore be summarised as

Property number – mandatory
Holding number – mandatory
Occupation Number – (not available for main lease and mandatory for occupation based lease)

An XML document "author" can therefore specify the lease identifier as either a unique OGC property number/OGC holding number/OGC lease ID (optional)/ OGC Occupation number (optional) combination (defined as complexType *OGCLeaseKeyStructure*) OR a unique departmental property number/departmental holding number/ departmental lease ID (optional)/ departmental occupation number (optional) combination (defined as complexType *DeptOccupationKeyStructure*) OR BOTH.
The structure for this is defined in the complexType *LeaseReferenceDetailsStructure.*

```
<xsd:element name="LeaseReference" type="LeaseReferenceDetailsStructure"/>

<xsd:complexType name="LeaseReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptLeaseRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"
minOccurs="0"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
```

```
                        <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"
minOccurs="0"/>
                    </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="OGCLeaseRef" type="OGCLeaseKeyStructure"/>
                <xsd:element name="DeptLeaseRef" type="DeptLeaseKeyStructure"/>
            </xsd:choice>
        </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OGCLeaseKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
        <xsd:element name="OGCOReference" type="OGCOccupationReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DeptLeaseKeyStructure">
    <xsd:sequence>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
        <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

4.2.6.2 LeaseStructure UML and XML Schema

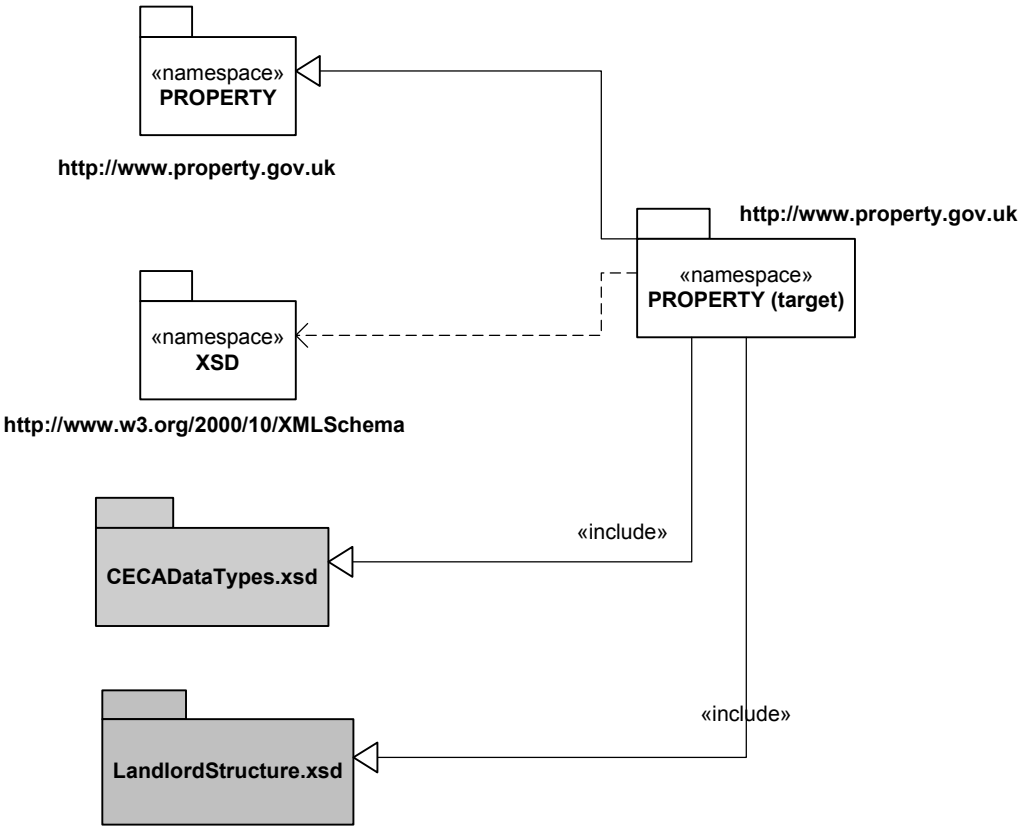


Figure 38: LeaseStructure – namespace

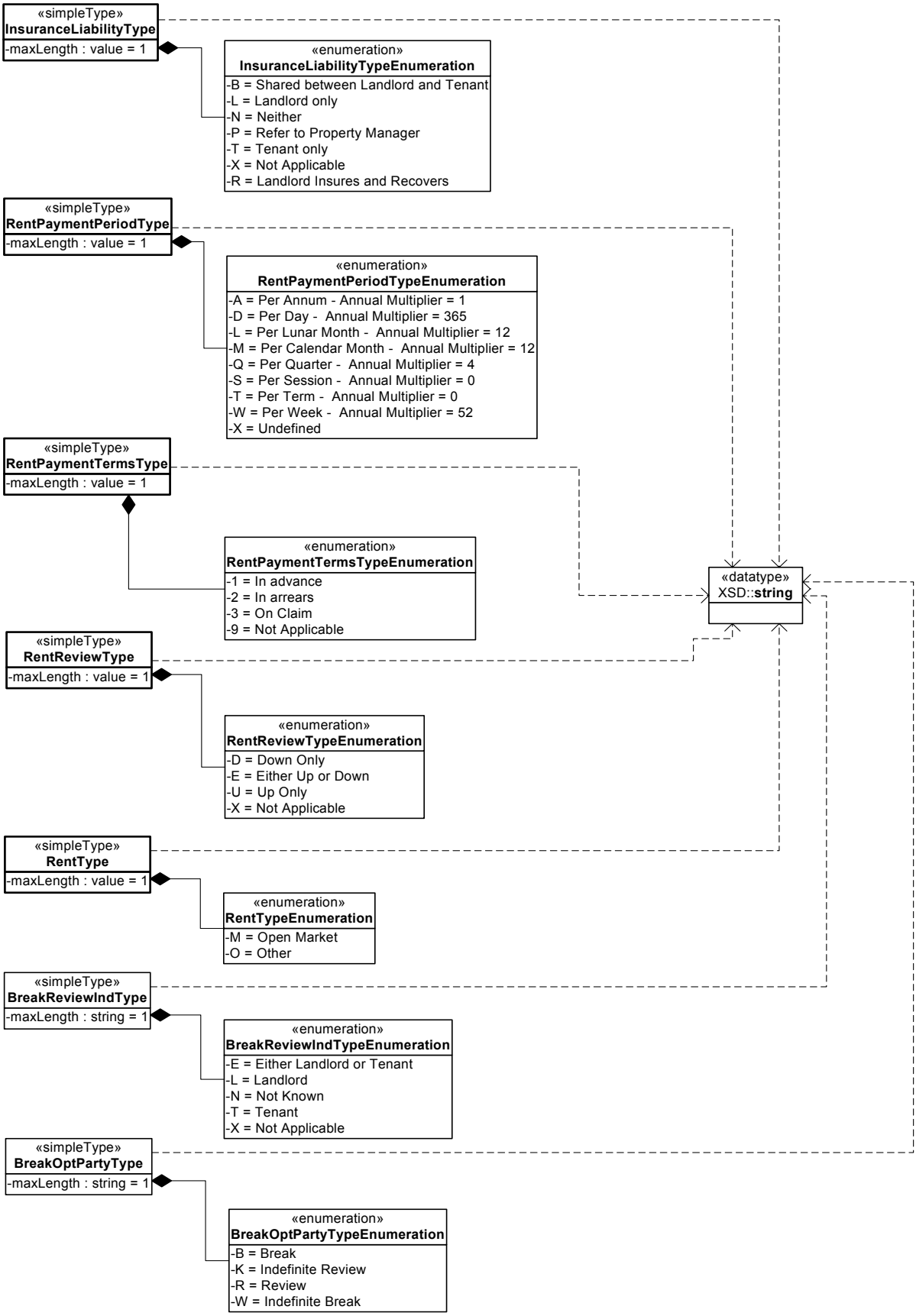
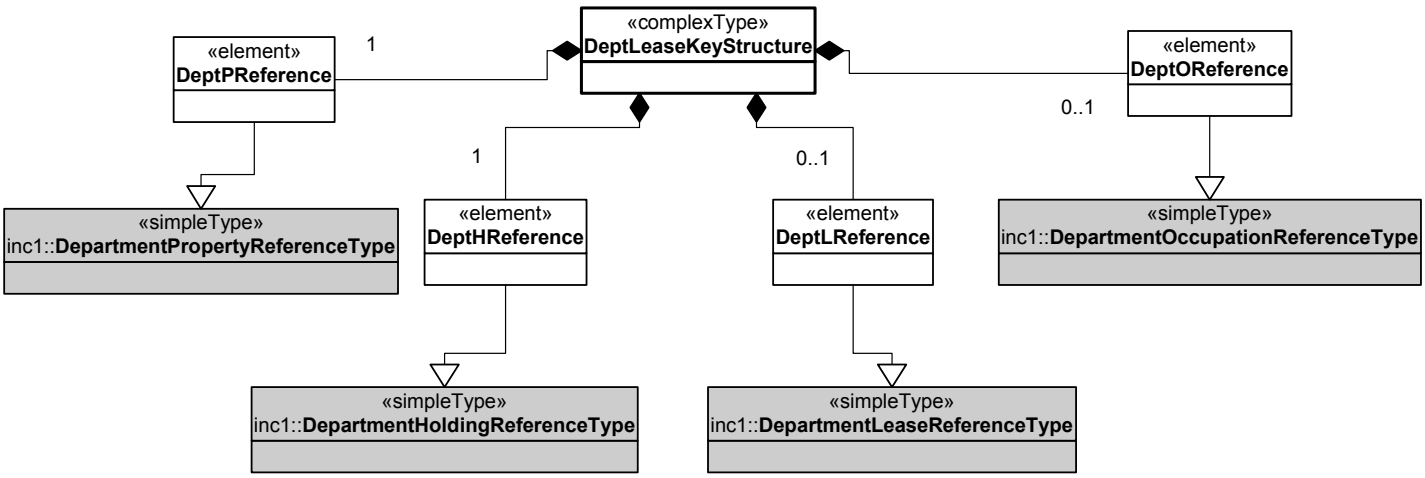
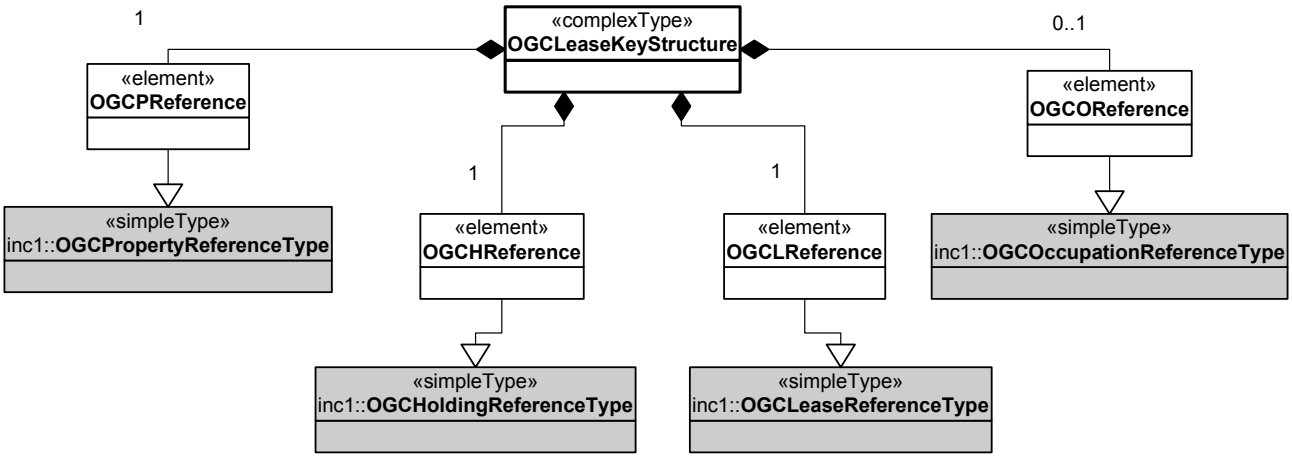


Figure 39: LeaseStructure - simpleTypes





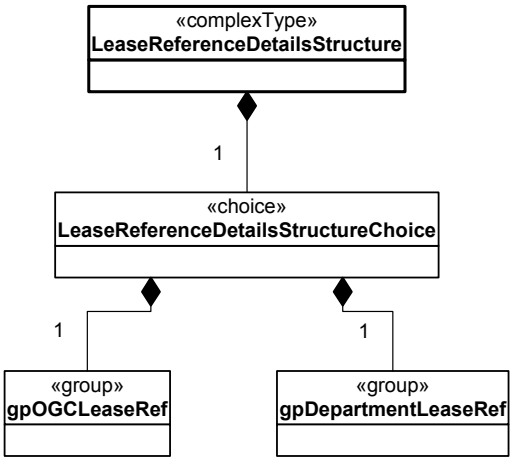


Figure 40: LeaseStructure – complexTypes

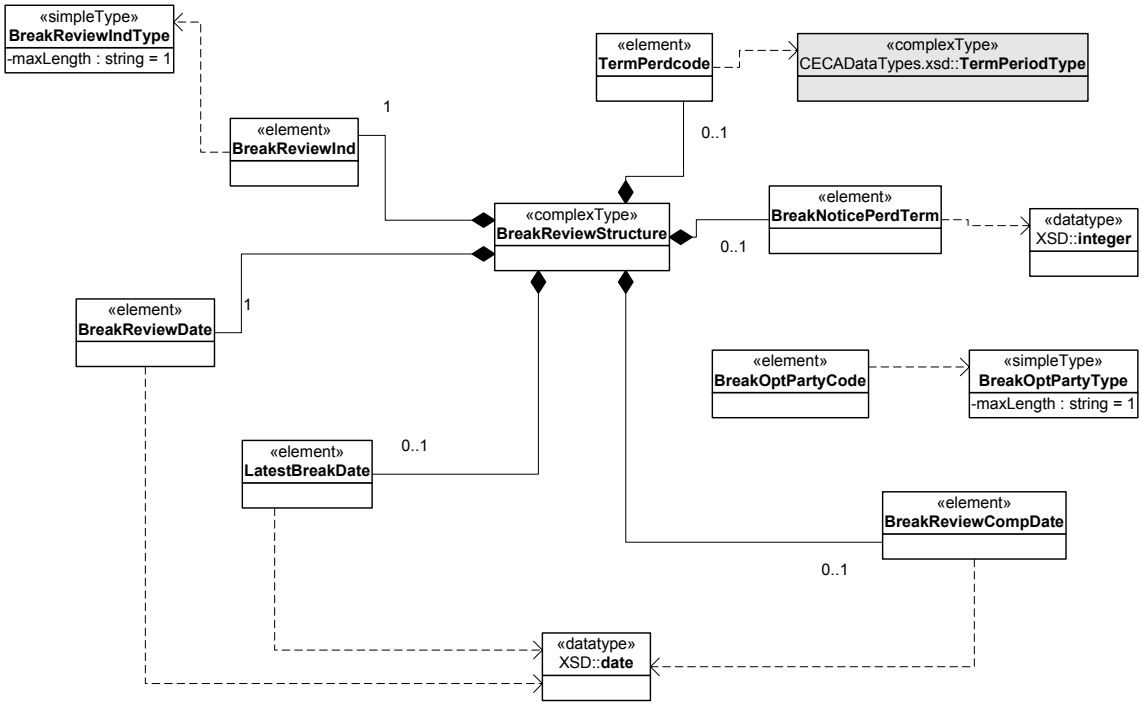


Figure 41: LeaseStructure – BreakReview ComplexType

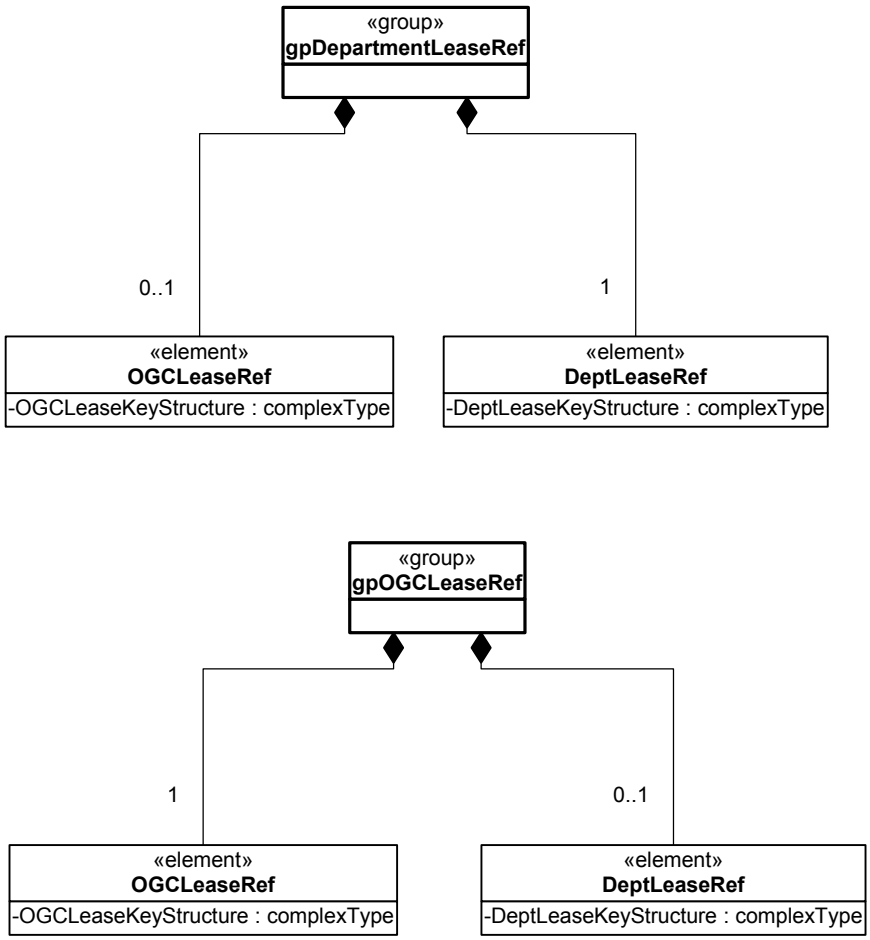


Figure 42: LeaseStructure - groups

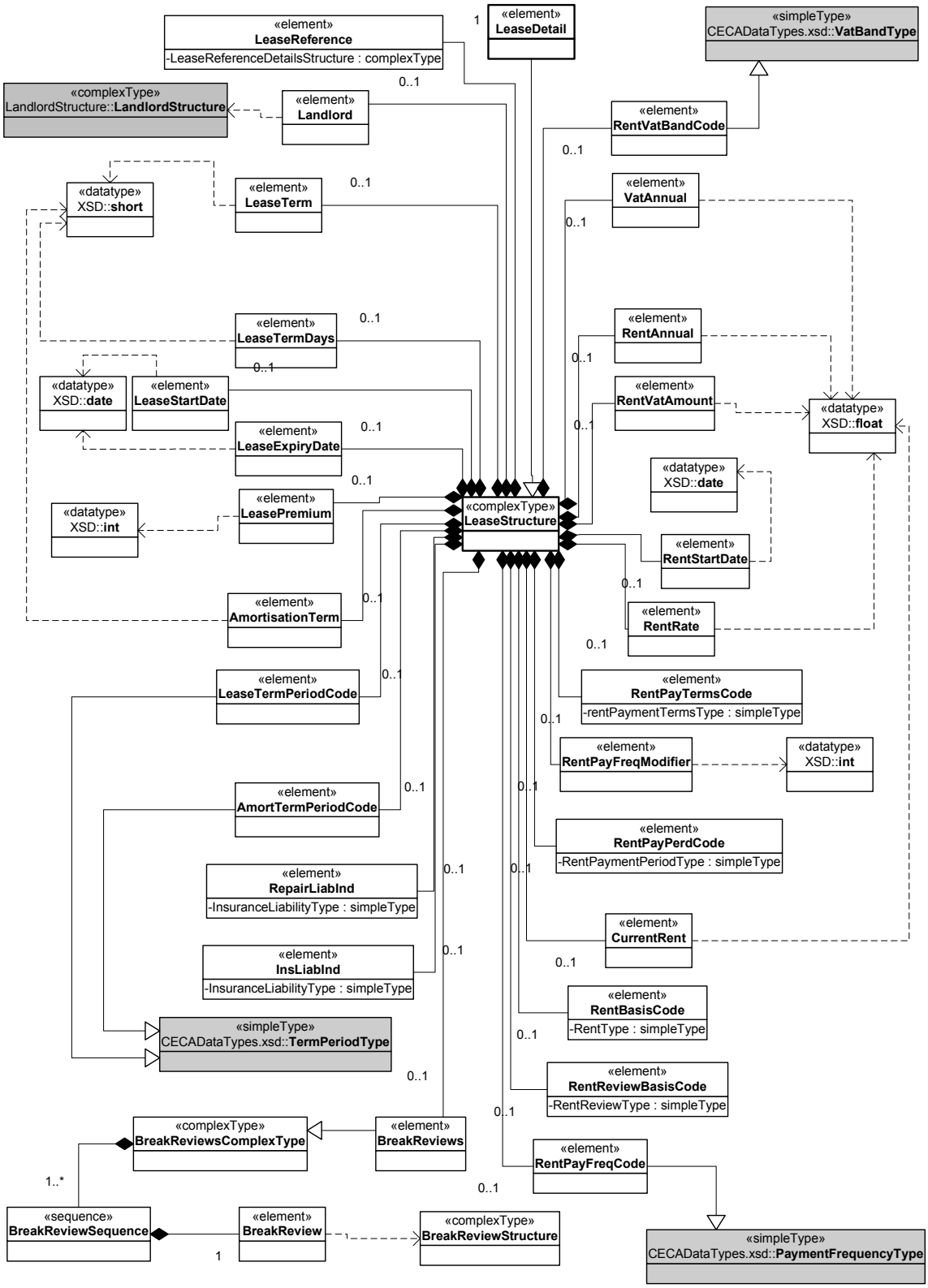


Figure 43: LeaseStructure – LeaseDetail element and LeaseStructure complexType

Figure 44: the XML schema definition for LeaseStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.property.gov.uk" elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1"
id="LeaseStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Property related lease reference data

    Purpose: This schema is used to supply the base lease data structures for use with lease  messaging schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core property lease details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/LandlordStructure.xsd"/>
    <xsd:simpleType name="InsuranceLiabilityType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        B: Shared between Landlord and Tenant
                        L: Landlord only
                        N: Neither
                        P: Refer to Property Manager
                        T: Tenant only
                        X: Not Applicable
                        R: Landlord Insures and Recovers
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="B"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="N"/>
            <xsd:enumeration value="P"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="X"/>
            <xsd:enumeration value="R"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="RentType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        M: Open Market
                        O: Other
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="O"/>
        </xsd:restriction>
```

```
    </xsd:simpleType>
    <xsd:simpleType name="RentReviewType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        D: Down Only
                        E: Either Up or Down
                        U: Up Only
                        X: Not Applicable
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="E"/>
            <xsd:enumeration value="U"/>
            <xsd:enumeration value="X"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="RentPaymentPeriodType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        A: Per Annum - Annual Multiplier = 1
                        D: Per Day - Annual Multiplier = 365
                        L: Per Lunar Month - Annual Multiplier = 12
                        M: Per Calendar Month - Annual Multiplier = 12
                        Q: Per Quarter (English, New English, Scottish, New Scottish, Northern Ireland/Gale, Crown
Estate, etc., as appropriate) - Annual Multiplier = 4
                        S: Per Session - Annual Multiplier = 0
                        T: Per Term - Annual Multiplier = 0
                        W: Per Week - Annual Multiplier = 52
                        X: Undefined
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="A"/>
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="Q"/>
            <xsd:enumeration value="S"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="W"/>
            <xsd:enumeration value="X"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="RentPaymentTermsType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        1: In advance
                        2: In arrears
                        3: On Claim
                        9: Not Applicable
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
            <xsd:enumeration value="9"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="BreakReviewIndType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        B    Break
```

```
                    K   Indefinite Review
                    R   Review
                    W   Indefinite Break
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="B"/>
            <xsd:enumeration value="K"/>
            <xsd:enumeration value="R"/>
            <xsd:enumeration value="W"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="BreakOptPartyType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                    E   Either Landlord or Tenant
                    L   Landlord
                    N   Not Known
                    T   Tenant
                    X   Not Applicable
                </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="E"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="N"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="X"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--The main element feeding off the LeaseStructure Complex Type-->
    <xsd:element name="LeaseDetail" type="LeaseStructure"/>
    <!--The main lease schema structure-->
    <xsd:complexType name="LeaseStructure">
        <xsd:sequence>
            <xsd:element name="LeaseReference" type="LeaseReferenceDetailsStructure"/>
            <xsd:element name="Landlord" type="LandlordStructure" minOccurs="0"/>
            <xsd:element name="LeaseTerm" type="xsd:short" minOccurs="0"/>
            <xsd:element name="LeaseTermPeriodCode" type="TermPeriodType" minOccurs="0"/>
            <xsd:element name="LeaseTermDays" type="xsd:short" minOccurs="0"/>
            <xsd:element name="LeaseStartDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="LeaseExpiryDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="LeasePremium" type="xsd:int" minOccurs="0"/>
            <xsd:element name="AmortisationTerm" type="xsd:short" minOccurs="0"/>
            <xsd:element name="AmortTermPeriodCode" type="TermPeriodType" minOccurs="0"/>
            <xsd:element name="RepairLiabInd" type="InsuranceLiabilityType" minOccurs="0"/>
            <xsd:element name="InsLiabInd" type="InsuranceLiabilityType" minOccurs="0"/>
            <xsd:element name="RentReviewBasisCode" type="RentReviewType" minOccurs="0"/>
            <xsd:element name="RentBasisCode" type="RentType" minOccurs="0"/>
            <xsd:element name="CurrentRent" type="xsd:float" minOccurs="0"/>
            <xsd:element name="RentPayPerdCode" type="RentPaymentPeriodType" minOccurs="0"/>
            <xsd:element name="RentPayFreqCode" type="PaymentFrequencyType" minOccurs="0"/>
            <xsd:element name="RentPayFreqModifier" type="xsd:int" minOccurs="0"/>
            <xsd:element name="RentPayTermsCode" type="RentPaymentTermsType" minOccurs="0"/>
            <xsd:element name="RentRate" type="xsd:float" minOccurs="0"/>
            <xsd:element name="RentStartDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="RentVatBandCode" type="VatBandType" minOccurs="0"/>
            <xsd:element name="RentVatAmount" type="xsd:float" minOccurs="0"/>
            <xsd:element name="RentAnnual" type="xsd:float" minOccurs="0"/>
            <xsd:element name="VatAnnual" type="xsd:float" minOccurs="0"/>
            <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
            <xsd:element name="BreakReviews" minOccurs="0">
                <xsd:complexType mixed="true">
                    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                        <xsd:element name="BreakReview" type="BreakReviewStructure"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="LeaseReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="OGCDeptLeaseRef">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                            <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                            <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"
minOccurs="0"/>
                            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                            <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                            <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"
minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="OGCLeaseRef" type="OGCLeaseKeyStructure"/>
                <xsd:element name="DeptLeaseRef" type="DeptLeaseKeyStructure"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="OGCLeaseKeyStructure">
        <xsd:sequence>
            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
            <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
            <xsd:element name="OGCOReference" type="OGCOccupationReferenceType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="DeptLeaseKeyStructure">
        <xsd:sequence>
            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
            <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
            <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="BreakReviewStructure">
        <xsd:sequence>
            <xsd:element name="BreakReviewInd" type="BreakReviewIndType"/>
            <xsd:element name="BreakReviewDate" type="CECADateFormatType"/>
            <xsd:element name="LatestBreakDate" type="CECADateFormatType" minOccurs="0"/>
            <xsd:element name="BreakNoticePerdTerm" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="TermPerdCode" type="TermPeriodType" minOccurs="0"/>
            <xsd:element name="BreakReviewCompDate" type="CECADateFormatType" minOccurs="0"/>
            <xsd:element name="BreakOptPartyCode" type="BreakOptPartyType" minOccurs="0"/>
            <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.7 Occupation Structure

This schema supports the architecture required for a CECA property data entity. It provides a complexType *OccupationStructure* that is used by the messaging schemas.

OccupationStructure defines the data elements of a CECA occupation object and also defines the object relationships for Tenant and Vacant Space as defined in the CECA object hierarchy model. These relational elements as described below

Tenant

The Tenant element is based on the structure implemented in the schema TenantStructure.

```
<xsd:element name="Tenant" type="TenantStructure" minOccurs="0"/>
```

VacantSpace

The VacantSpace element defines a sequence of zero to many Vacant elements based on the complexType structure implemented in the schema CECADataTypes.

```
<xsd:element name="VacantSpace" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="Vacant" type="VacantSpaceStructure"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

In addition it also supplied a number of simple types which provide an enumerated list of valid values for specific data items. These are listed below.

- **OccupationType**: this contains a list of all valid types of occupation.

4.2.7.1 Unique Identifier

There is a complex Type defined *OccupationReferenceDetails*, this provide the XML document author with the ability to supply unique occupation identifiers. This is a combination of a unique property, holding, and occupation ID, based on the types *OGCPropertyReference/OGCHoldingReference/OGCOccupationReference* or *DepartmentPropertyReference/DepartmentHoldingReference/ DepartmentOccupationReference* OR BOTH (these types are sourced from CECADataTypes.xsd).

An occupation can be uniquely identified by a complex key containing the associated unique property reference, related holding and occupation reference (known as a related occupation number). The occupation reference number does not have to be unique within the scope of the entire database. It must however be unique in the context of its relationship with a specific holding.
The following property no/holding no/occupation no combinations are allowed:

58106/1/1
58106/99999/1
33303/1/1
33303/1/2

An XML document "author" can therefore specify the occupation identifier as either a unique OGC property number/OGC holding number/ OGC Occupation number combination (defined as complexType *OGCOccupationKeyStructure*) OR a unique departmental property number/departmental holding number/ departmental occupation number combination (defined as complexType *DeptOccupationKeyStructure*) OR BOTH. The structure for this is defined in the complexType *OccupationReferenceDetailsStructure.*

```
<xsd:element name="OccupationReference" type="OccupationReferenceDetailsStructure"/>

<xsd:complexType name="OccupationReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptOccupationRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                        <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCOccupationRef" type="OGCOccupationKeyStructure"/>
            <xsd:element name="DeptOccupationRef" type="DeptOccupationKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OGCOccupationKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
        <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DeptOccupationKeyStructure">
    <xsd:sequence>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
        <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
```

4.2.7.2 OccupationStructure UML and XML Schema

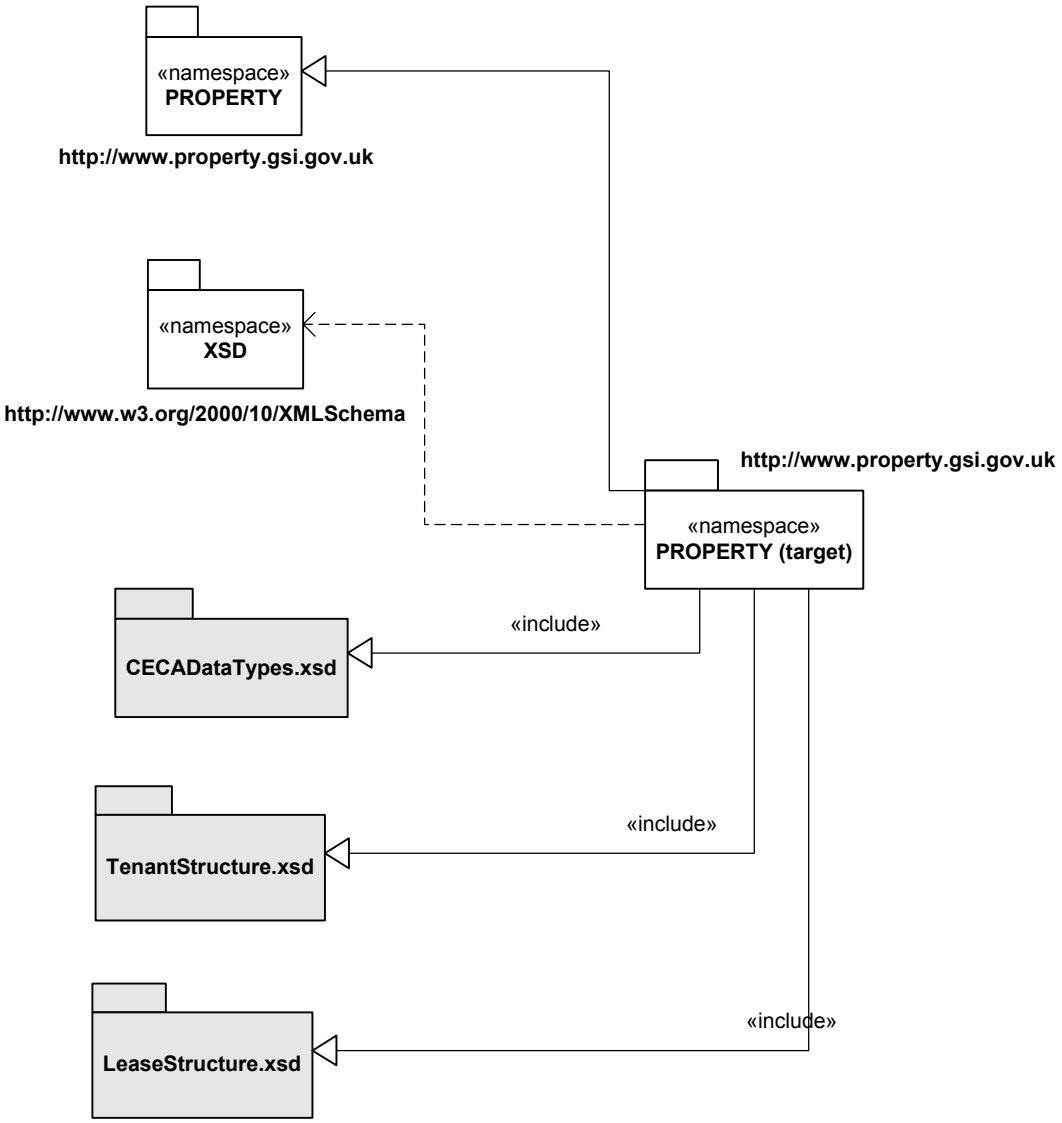


Figure 45: OccupationStructure – namespace

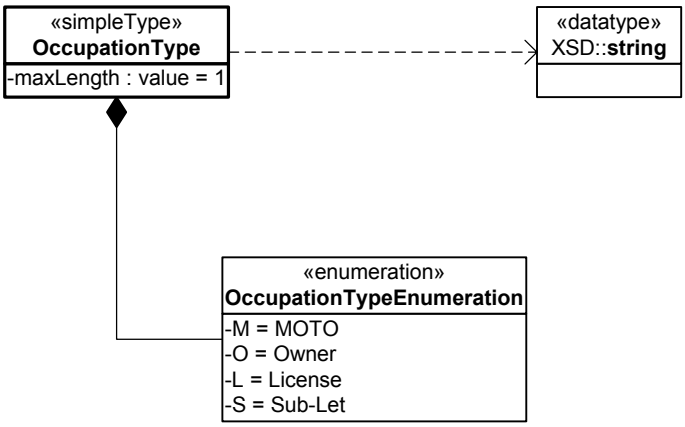


Figure 46: OccupationStructure OccupationType simpleType

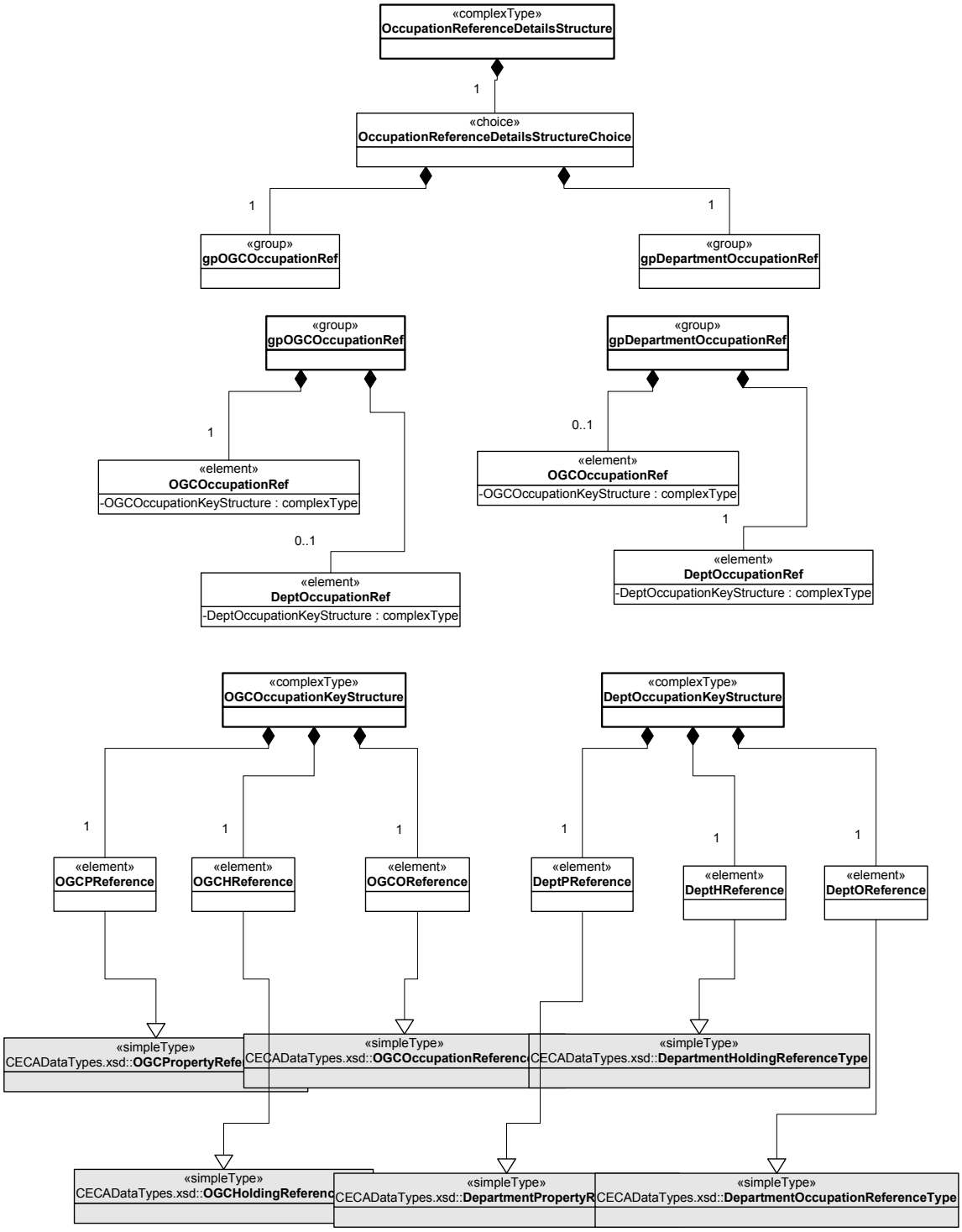


Figure 47: OccupationStructure – groups and complexTypes 1

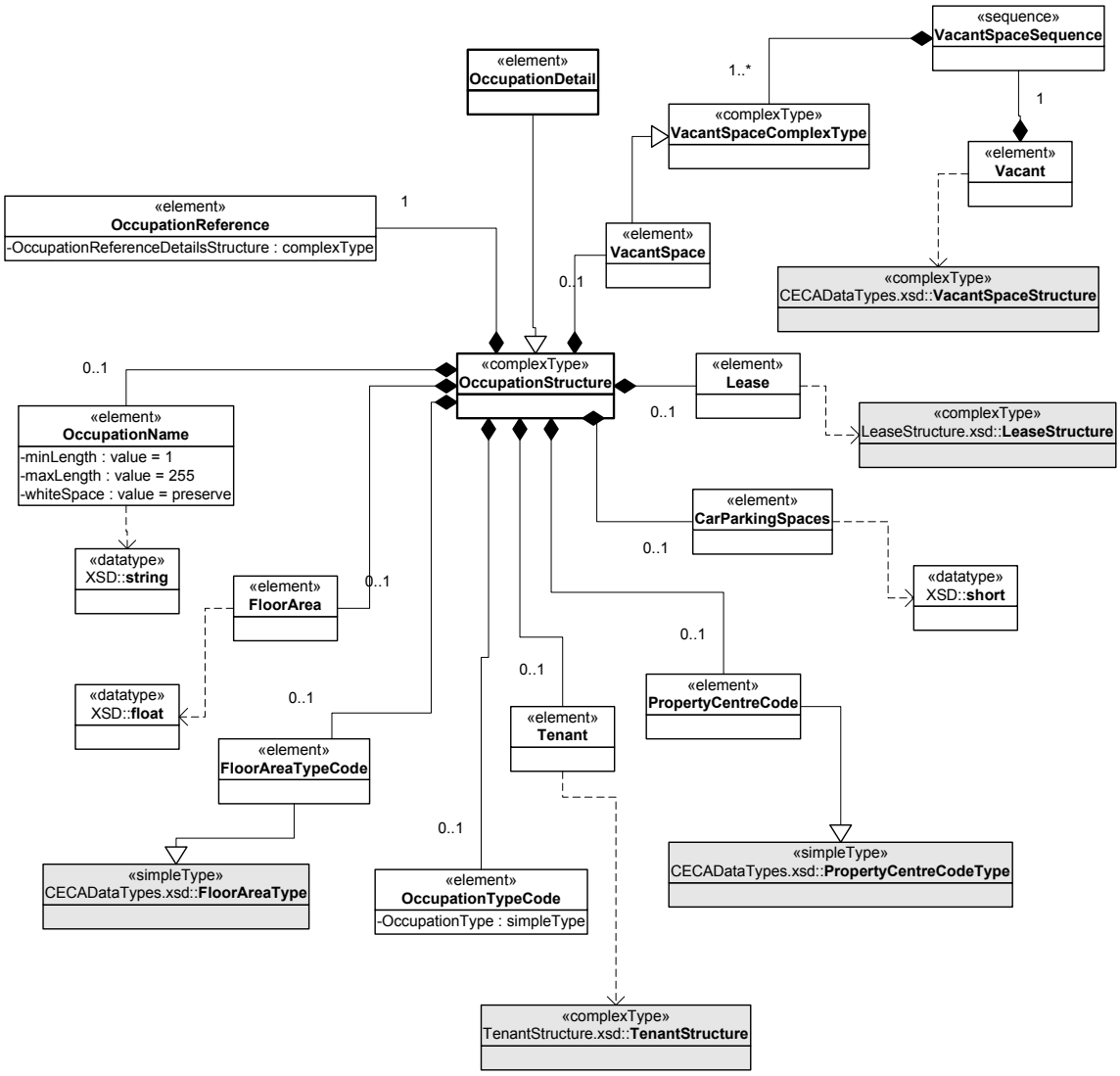


Figure 48: OccupationStructure – OccupationDetail element and OccupationStructure complexType

Figure 49: the XML schema definition for OccupationStructure.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="OccupationStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Property related occupation reference data

    Purpose: This schema is used to supply the base occupation data structures for use with occupation  messaging
schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core property occupation details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/LeaseStructure.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/TenantStructure.xsd"/>
    <xsd:simpleType name="OccupationType">
        <xsd:annotation>
            <xsd:documentation>
                The following descriptions apply to the enumerated types listed below
                        M: MOTO
                        O: Owner
                        L: Licence
                        S: Sub-Let
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="1"/>
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="O"/>
            <xsd:enumeration value="L"/>
            <xsd:enumeration value="S"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--The main element feeding off the LeaseStructure Complex Type-->
    <xsd:element name="OccupationDetail" type="OccupationStructure"/>
    <!--The main lease schema structure-->
    <xsd:complexType name="OccupationStructure">
        <xsd:sequence>
            <xsd:element name="OccupationReference" type="OccupationReferenceDetailsStructure"/>
            <xsd:element name="OccupationName" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="255"/>
                        <xsd:minLength value="1"/>
                        <xsd:whiteSpace value="preserve"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="FloorArea" type="xsd:float" minOccurs="0"/>
            <xsd:element name="FloorAreaTypeCode" type="FloorAreaType" minOccurs="0"/>
```

```
            <xsd:element name="OccupationTypeCode" type="OccupationType" minOccurs="0"/>
            <xsd:element name="PropertyCentreCode" type="PropertyCentreCodeType" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>The OGC reference to the owning property centre</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="CarParkingSpaces" type="xsd:short" minOccurs="0"/>
            <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
            <xsd:element name="Tenant" type="TenantStructure" minOccurs="0"/>
            <xsd:element name="Lease" type="LeaseStructure" minOccurs="0"/>
            <xsd:element name="VacantSpace" minOccurs="0">
                <xsd:complexType mixed="true">
                    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                        <xsd:element name="Vacant" type="VacantSpaceStructure"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="OccupationReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="OGCDeptOccupationRef">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                            <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                            <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"/>
                            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                            <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                            <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="OGCOccupationRef" type="OGCOccupationKeyStructure"/>
                <xsd:element name="DeptOccupationRef" type="DeptOccupationKeyStructure"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="OGCOccupationKeyStructure">
        <xsd:sequence>
            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
            <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
            <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="DeptOccupationKeyStructure">
        <xsd:sequence>
            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
            <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
            <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.8 LandlordStructure

This schema supports the architecture required for a CECA Landlord data entity. It provides a complexType Landlord*Structure* that is used by the messaging schemas.

LandlordStructure defines the data elements of a CECA Landlord.

LandlordStructure also defines a contact address for the associated landlord. This address element is defined from the complexType PropertyAddressStructure in schema CECAAddressStructure.

4.2.8.1 Unique Identifier

There is a complex Type defined Landlord*ReferenceDetails*, this provide the XML document author with the ability to supply unique tenant identifiers based on the types *OGCLandlordReference* as a mandatory element and optionally *DepartmentLandlordReference* (these types are sourced from CECADataTypes.xsd).

A tenant can be uniquely identified by a single reference (to be known as the Landlord code). An XML document "author" can therefore specify the tenant identifier as a unique OGC Landlord code (defined as simpleType OGCLandlordReferenceType) and optionally a unique departmental Landlord code (defined as simpleType DepartmentLandlordReferenceType).
The structure for this is defined in the complexType *LandlordReferenceDetailsStructure.*

```
<xsd:element name="LandlordReference" type="LandlordReferenceDetailsStructure"/>

<xsd:complexType name="LandlordReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:element name="OGCLandlordRef" type="OGCLandlordReferenceType"/>
        <xsd:element name="DeptLandlordRef" type="DepartmentLandlordReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

4.2.8.2 LandlordStructure UML and XML Schema

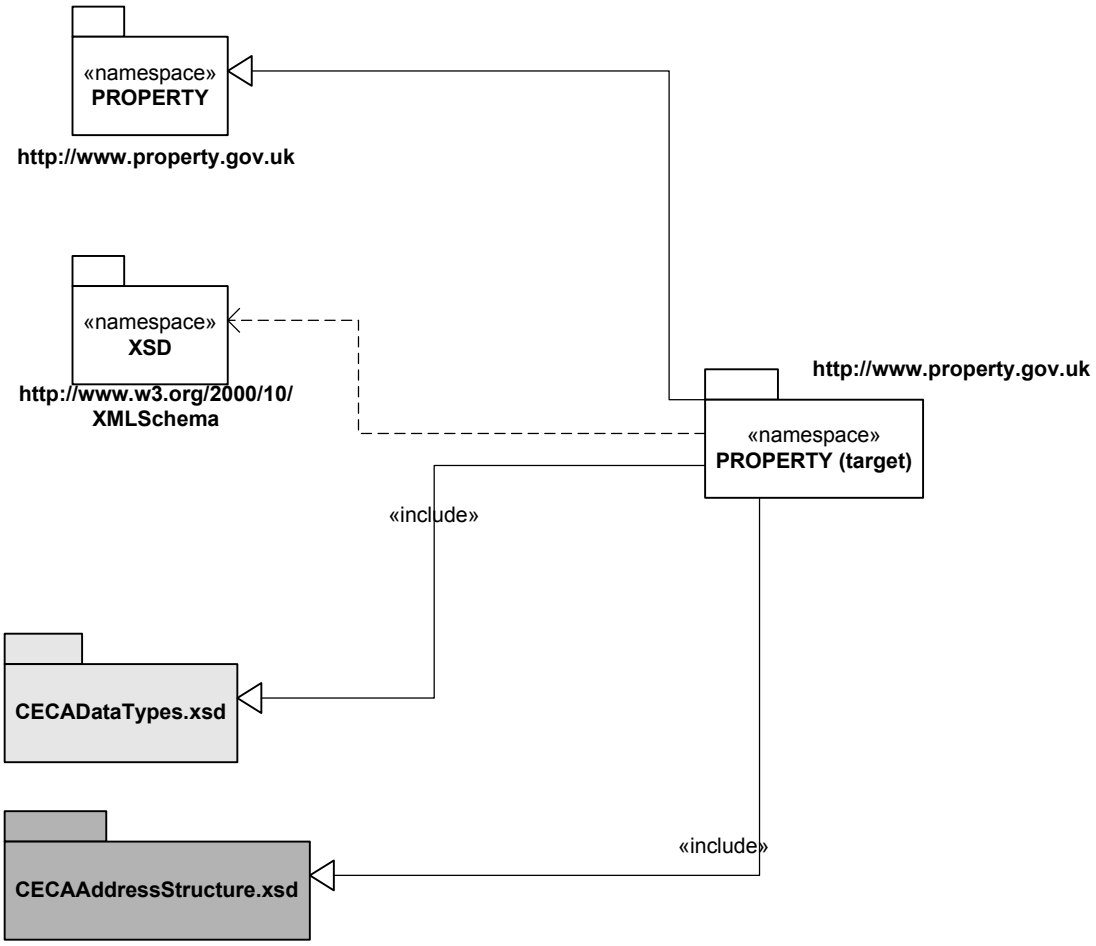
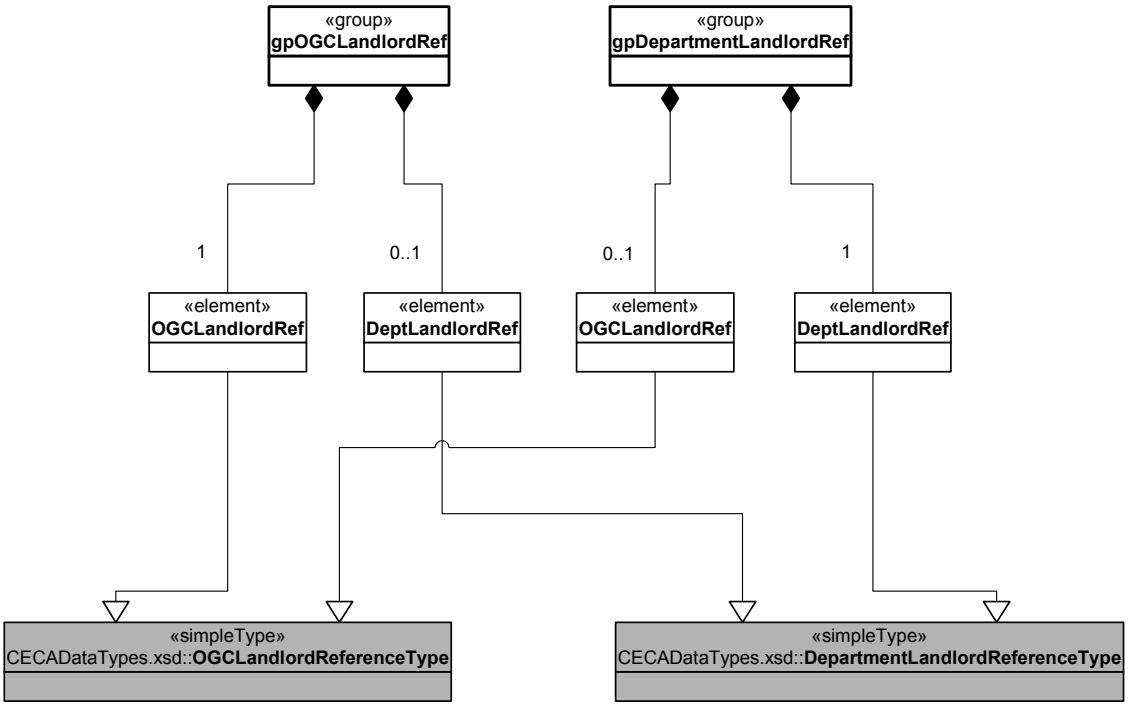


Figure 50: LandlordStructure – namespace



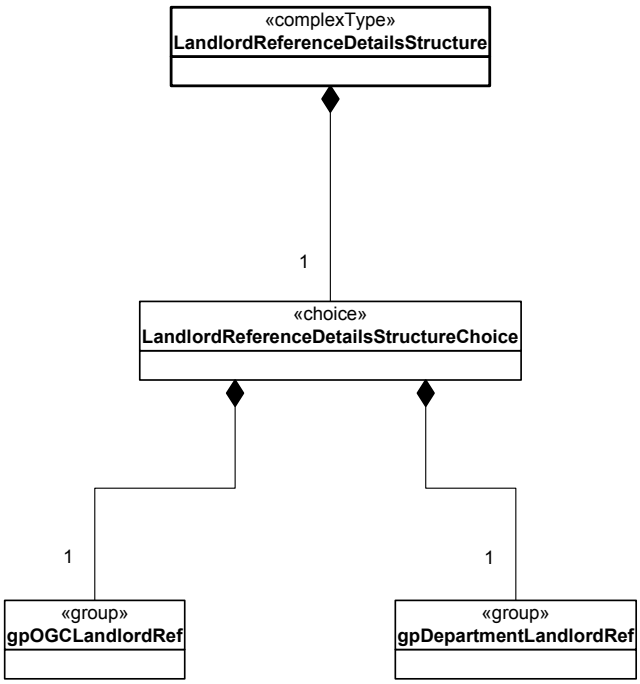


Figure 51: LandlordStructure – groups and complexTypes 1

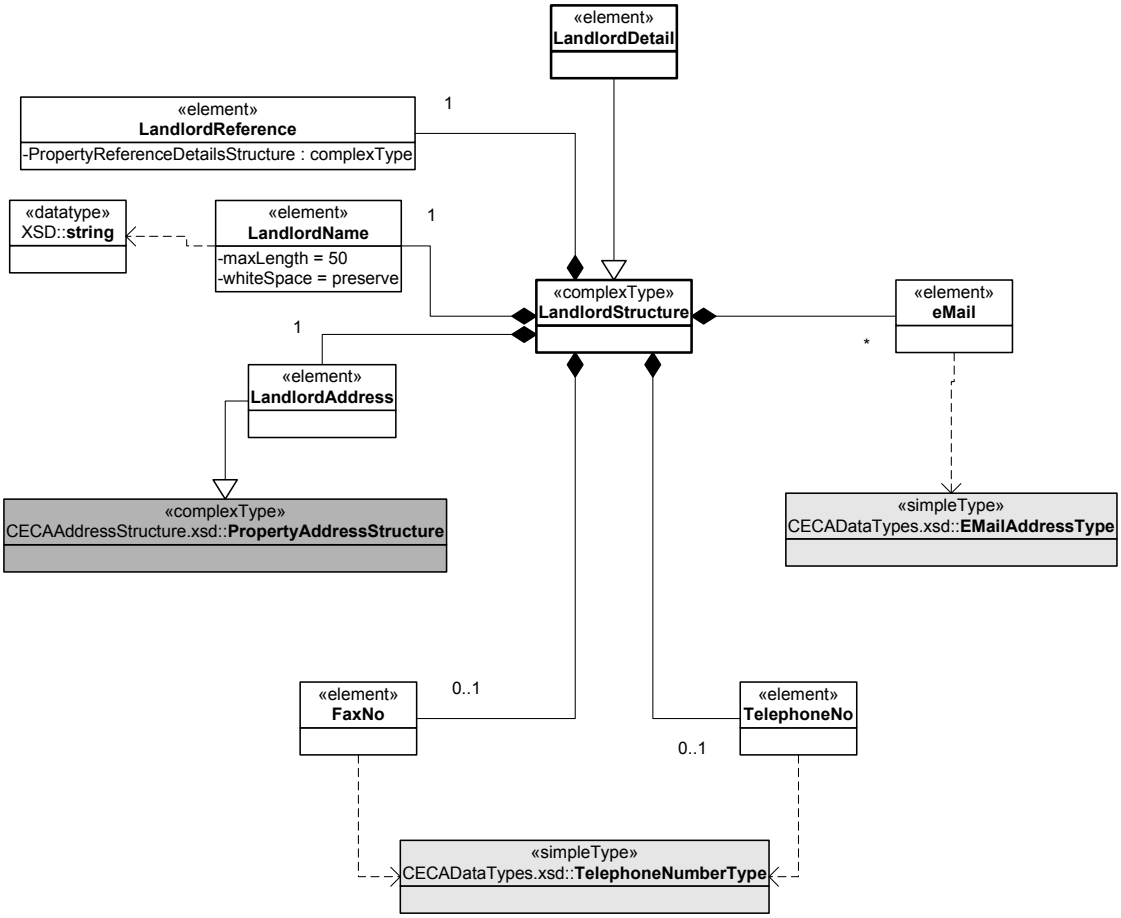


Figure 52: LandlordStructure – LandlordDetail element and LandlordStructure complexType

Figure 53: the XML schema definition for LandlordStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by Geoff E Parkin (Development Division) -->
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.property.gov.uk" elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1"
id="LandlordStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Landlord data

    Purpose: This schema is used to supply landlord reference details for use with lease messaging schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core landlord details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAAddressStructure.xsd"/>
    <!--The main element feeding off the PropertyStructure Complex Type-->
    <xsd:element name="LandlordDetail" type="LandlordStructure"/>
    <!--The main property schema structure-->
    <xsd:complexType name="LandlordStructure">
        <xsd:sequence>
            <xsd:element name="LandlordReference" type="LandlordReferenceDetailsStructure">
                <xsd:annotation>
                    <xsd:documentation>
                        This is the unique landlord reference which can be expressed in the following 3 formats
                        1. Using OGC landlord reference indicators. These are ID's used to reference landlord's within
the  OGC property reference database. Uses the simpleType OGCLandlordReference as a base.
                        2. Using departmental landlord reference ID. These are the ID's used within a departments
property management system (either in digital or manual format). Uses the simple type DepartmentLandlordReference
as a base.
                        3. An option to specify both. The xml document generator can specify both refrerence numbers
for completeness.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="LandlordName">
                <xsd:annotation>
                    <xsd:documentation>The OGC/Departmental name given to the Landlord</xsd:documentation>
                </xsd:annotation>
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="50"/>
                        <xsd:whiteSpace value="preserve"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="LandlordAddress" type="PropertyAddressStructure" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>
                        The xml document generator has an option as to which address format to use against the
landlord details.                    There are 2 choices
                        1. An address based on the OGC internal address format. This references the type
CECAAddressType                    located in the schema CECAAddressStructure.xsd
```

```
                    2. An address which is compliant to the bs7666 address format. The xml document uses the
type                        BSaddressStructure which is defined in the main bs7666 schema located in govtalk.

                    NOTE: Only 1 address format is acceptable. The preferred option for departments being bs7666.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="TelephoneNo" type="TelephoneNumberType" minOccurs="0"/>
            <xsd:element name="FaxNo" type="TelephoneNumberType" minOccurs="0"/>
            <xsd:element name="eMail" type="EmailAddressType" minOccurs="0"/>
            <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="LandlordReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:element name="OGCLandlordRef" type="OGCLandlordReferenceType"/>
            <xsd:element name="DeptLandlordRef" type="DepartmentLandlordReferenceType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.2.9 Tenant Structure

This schema supports the architecture required for a CECA tenant data entity. It provides a complexType *TenantStructure* that is used by the messaging schemas.

TenantStructure defines the data elements of a CECA tenant.

TenantStructure also defines a contact address for the associated tenant. This address element is defined from the complexType PropertyAddressStructure in schema CECAAddressStructure.

4.2.9.1 Unique Identifier

There is a complex Type defined *TenantReferenceDetails*, this provide the XML document author with the ability to supply unique tenant identifiers based on the types *OGCTenantReference* as a mandatory element and optionally *DepartmentTenantReference* (these types are sourced from CECADataTypes.xsd).

A tenant can be uniquely identified by a single reference (to be known as the tenant code). An XML document "author" can therefore specify the tenant identifier as a unique OGC tenant code (defined as simpleType *OGCTenantReferenceType*) and optionally a unique departmental tenant code (defined as simpleType *DepartmentTenantReferenceType*).
The structure for this is defined in the complexType *TenantReferenceDetailsStructure.*

```
<xsd:element name="TenantReference" type="TenantReferenceDetailsStructure"/>

<xsd:complexType name="TenantReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:element name="OGCTenantRef" type="OGCTenantReferenceType"/>
        <xsd:element name="DeptTenantRef" type="DepartmentTenantReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

4.2.9.2 TenantStructure UML and XML Schema

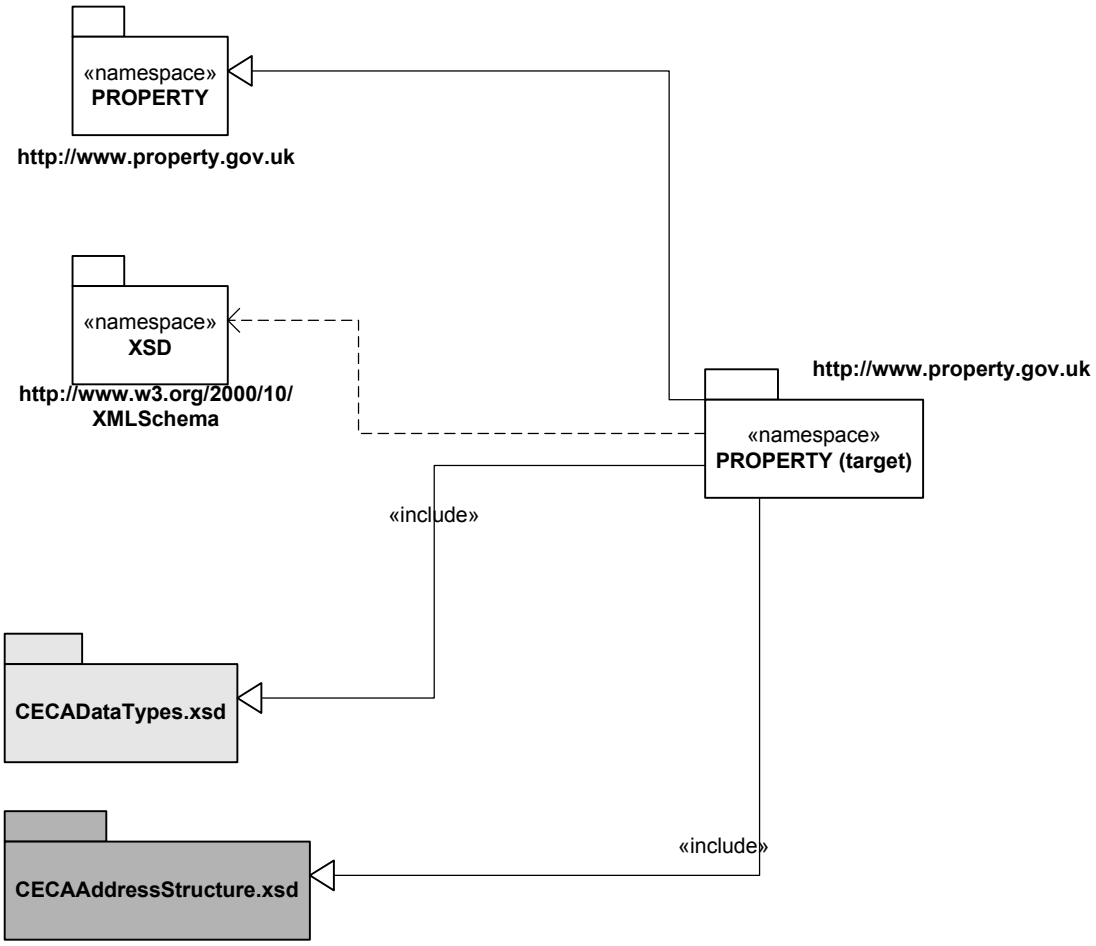
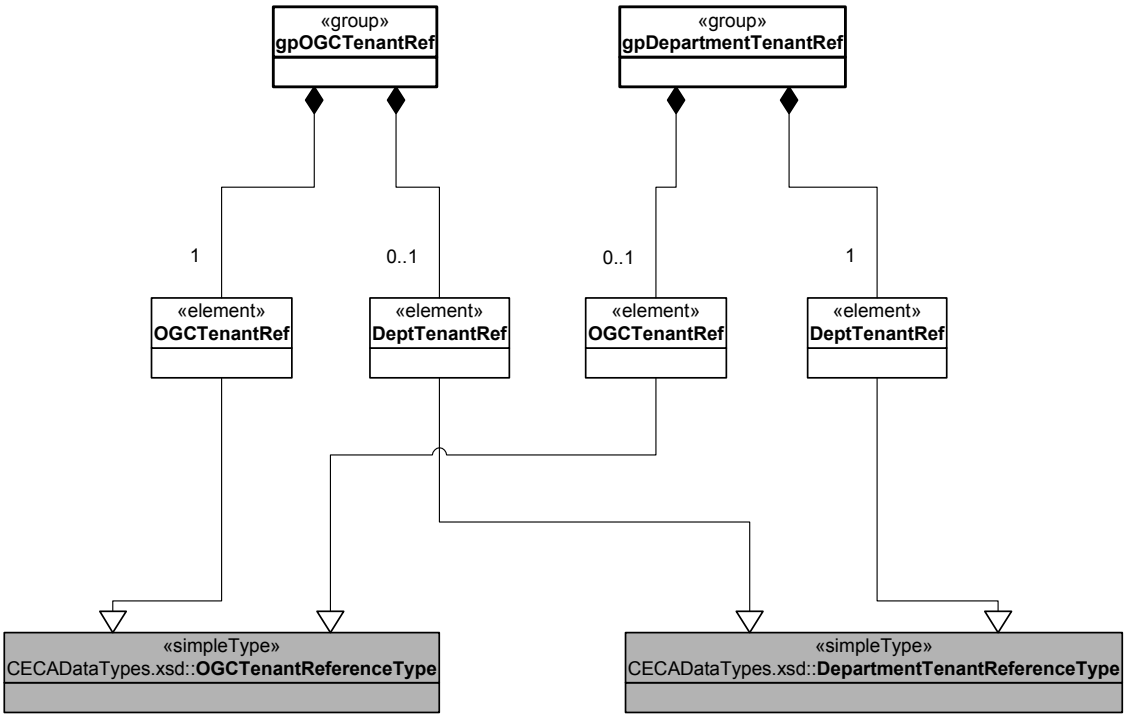


Figure 54: TenantStructure – namespace



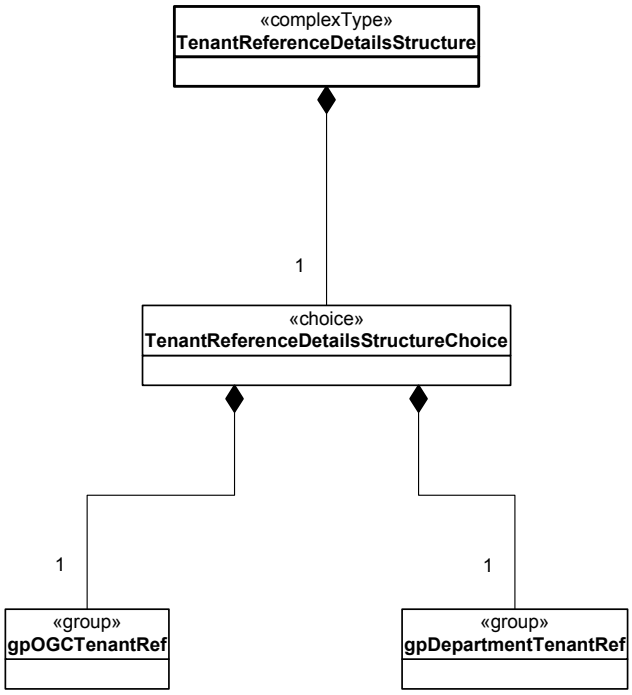


Figure 55: TenantStructure – groups and complexTypes 1

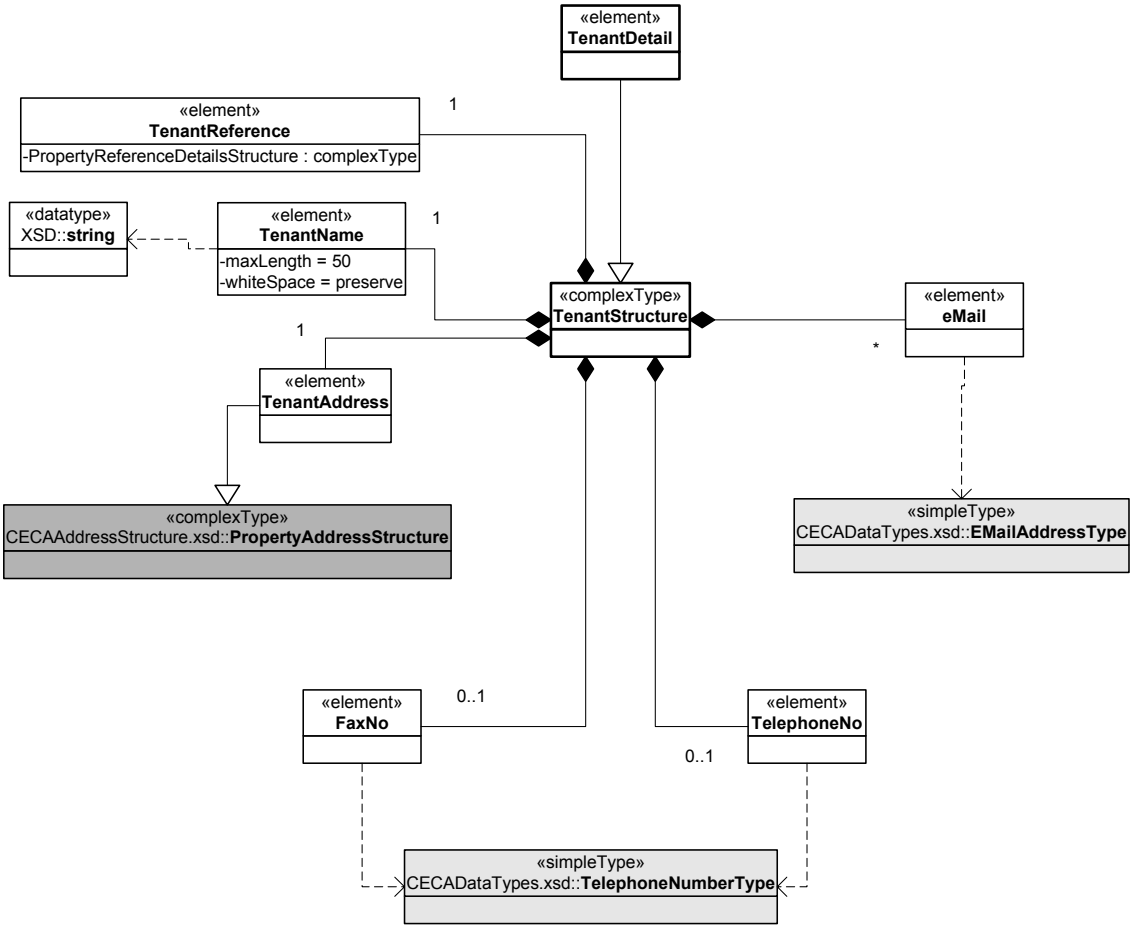


Figure 56: TenantStructure – TenantDetail element and TenantStructure complexType

Figure 57: the XML schema definition for TenantStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="TenantStructure">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Architecture Schema for Tenant data

    Purpose: This schema is used to supply Tenant reference details for use with occupation messaging schemas.

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
            This schema is used by a Government organisation to record and distribute core property details
         </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECADataTypes.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAAddressStructure.xsd"/>
    <xsd:element name="TenantDetail" type="TenantStructure"/>
    <xsd:complexType name="TenantStructure">
        <xsd:sequence>
            <xsd:element name="TenantReference" type="TenantReferenceDetailsStructure">
                <xsd:annotation>
                    <xsd:documentation>
                        This is the unique Tenant reference which can be expressed in the following 3 formats
                        1. Using OGC Tenant reference indicators. These are ID's used to reference Tenant's within the
OGC property reference database. Uses the simpleType OGCTenantReference as a base.
                        2. Using departmental Tenant reference ID. These are the ID's used within a departments
property management system (either in digital or manual format). Uses the simple type DepartmentTenantReference as
a base.
                        3. An option to specify both. The xml document generator can specify both refrerence numbers
for completeness.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="TenantName">
                <xsd:annotation>
                    <xsd:documentation>The OGC/Departmental name given to the Tenant</xsd:documentation>
                </xsd:annotation>
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="50"/>
                        <xsd:whiteSpace value="preserve"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="TenantAddress" type="PropertyAddressStructure" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>
                        The xml document generator has an option as to which address format to use against the
Tenant details.                    There are 2 choices
                        1. An address based on the OGC internal address format. This references the type
CECAAddressType                    located in the schema CECAAddressStructure.xsd
                        2. An address which is compliant to the bs7666 address format. The xml document uses the
type                    BSaddressStructure which is defined in the main bs7666 schema located in govtalk.
```

```
                    NOTE: Only 1 address format is acceptable. The preferred option for departments being bs7666.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="TelephoneNo" type="TelephoneNumberType" minOccurs="0"/>
            <xsd:element name="FaxNo" type="TelephoneNumberType" minOccurs="0"/>
            <xsd:element name="eMail" type="EmailAddressType" minOccurs="0"/>
            <xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="TenantReferenceDetailsStructure">
        <xsd:sequence>
            <xsd:element name="OGCTenantRef" type="OGCTenantReferenceType"/>
            <xsd:element name="DeptTenantRef" type="DepartmentTenantReferenceType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.3 CECA Messaging schemas

These schemas are base on the architecture schemas detailed in section 3.2. These are the schemas with which XML document generators must use for the transfer of bulk Civil Estate data from Estate management systems.

There are 2 types of messaging schemas defined:

- CECAEnvelope has been designed as the schema used to generate documents for transmission. This includes signature elements and a list of CECA messages based on one of the CECA object message schemas.
  THIS IS THE SCHEMA USED FOR BULK DATA TRANSMISSION
- CECA message schemas. These are based on the main Core CECA architecture schemas defined in section 3.2. Each message contains 2 elements; a reference to an architecture schema and MessageSignature element. The following CECA message schemas are defined:

  CECAProperty
  CECABuilding
  CECAHolding
  CECAOccupation
  CECALease

For bulk update document transmission these message schemas are never used directly, only the CECAEnvelope. However they have been defined to allow an Estate management system to generate individual messages held in individual documents independently prior to collating them together for transmission via the CECAEnvelope. Specifying each CECA object as a separate message provides flexibility to the CECA XML specification for other requirements other than Bulk Update message transmission.

The message signature associated with each message schema is referenced from the complexType CECAMessageSignature which specifies the following elements

- **Action**: the action to take with the data being presented in the message. Note where a message contains multiple data sets based on a hierarchical relationship (eg a CECAProperty message containing property and related buildings and holdings) the action is applied to each data entity.
  The valid values are Amend, Create, Delete.
- **UserID:** an identifier of the user who last updated the data or generated the actual message. This can be different from the User ID of actual document sender specified in the CECAEnvelope element.
- **MessageDate:** the date the message was generated.
- **AmendmentSource:** has a value OGC or DEPARTMENT. This indicates the direction of data transfer from Master Civil Estate database held by OGC and a departmental Estate management system
- **MessageComment:** any comment specific to the individual message.

4.3.1 CECAEnvelope

This is the main messaging schema which has been defined to facilitate the transmission of CECA data. This schema contains document "signing" elements which identify the XML message(s) sender.

The elements defined by the CECAEnvelope schemas are

- SchemaVersion: the version number of the schema(s) from which the XML message has been validated. This corresponds to the version attribute of the CECAEnvelope schema header. This provides an initial validation check to ensure XML messages submitted for processing conform to a supported version of the specification.
- UserID: the Epims user ID of the individual sending transmitting the document. Users must be registered as an Epims user data synchronisation/transmission with the main Civil Estate database. See http://www.epims.ogc.gsi.gov.uk/epims/epims_context_help.asp?help_file=HowToRegister for further details.
- AuthorName: name of the individual sending the document
- Date: date of document generation
- Email: email address of the individual sending the document as recorded within the epims user registration
- Department: the department code of the registered user
- PropertyCentre: the property centre of the registered user
- ConfirmProcessing: an indicator (Y/N) which confirms whether the sender wants to receive a message detailing the processing results of the data transmitted.
- AuthorisationRequired: an indicator (Y/N) which confirms whether the department sending the document requires the document to be processed an implemented XML document authentication system within the Epims architecture.
- VersionDetails: allows document sender to specify a versioning ID for the management of multiple document transmissions.
- Comments: any additional comments from the document sender

The schema also contains an messages element which define a sequence of actual CECA messages. These CECA messages are instances of one of the other message schemas detailed below which are in turn based on the architecture schemas defined in section 3. The messages element can contain messages of different types eg

```
<Messages>
    <CECAProperty>
        <Property>
            <PropertyReference>
                <OGCPropertyRef>58106</OGCPropertyRef>
                <DeptPropertyRef>0058106</DeptPropertyRef>
            </PropertyReference>
            Other elements here
        <Property>
    <CECAProperty>
    <CECAProperty>
        <Property>
            <PropertyReference>
                <OGCPropertyRef>33303</OGCPropertyRef>
                <DeptPropertyRef>0033303</DeptPropertyRef>
            </PropertyReference>
            Other elements here
        </Property>
    </CECAProperty>
    <CECAHolding>
        <Holding>
            Other elements here
```

```
        </Holding>
    </CECAHolding >
    <CECAOccupation>
        <Occupation>
            Other elements here
        </ Occupation >
    </ CECAOccupation >
</Messages>
```

This structure enables an XML document to contain a number of CECA messages using the CECAEnvelope as a wrapper. This addresses the requirement to allow the bulk transmission of multiple CECA objects.

A complexType has been defined to allow the choice of an individual message based on each of the CECA messages defined below

```
<xsd:complexType name="CECAMessageType">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:choice id="MessageChoice">
            <xsd:element ref="CECAProperty"/>
            <xsd:element ref="CECABuilding"/>
            <xsd:element ref="CECAHolding"/>
            <xsd:element ref="CECALease"/>
            <xsd:element ref="CECAOccupation"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

4.3.1.1 CECAEnvelope UML and XML Schema

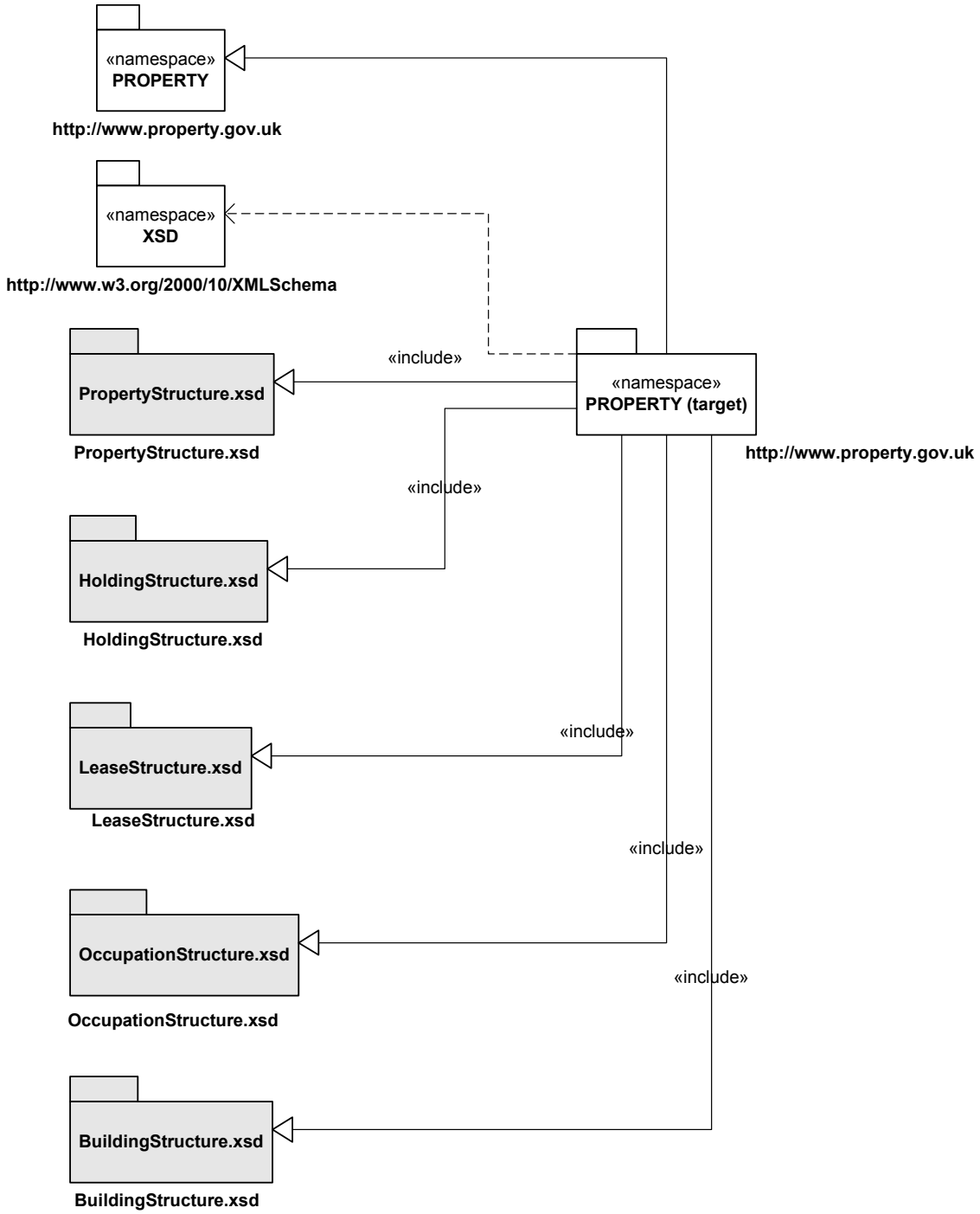


Figure 58: CECAEnvelope – namespace

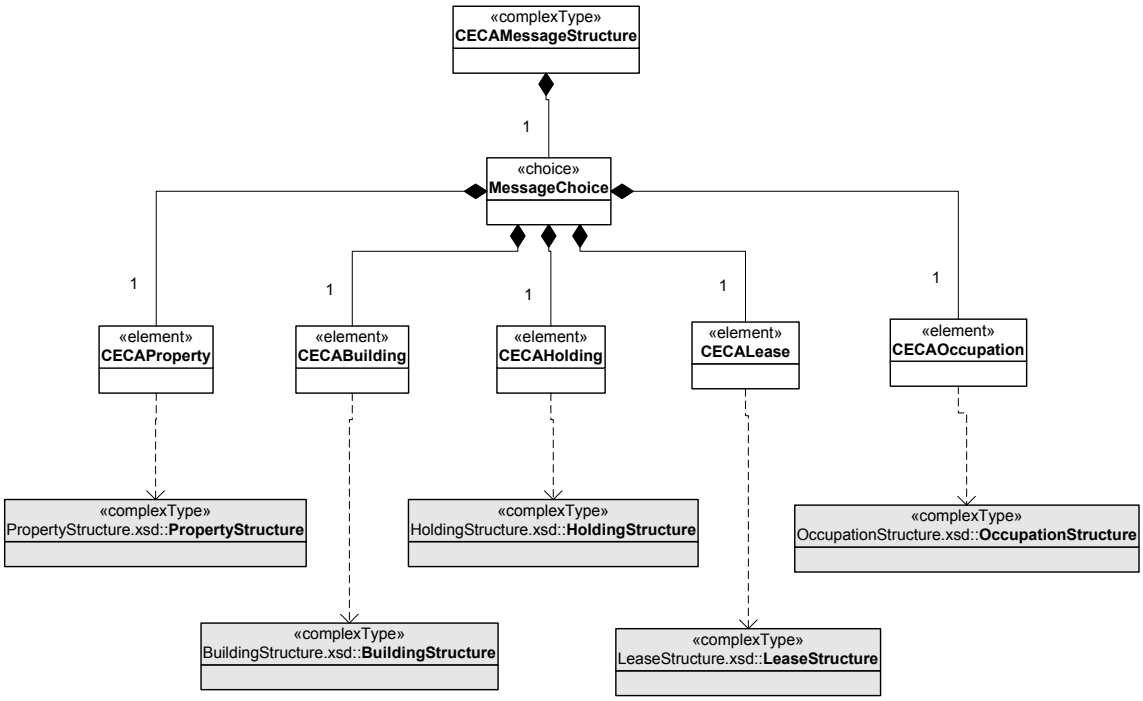


Figure 59: CECAEnvelope – CECAMessageStructure complexType

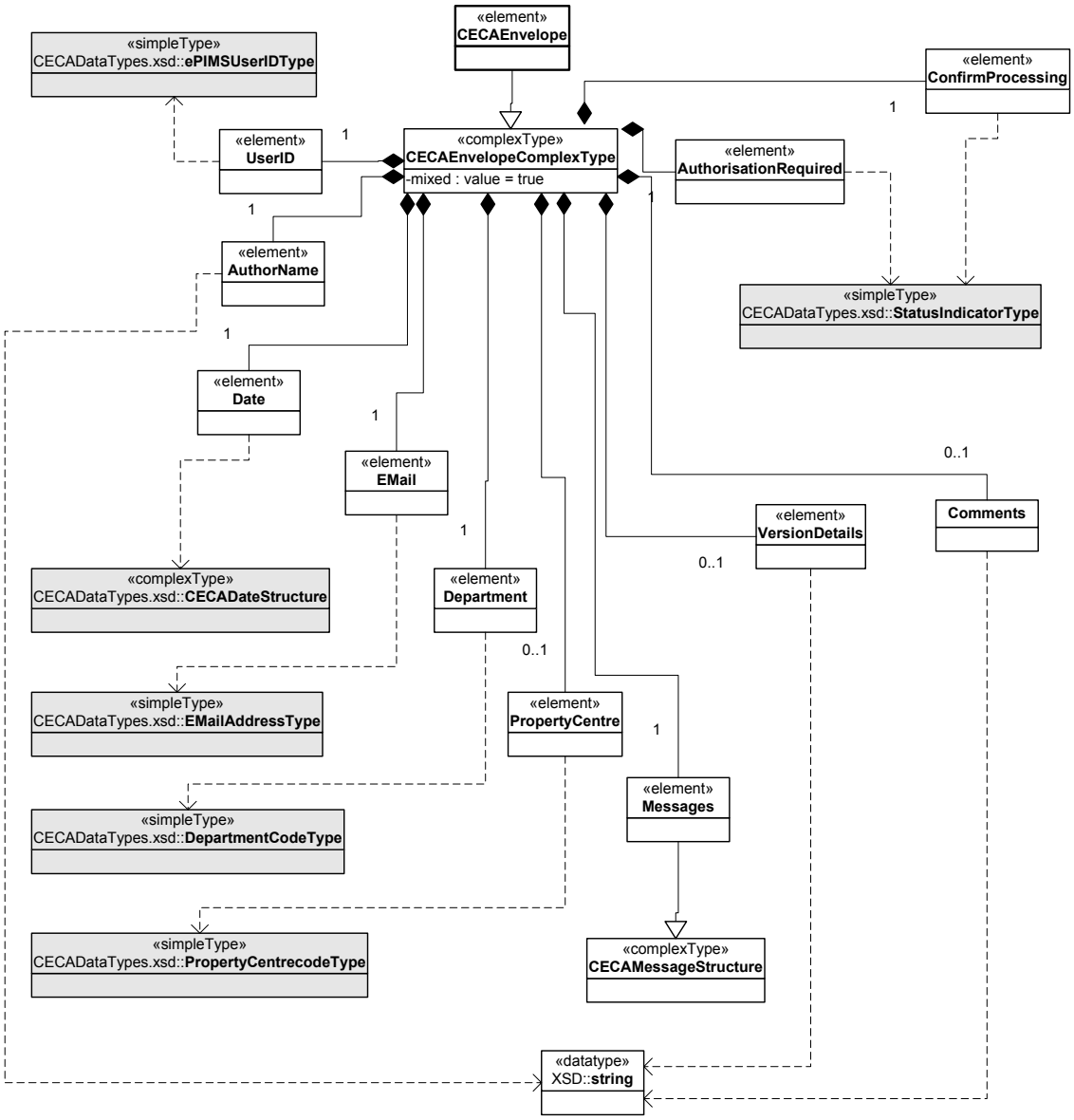


Figure 60: CECAEnvelope – CECAEnvelope element

Figure 61: the XML schema definition for CECAEnvelope.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.property.gov.uk" elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1"
id="CECAEnvelope">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Messaging Schema for Bulk Civil Estate data transfer

    Purpose: See above

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides the structure for the main message used in Bulk Update data transfer between
goevrnemnt          departmental estate management systems. This allows the transmission of a number of CECA
messages.           </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAProperty.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAbuilding.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAholding.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAlease.xsd"/>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/CECAoccupation.xsd"/>
    <xsd:element name="CECAEnvelope" type="EnvelopeType"/>
    <xsd:complexType name="EnvelopeType">
        <xsd:sequence>
            <xsd:element name="SchemaVersion" type="xsd:string"/>
            <xsd:element name="UserID" type="ePIMSUserIDType"/>
            <xsd:element name="AuthorName" type="xsd:string"/>
            <xsd:element name="Date" type="CECADateStructure"/>
            <xsd:element name="EMail" type="EmailAddressType"/>
            <xsd:element name="Department" type="DepartmentCodeType"/>
            <xsd:element name="PropertyCentre" type="PropertyCentreCodeType" minOccurs="0"/>
            <xsd:element name="ConfirmProcessing" type="StatusIndicatorType"/>
            <xsd:element name="AuthorisationRequired" type="StatusIndicatorType"/>
            <xsd:element name="VersionDetails" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Comments" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Messages" type="CECAMessageStructure"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CECAMessageStructure">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:choice id="MessageChoice">
                <xsd:element ref="CECAProperty"/>
                <xsd:element ref="CECABuilding"/>
                <xsd:element ref="CECAHolding"/>
                <xsd:element ref="CECALease"/>
                <xsd:element ref="CECAOccupation"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

4.3.2 CECAProperty

This messaging schema is used to transfer a CECA Property message based on the structures defined by PropertyStructure schema.

4.3.2.1 CECAProperty UML and XML Schema

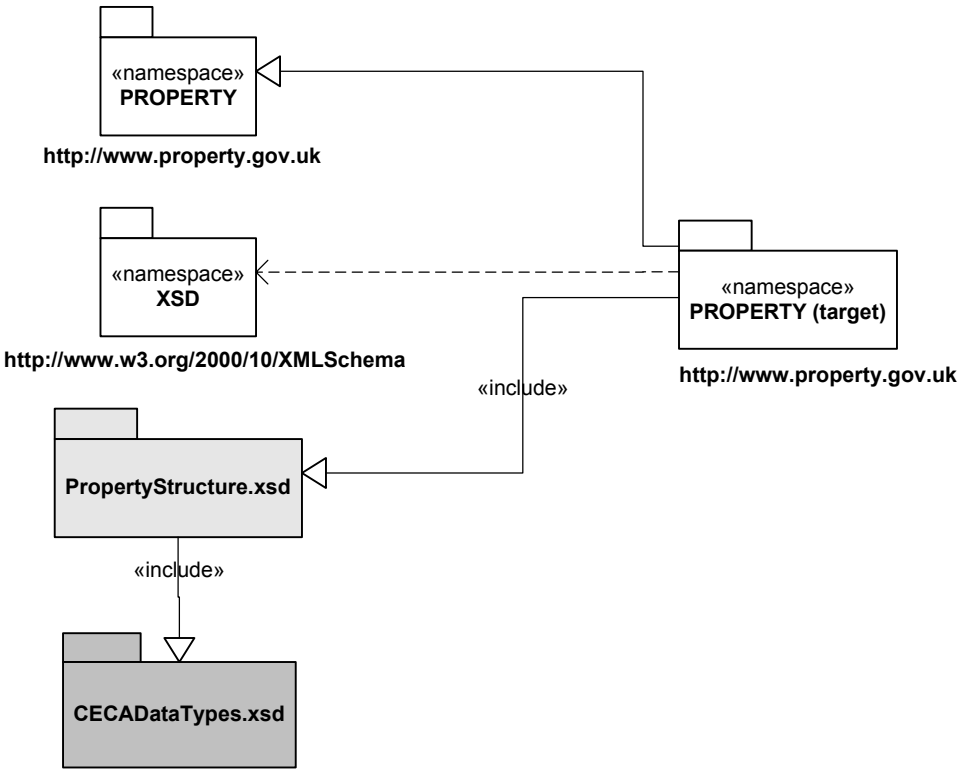


Figure 62: CECAProperty – namespace

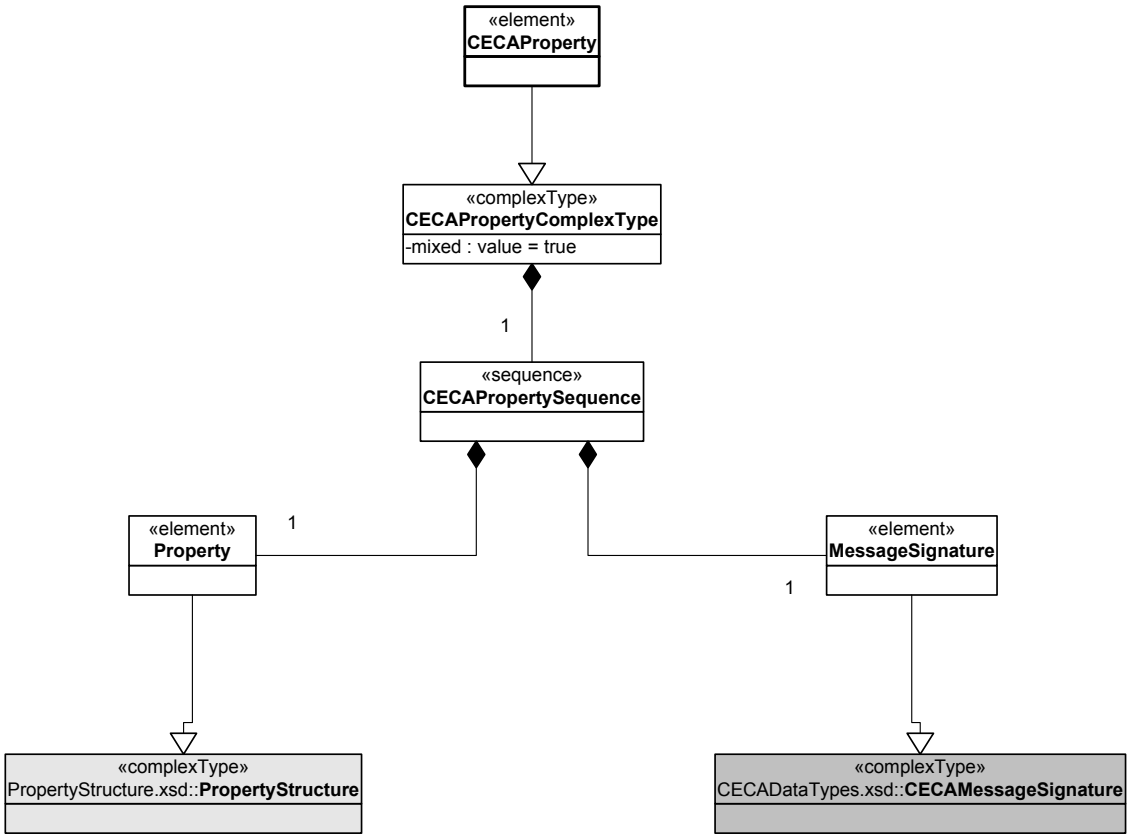


Figure 63: CECAProperty - CECAProperty element

Figure 64: the XML schema definition for CECAProperty.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.property.gov.uk" elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1"
id="CECAProperty">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Messaging Schema for transmission of a single property message

    Purpose: See above

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides the structure for a property message to be transmitted. this is a single property and
related CECA details (buildings, holdings etc).
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/PropertyStructure.xsd"/>
    <xsd:element name="CECAProperty">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:element name="Property" type="PropertyStructure"/>
                <xsd:element name="MessageSignature" type="CECAMessageSignature"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

4.3.3 CECABuilding

This messaging schema is used to transfer a CECA Building message based on the structures defined by BuildingStructure schema.

4.3.3.1 CECABuilding UML and XML Schema

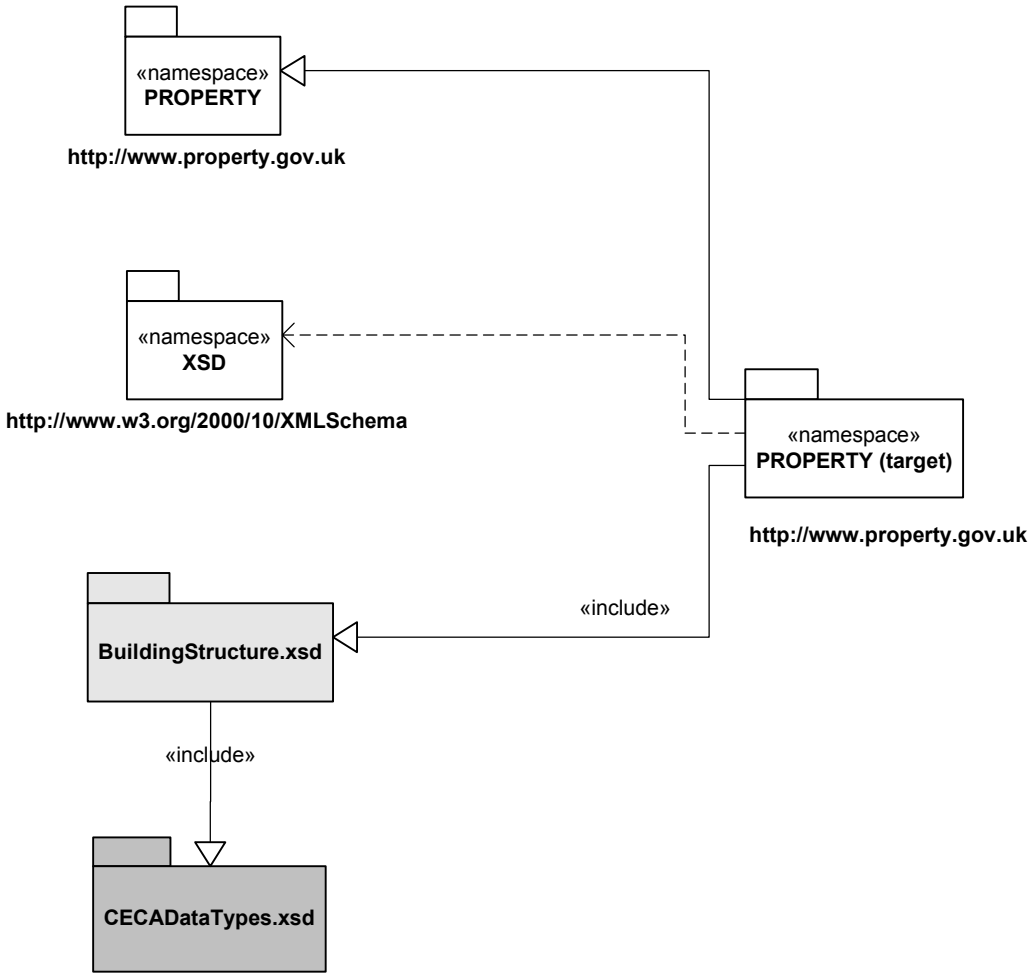


Figure 65: CECABuilding – namespace

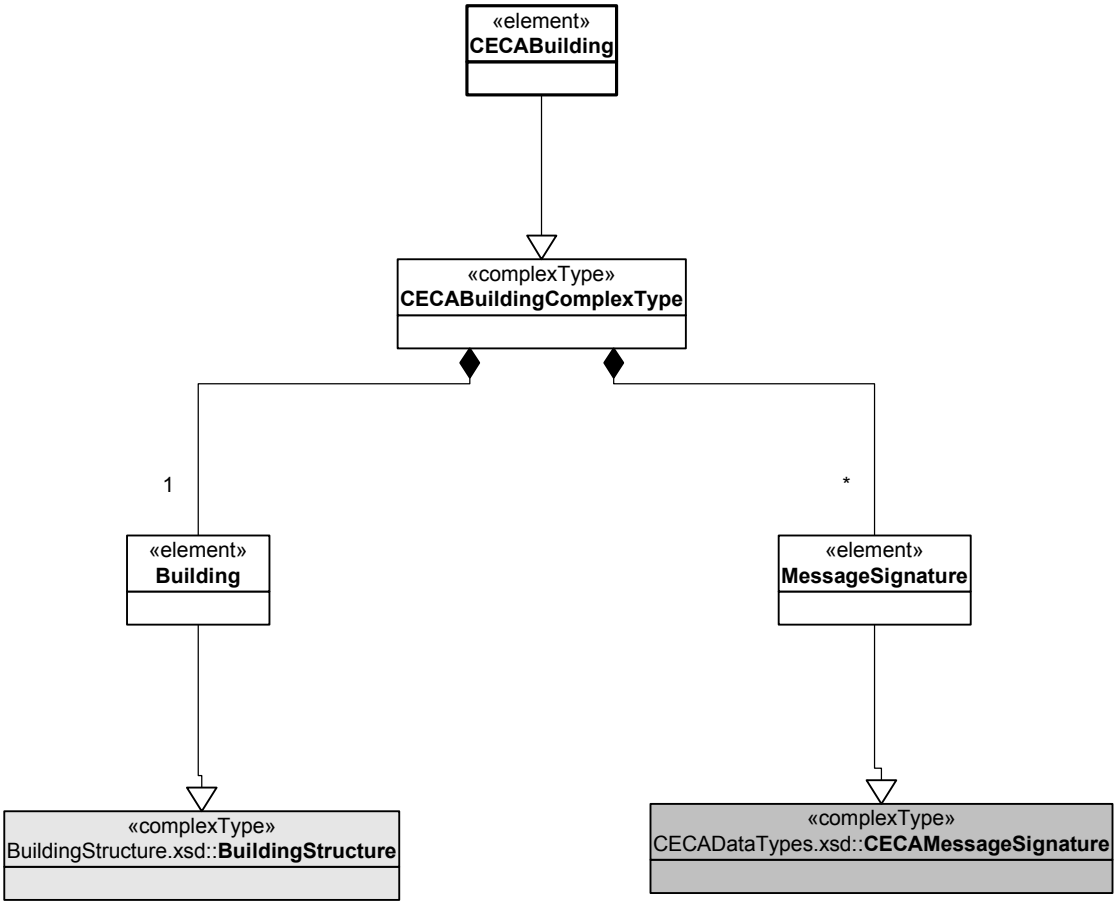


Figure 66: CECABuilding – CECABuilding element

Figure 67: the XML schema definition for CECABuilding.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="CECABuilding">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Messaging Schema for transmission of a single building message

    Purpose: See above

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides the structure for a building message to be transmitted. this is a single building.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/BuildingStructure.xsd"/>
    <xsd:element name="CECABuilding">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:element name="Building" type="BuildingStructure"/>
                <xsd:element name="MessageSignature" type="CECAMessageSignature"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

4.3.4 CECAHolding

This messaging schema is used to transfer a CECA Holding message based on the structures defined by HoldingStructure schema.

4.3.4.1 CECAHolding UML and XML Schema

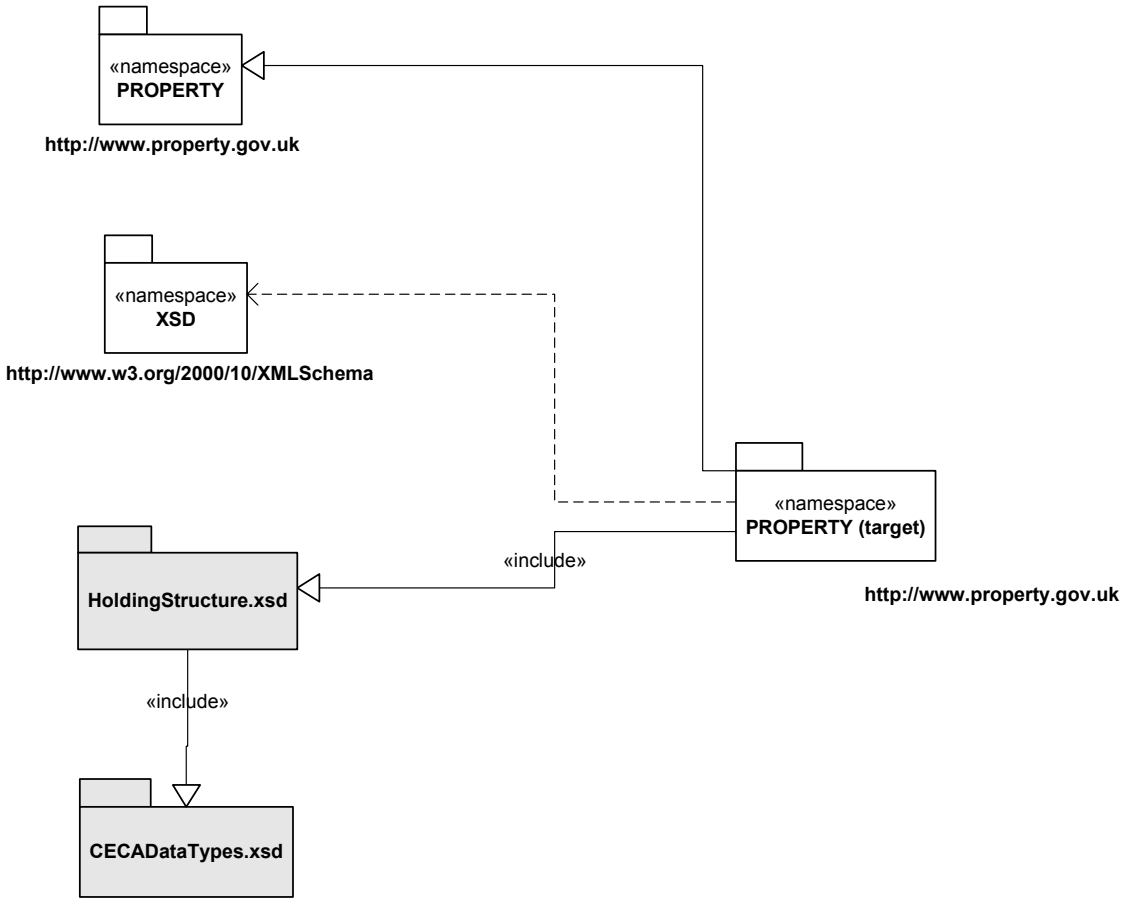


Figure 68: CECAHolding - namespace

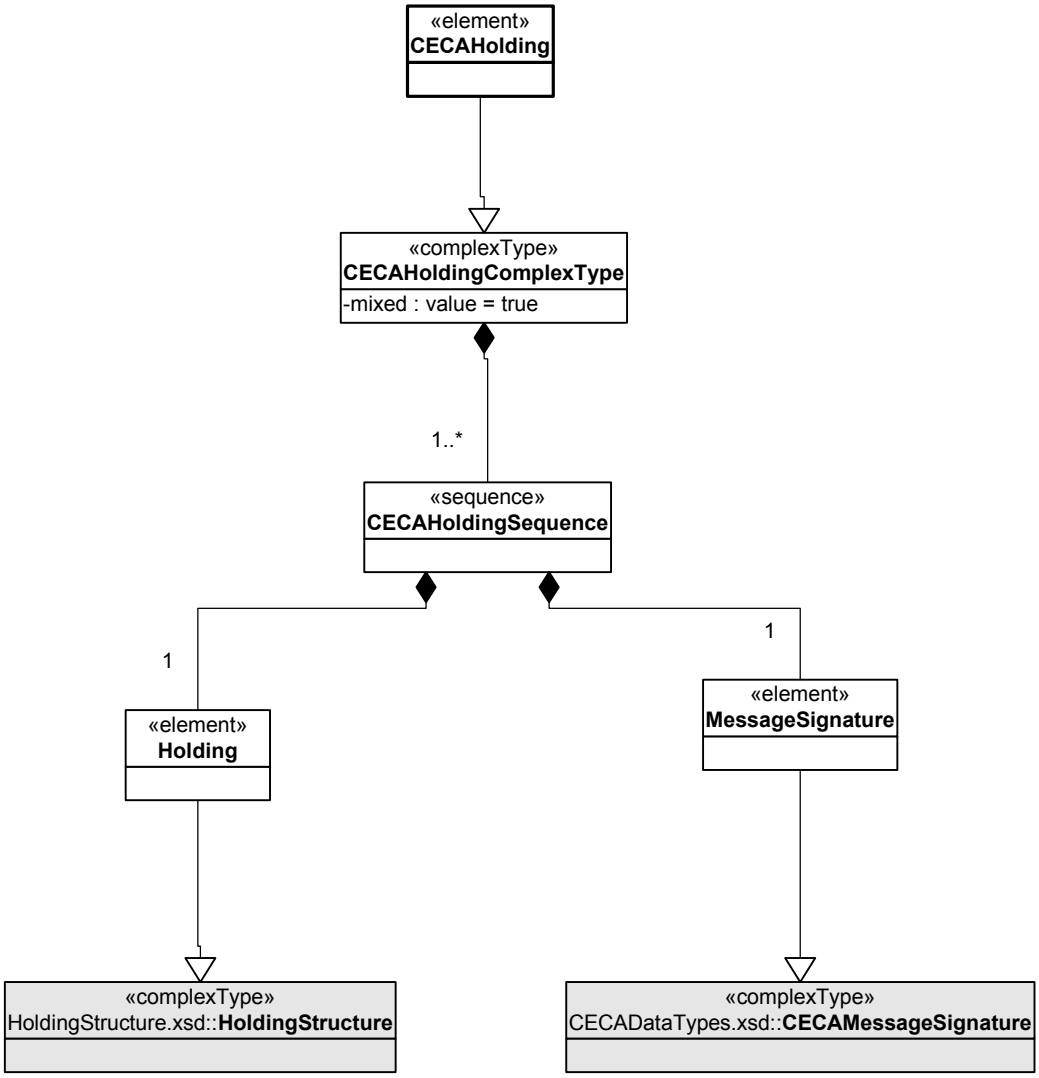


Figure 69: CECAHolding – CECAHolding element

Figure 70: the XML schema definition for CECAHolding.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.1" id="CECAHolding">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Messaging Schema for transmission of a single holding message

    Purpose: See above

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides the structure for a holding message to be transmitted. this is a single holding.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/HoldingStructure.xsd"/>
    <xsd:element name="CECAHolding">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:element name="Holding" type="HoldingStructure"/>
                <xsd:element name="MessageSignature" type="CECAMessageSignature"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

4.3.5 CECALease

This messaging schema is used to transfer a CECA Lease message based on the structures defined by LeaseStructure schema.

4.3.5.1 CECALease UML and XML Schema

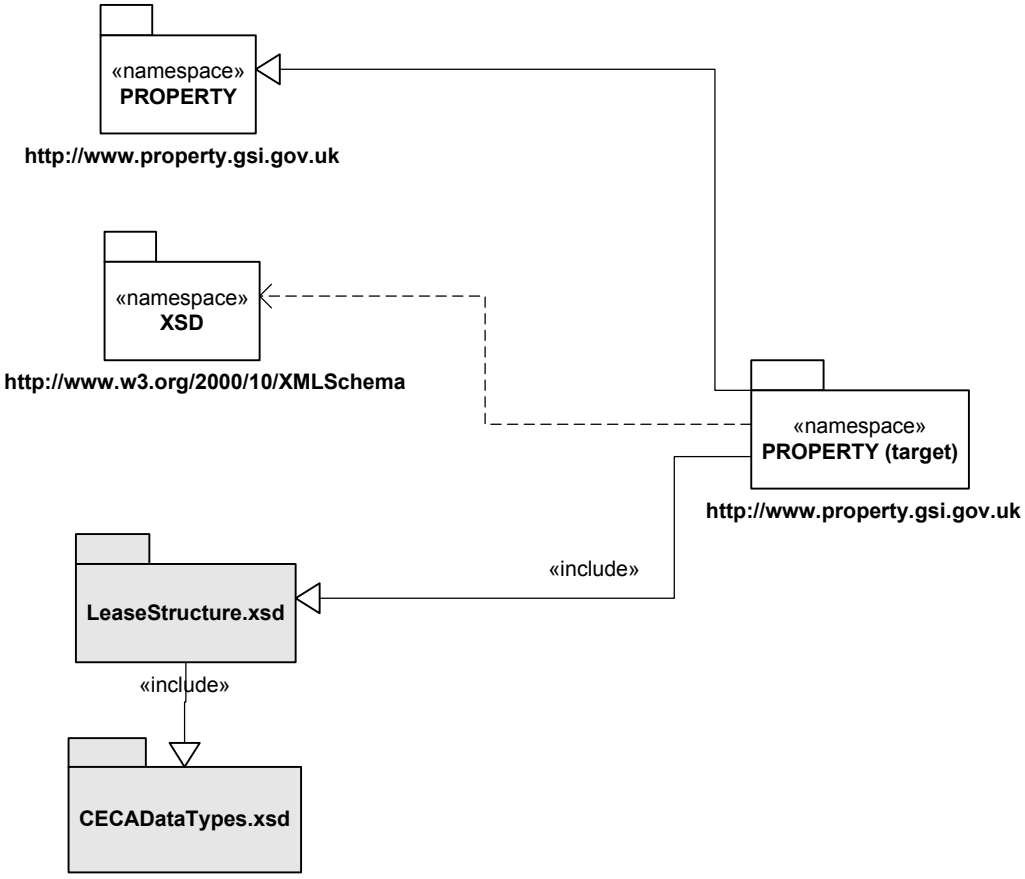


Figure 71: CECALease – namespace

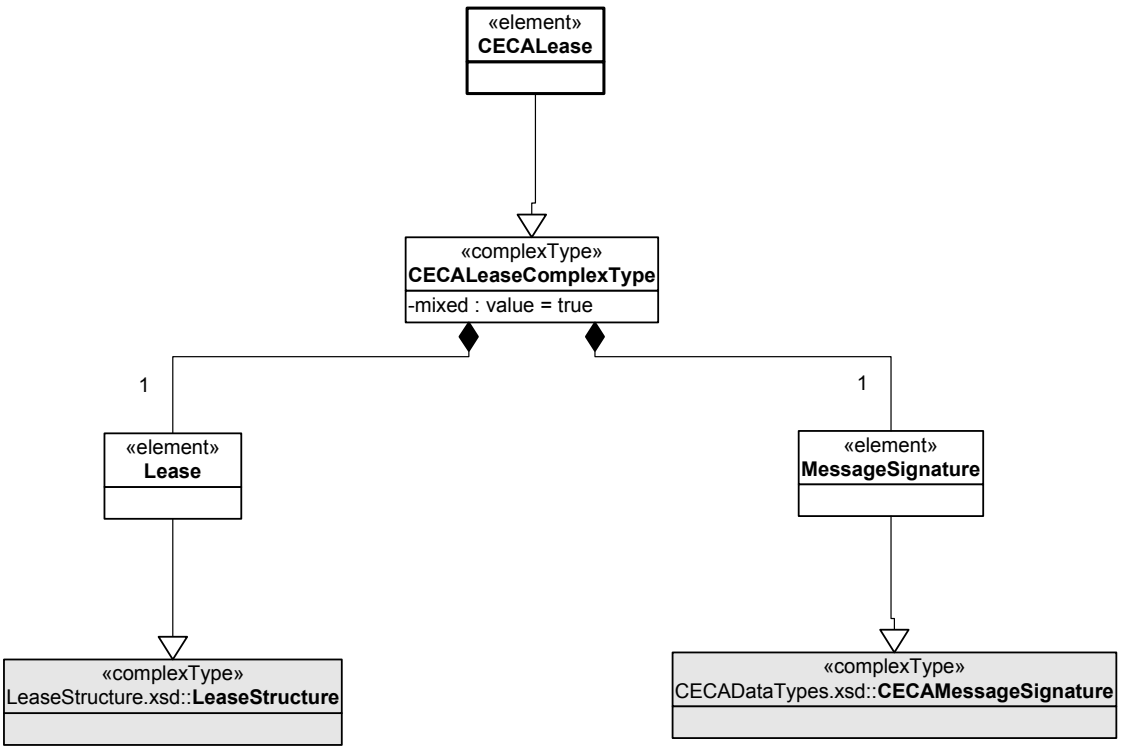


Figure 72: CECALease - CECALease element

Figure 73: the XML schema definition for CECALease.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.property.gov.uk" elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1"
id="CECALease">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Messaging Schema for transmission of a single lease message

    Purpose: See above

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides the structure for a lease message to be transmitted. this is a single lease.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/LeaseStructure.xsd"/>
    <xsd:element name="CECALease">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:element name="Lease" type="LeaseStructure"/>
                <xsd:element name="MessageSignature" type="CECAMessageSignature"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

4.3.6 CECAOccupation

This messaging schema is used to transfer a CECA Occupation message based on the structures defined by OccupationStructure schema.

4.3.6.1 CECAOccupation UML and XML Schema

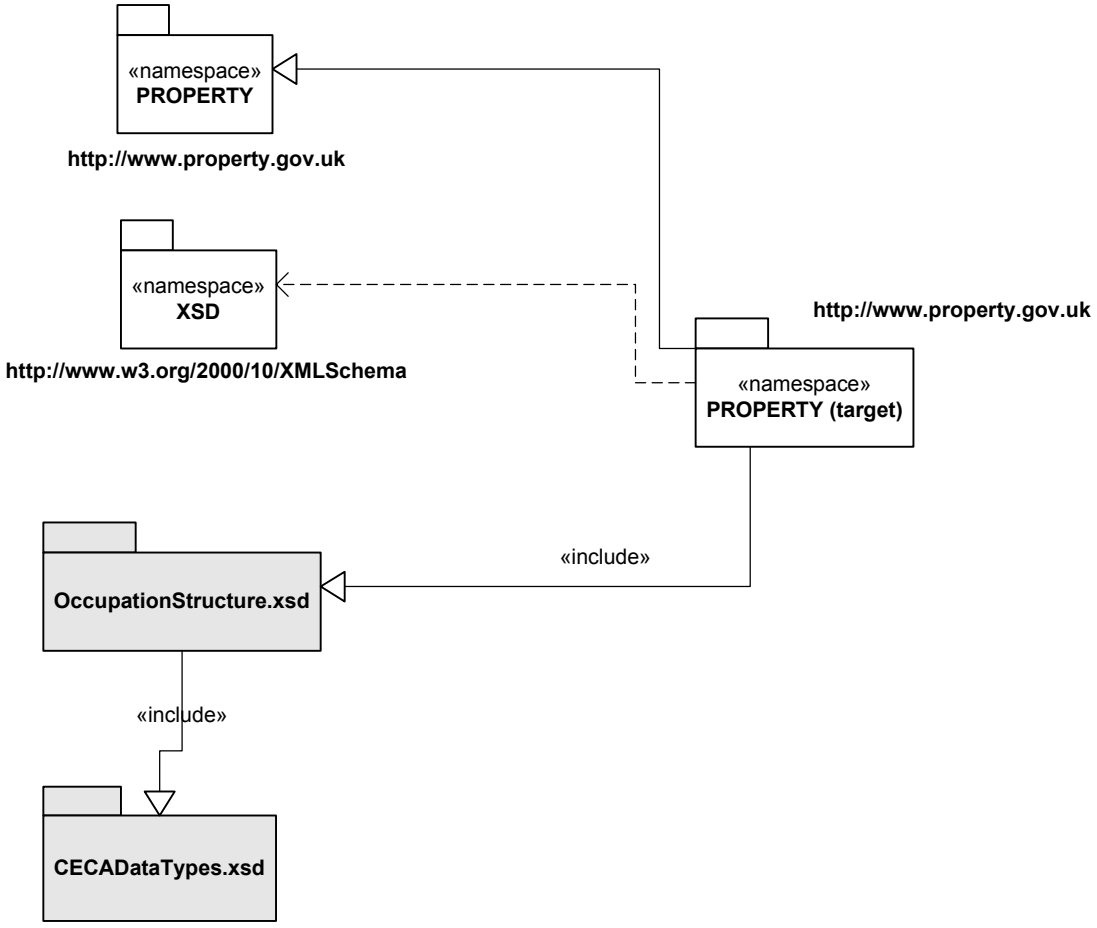


Figure 74: CECAOccupation – namespace

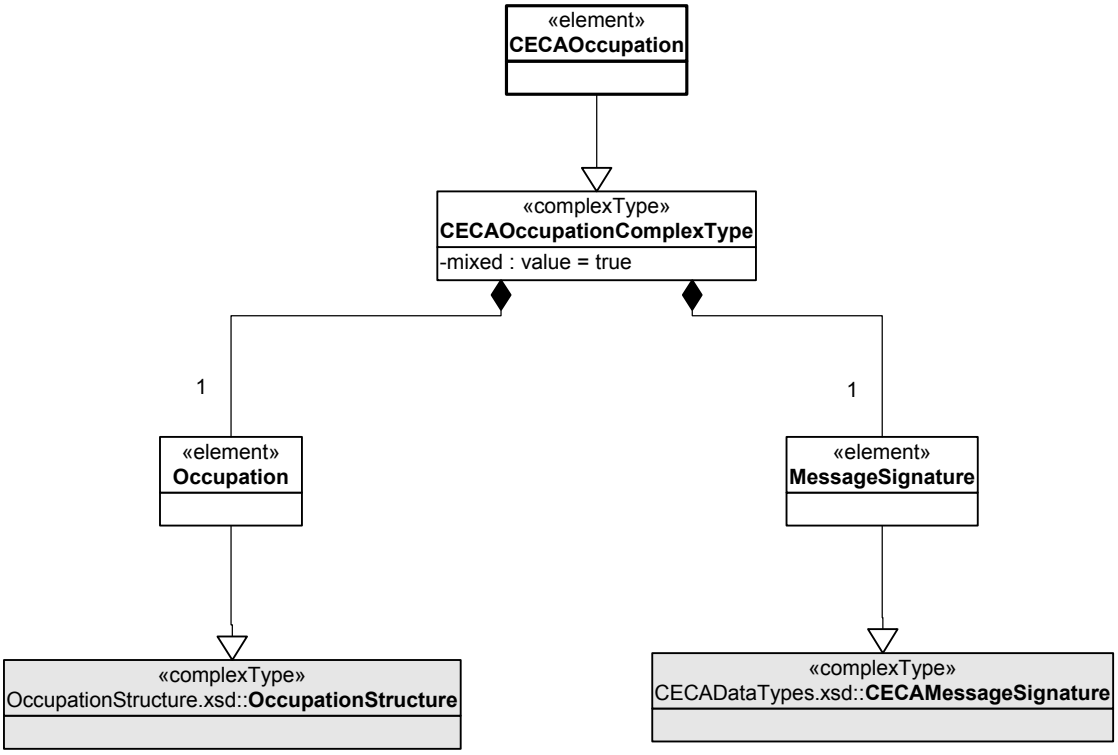


Figure 75: CECAOccupation – CECAOccupation element

Figure 76: the XML schema definition for CECAOccupation.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Geoff Parkin - Office of Government Commerce -->
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.property.gov.uk" elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1"
id="CECAOccupation">
    <!--
    OGC - Office of Government Commerce : PCD - Property and Construction Directorate

    XML Messaging Schema for transmission of a single occupation message

    Purpose: See above

    Date: 28/02/2002

    Version: 2.1
    Author: Geoff Parkin, ePIMS Development Team
    -->
    <xsd:annotation>
        <xsd:appinfo>
            <xsd:KeyWords>
            property, building, OGC, OS, holding, CECA, occupation, lease, vacant space, landlord, tenant, ePims, Civil
Estate
            </xsd:KeyWords>
        </xsd:appinfo>
        <xsd:documentation>
        This schema provides the structure for a occupation message to be transmitted. this is a single occupation.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="http://www.property.gov.uk/schemas/ceca/xml/OccupationStructure.xsd"/>
    <xsd:element name="CECAOccupation">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:element name="Occupation" type="OccupationStructure"/>
                <xsd:element name="MessageSignature" type="CECAMessageSignature"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

5 Approval

6 Abbreviations

Abb	Definition
OGC	Office of Government Commerce
eGIF	Electronic Government Interoperability Framework
HTTP	Hypertext Transfer Protocol
OeE	Office of the E-Envoy
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

7 References

	Document References
[1]	The eGovernment Interoperability Framework
[2]	UK Online – Information Architecture - Overview, draft 0.1a, 28 July 2000
[3]	XML Schema Part 0: Primer, W3C Working Draft, 7 April 2000 http://www.w3.org/TR/xmlschema-0/
[4]	XML Schema Part 1: Structures, W3C Working Draft, 7 April 2000 http://www.w3.org/TR/xmlschema-1/
[5]	XML Schema Part 2: Datatypes, W3C Working Draft, 7 April 2000 http://www.w3.org/TR/xmlschema-2/
[6]	Data Standards Catalogue, Version 0.3, March 2001
[7]	UK Govtalk A UML notation for XML Schema, Version 0.2, December 2000
[8]	UK Online – Information Architecture – Change of Address Schema, Version 0.1c, February 2001

Appendix A: Version control

Version 2.0 to 2.1 changes

BuildingStructure.xsd

- Removed group definitions for Building Key structure

```
<xsd:group name="gpOGCFGNHoldingRef">
    <xsd:sequence>
        <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
        <xsd:element name="DepHoldingRef" type="DepartmentHoldingReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpDepartmentFGNHoldingRef">
    <xsd:sequence>
        <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType" minOccurs="0"/>
        <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpDepartmentBuildingRef">
    <xsd:sequence>
        <xsd:element name="OGCBuildingRef" type="OGCBuildingKeyStructure" minOccurs="0"/>
        <xsd:element name="DeptBuildingRef" type="DeptBuildingKeyStructure"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCBuildingRef">
    <xsd:sequence>
        <xsd:element name="OGCBuildingRef" type="OGCBuildingKeyStructure"/>
        <xsd:element name="DeptBuildingRef" type="DeptBuildingKeyStructure" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
```

- Removed element BuildingItemNo

```
<xsd:element name="BuildingItemNo" type="xsd:int">
    <xsd:annotation>
        <xsd:documentation>The OGC unique identifier for each building related to a specific
property</xsd:documentation>
    </xsd:annotation>
</xsd:element>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended BuildingKey structures for BuildingReferenceDetailsStructure

```
<xsd:complexType name="BuildingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptBuildingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCBReference" type="OGCBuildingReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptBReference" type="DepartmentBuildingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCBuildingRef" type="OGCBuildingKeyStructure"/>
            <xsd:element name="DeptBuildingRef" type="DeptBuildingKeyStructure"/>
        </xsd:choice>
```

```
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="OGCBuildingKeyStructure">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="OGCBReference" type="OGCBuildingReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DeptBuildingKeyStructure">
    <xsd:sequence>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        <xsd:element name="DeptBReference" type="DepartmentBuildingReferenceType"/>
    </xsd:sequence>
 </xsd:complexType>
```

- Amended linked holdings reference structure

```
<xsd:complexType name="FGNHoldingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptHoldingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
                        <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCHoldingRef" type="OGCHoldingReferenceType"/>
            <xsd:element name="DeptHoldingRef" type="DepartmentHoldingReferenceType"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

HoldingStructure.xsd

- Removed group definitions for Holding Key structure

```
<xsd:group name="gpDepartmentHoldingRef">
    <xsd:sequence>
        <xsd:element name="OGCHoldingRef" type="OGCHoldingKeyStructure" minOccurs="0"/>
        <xsd:element name="DeptHoldingRef" type="DeptHoldingKeyStructure"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCHoldingRef">
    <xsd:sequence>
        <xsd:element name="OGCHoldingRef" type="OGCHoldingKeyStructure"/>
        <xsd:element name="DeptHoldingRef" type="DeptHoldingKeyStructure" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended HoldingKey structures for HoldingReferenceDetailsStructure

```
<xsd:complexType name="HoldingReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptHoldingRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCHoldingRef" type="OGCHoldingKeyStructure"/>
            <xsd:element name="DeptHoldingRef" type="DeptHoldingKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

LeaseStructure.xsd

- Removed group definitions for Lease Key structure

```
<xsd:group name="gpDepartmentLeaseRef">
    <xsd:sequence>
        <xsd:element name="OGCLeaseRef" type="OGCLeaseKeyStructure" minOccurs="0"/>
        <xsd:element name="DeptLeaseRef" type="DeptLeaseKeyStructure"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCLeaseRef">
    <xsd:sequence>
        <xsd:element name="OGCLeaseRef" type="OGCLeaseKeyStructure"/>
        <xsd:element name="DeptLeaseRef" type="DeptLeaseKeyStructure" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended LeaseKey structures for LeaseReferenceDetailsStructure

```
<xsd:complexType name="LeaseReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptLeaseRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"
minOccurs="0"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                        <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"
minOccurs="0"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCLeaseRef" type="OGCLeaseKeyStructure"/>
            <xsd:element name="DeptLeaseRef" type="DeptLeaseKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

- Added new element DepartmentNotes to complexType BreakReviewStructure

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

OccupationStructure.xsd

- Removed group definitions for Occupation Key structure

```
<xsd:group name="gpDepartmentOccupationRef">
    <xsd:sequence>
        <xsd:element name="OGCOccupationRef" type="OGCOccupationKeyStructure" minOccurs="0"/>
        <xsd:element name="DeptOccupationRef" type="DeptOccupationKeyStructure"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCOccupationRef">
    <xsd:sequence>
        <xsd:element name="OGCOccupationRef" type="OGCOccupationKeyStructure"/>
        <xsd:element name="DeptOccupationRef" type="DeptOccupationKeyStructure" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended OccupationKey structures for OccupationReferenceDetailsStructure

```
<xsd:complexType name="OccupationReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptOccupationRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="OGCHReference" type="OGCHoldingReferenceType"/>
                        <xsd:element name="OGCOReference" type="OGCOccupationReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                        <xsd:element name="DeptHReference" type="DepartmentHoldingReferenceType"/>
                        <xsd:element name="DeptOReference" type="DepartmentOccupationReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCOccupationRef" type="OGCOccupationKeyStructure"/>
            <xsd:element name="DeptOccupationRef" type="DeptOccupationKeyStructure"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

PropertyStructure.xsd

- Removed group definitions for Property Key structure

```
<xsd:group name="gpDepartmentPropertyRef">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType" minOccurs="0"/>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCPropertyRef">
    <xsd:sequence>
        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:group
```

- Amended base type of element OSGridRefAccuracy

```
<xsd:element name="OSGridRefAccuracy" type="xsd:integer">
    <xsd:annotation>
        <xsd:documentation>OGC internal data to be decided</xsd:documentation>
    </xsd:annotation>
</xsd:element>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended PropertyKey structures for PropertyReferenceDetailsStructure

```
<xsd:complexType name="PropertyReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="OGCDeptPropertyRef">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
                        <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="OGCPReference" type="OGCPropertyReferenceType"/>
            <xsd:element name="DeptPReference" type="DepartmentPropertyReferenceType"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

TenantStructure.xsd

- Removed group definitions for Tenant Key structure

```
<xsd:group name="gpDepartmentTenantRef">
    <xsd:sequence>
        <xsd:element name="OGCTenantRef" type="OGCTenantReferenceType" minOccurs="0"/>
        <xsd:element name="DeptTenantRef" type="DepartmentTenantReferenceType"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCTenantRef">
    <xsd:sequence>
        <xsd:element name="OGCTenantRef" type="OGCTenantReferenceType"/>
        <xsd:element name="DeptTenantRef" type="DepartmentTenantReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended TenantKey structures for TenantReferenceDetailsStructure

```
<xsd:complexType name="TenantReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:element name="OGCTenantRef" type="OGCTenantReferenceType"/>
        <xsd:element name="DeptTenantRef" type="DepartmentTenantReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

LandlordStructure.xsd

- Removed group definitions for Landlord Key structure

```
<xsd:group name="gpDepartmentLandlordRef">
    <xsd:sequence>
        <xsd:element name="OGCLandlordRef" type="OGCLandlordReferenceType" minOccurs="0"/>
        <xsd:element name="DeptLandlordRef" type="DepartmentLandlordReferenceType"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="gpOGCLandlordRef">
    <xsd:sequence>
        <xsd:element name="OGCLandlordRef" type="OGCLandlordReferenceType"/>
        <xsd:element name="DeptLandlordRef" type="DepartmentLandlordReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
```

- Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

- Amended TenantKey structures for LandlordReferenceDetailsStructure

```
<xsd:complexType name="LandlordReferenceDetailsStructure">
    <xsd:sequence>
        <xsd:element name="OGCLandlordRef" type="OGCLandlordReferenceType"/>
        <xsd:element name="DeptLandlordRef" type="DepartmentLandlordReferenceType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

CECADataTypes.xsd

- Amendments to complexType VacantSpaceStructure
    - Changed maxLength of element VacantSpaceDesc to 2000

```
<xsd:element name="VacantSpaceDesc" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="2000"/>
            <xsd:whiteSpace value="preserve"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
```

    - Added new element DepartmentNotes

```
<xsd:element name="DepartmentNotes" type="DepartmentalNotesType" minOccurs="0"/>
```

CECAEnvelope.xsd

- Removed duplicate/invalid element schemaVersion

```
<xsd:element name="name=schemaVersion" type="xsd:string"/>
```

- Amended CECAMessageStructure assigning minOccurs="0" to allow no messages to be implemented in the CECAEnvelope

```
<xsd:complexType name="CECAMessageStructure">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:choice id="MessageChoice">
            <xsd:element ref="CECAProperty"/>
            <xsd:element ref="CECABuilding"/>
            <xsd:element ref="CECAHolding"/>
            <xsd:element ref="CECALease"/>
            <xsd:element ref="CECAOccupation"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

Version 1.0 to 2.0 changes

The CECA XML Specification version 1.0 has undergone a formal Request for Comments (RFC) process through the UK Govtalk e-Gif process and a number of issues and enhancements to the specification have been raised.

Following the draft of the new version (to be labeled version 1.1) the specification will undergo another formal RFC via e-Gif. Further drafts and modifications to the standard will be dependant upon comments received.

The proposed changes to the specification are detailed below.

Syntax, logic and validation errors

A number of errors both syntactically and logically based were found on a quality review of the specification. These are detailed below

1. Figure 1 , Page 9, Join from Property to building is misleading;
   Page 89, comment - a new property must contain at least one building - NO, a property does not require a building although advisable;

The property building relationship is zero to many.
The figure is indicating a one to many relationship this should be zero to many.
The changes required for this is detailed further below.

2. Page 44, <xsd:element_name=OSNorttherly type double t;

3. Page 58, <xsd:element_name=AirCondInd incorrect description, refers to central heating;

Annotation element should read "Does the building have air conditioning"

4. Page 65, Tenure Type, Should be Freehold/Feuhold not Freehold/Feahold;

5. Page 98, comments at bottom rel new holding must contain 1 lease and holding can contain zero lease contradict each other;

New schemas

A number of new schemas have been introduced to support the CECA object hierarchy architecture and to manage CECA XML messages.
These new schemas are detailed below

CECAEnvelope.xsd

The CECAEnvelope schema defines a structure for the transmission of Bulk Updates of Civil Estate data and provides a number of element to allow the "signing" of the data providing identification, authorisation and version control details.

CECAEnvelope schema : one instance of this message will accompany an entire "file/document" of CECA messages and contain the following details

UserID – Epims User ID of person generating the message
AuthorName – name of the person generating the message
Date – date of message generation
Email – email address of person generating messages as recorded against Epims userID
PropertyCentre – the property centre code of the person generating the message (note source of available codes still to be defined)
ConfirmProcessing – a flag indicating whether the message instigator requires confirmation and processing results.
VersionDetails – optional. To allow versioning details both departmental and OGC related. Needs further discussion regarding ability to manage and record versioning details for specific messages which will be dynamic in content and action.
Comments – optional comments from XML document author.
Messages – a sequence of one or more CECA messages based on the 5 main CECAMessage schemas defined in this specification.

LandlordStructure.xsd

A schema defining the structure of data to represent a Landlord.

The LandlordStructure type will be applied as a referenced element in LeaseStructure architecture schema.

The following elements to be implemented for LandlordStructure

LandlordID – a complexType consisting of OGC and departmental reference code.
LandlordName
LandlordAddress (reference CECAAddressStructure type)
LandlordTelNo – telephone number
LandlordFaxNo – fax number
LandlordContact – contact at landlord
LandlordeMail – email contact address

TenantStructure.xsd

A schema defining the structure of data to represent a Tenant.

The TenantStructure type will be applied as a referenced element in OccupationStructure architecture schema.

The following elements to be implemented for TenantStructure:

TenantID – a complexType consisting of OGC and departmental reference code.

DepartmentCode – associated owning department code
TenantName
TenantAddress (reference CECAAddressStructure type)
ContactName – name of contact person
ContactTelNo – contact telephone number
ContactFaxNo – contact fax number
ContacteMail – contact email address

Deleted Schemas

CECAFullProperty.xsd

The functionality implemented by this schemas has been replaced by the existing schema CECAProperty due to changes to the relationships implemented in the CECAHierarchy.

Amended Schemas

CECADataTypes

New simple Types

OGCLandlordReferenceType
OGCTenantReferenceType
DepartmentOccupationReferenceType
DepartmentLandlordReferenceType
DepartmentCodeType
VacancyCodeType
VacantStatusType
RefurbReqdType
EmailAddressType
TelephoneNumberType
EPIMSUserIDType

New ComplexTypes

CECAMessageSignature
VacantSpaceStructure

Document Signing
A requirement exists to identify what a CECA XML message is attempting to do any by whom. This "action" can be targeted toward an entire document containing multiple messages or a single message contained in an XML "instance" ie a complete single message.
Some consideration should be made relating to the validity/use of the eGif envelope for basic message details.

For this requirement we will implement the following

CECAAction – this is a complexType defined in CECADataTypes schema which determines the type of action being implemented in a message. This is implemented in all messaging schemas and will be mandatory. The following elements will be defined

Action – based on a defined enumerated type with the following values (Add, Amend, Delete)
UserID – Epims user ID of person implementing change to the data
Date – date (and optionally time) the data was actually changed.
Comments – to be used to specify any specific comments regarding this specific update.

Vacant Space

Data structures are required for the definition of vacant space. These will be implemented as a complex type definition in CECADataTypes schema.
The vacant space type will be implemented as a list of elements (zero or many) applied to HoldingStructure and OccupationStructure archirecture schemas.

The required elements are:

VacantSpaceID – internal reference for a vacant space record as specified within OGC data structures
VacantSpace – total vacant space in m^2
VacantSpaceDesc – description of vacant space
VacancyTypeCode – type of vacant space record referencing pisces table code_vacancy_type
FloorAreaTypeCode – type of floor area referencing pisces table code_floor_area_type

DateAvailFrom – date vacant space available from
DateAvailUntil – date vacant space available until
DateLogged – date the vacant space details were logged
StatusCode – status of the vacant space record referenced from pisces table code_status (where object = 'V')
ContactName and contact details – build a contact complexType to define the structure of a contact
OGCProfAdvisor – pace officer id associated with vacant space ??? Maybe auto populate from holding details as per ePIMS.
NoofBuildings – number of buildings
NoofFloors – number of floors
CarParkingSpaces
DisabledAccessInd – disabled access indicator
RefurbReqdCode – refurbishment required code referencing pisces table code_refurb_reqd
RefurbComments – comments related to the refurbishment
SensitiveConfInd – sensitive vacant space record ???

Deleted simple types

RecordStateType

Amended types

The simpleType DataDescriptionType changed. MaxLength changed to 255.

```
<xsd:simpleType name="DataDescriptionType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="255"/>
        <xsd:minLength value="1"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

CECAAddressStructure

Amended elements

```
<xsd:element name="PostCodeArea" type="AddressPostCodeAreaType" minOccurs="0"/>
<xsd:element name="PostCodeStreet" type="AddressPostCodeStreetType" minOccurs="0"/>
```

PropertyStructure

New elements

Buildings
Holdings

Deleted elements

CheckFileInd
RecordStateCode
PropertyNotes
PropertyChangedDate

Amended elements

```
<xsd:element name="OSNortherly" type="OSGridReferenceNorthType"/>
```

Building Structure

Deleted complexTypes

ConstructionType
RoofType
WindowType
WallType
BuildingElementDescriptorType

Deleted elements

SuperStructure
InternalFinish
OtherDetails
LiftInd
CentralHeatingInd
AirCondInd
RecordStateCode
ConstructionTypeCode
RoofTypeCode
WindowTypeCode
Externals
Remarks
FloorLoading
HaxMaterialInd
DisabledAccessInd
DisabledToiletsInd
CarParkingInd
BuildingNotes
BuildingChangedDate

HoldingStructure

New elements

MainLease
VacantSpace
Occupations

Deleted elements

DisposalCompDate
PRSRentRate
CheckfileInd
RecordStateCode
SettingUpReport
HoldingNotes
HoldingChangedDate

LeaseStructure

New simpleTypes

BreakReviewIndType
BreakOptPartyType

New *complexTypes*

BreakReviewStructure

Data structures are required for the definition of a break/review. This will be implemented as a complex type definition in CECADataTypes schema.
Break/review type will be implemented as a list of elements (zero or many) applied to LeaseStructure architecture schema.

The required elements are:

BreakReviewID – internal reference for a break/review reord as specified within OGC data structures.
BreakReviewInd – type of break/review record referencing pisces table code_break_review_ind
BreakReviewDate – earliest date of break/review
LatestBreakDate – last date of break/review
BreakNoticePerdTerm – notice period value
TermPerdCode – notice period code referencing pisces table code_term_perd
BreakReviewCompDate – completion date
BreakOptPartyCode – break option code referencing pisces table code_break_opt_party

New elements

Landlord
BreakReviews

Deleted elements

LandlordCode
RestrictCovInd
PreemptRightsInd
AlienationPermInd
RestrictUseInd
TenantImprovInd
LandlordTenantActInd
LandlordConsentRqdInd
CheckFileInd
SurrenderDate
LeaseNotes
LeaseChangedDate

OccupationStructure

New elements

Tenant
Lease
VacantSpace

Deleted elements

VacantArea
TenantCode
RecordStateCode

CECAProperty

New elements
MessageSignature

Deleted elements
Buildings
EffectiveDate
ChangedDate
AmendmentSource
AmendmentType

CECABuilding

New elements
MessageSignature

Deleted elements
EffectiveDate
ChangedDate
AmendmentSource
AmendmentType

CECAHolding

New elements
MessageSignature

Deleted elements
MainLease
Occupations
EffectiveDate
ChangedDate
AmendmentSource
AmendmentType

CECALease

New elements
MessageSignature

Deleted elements
EffectiveDate
ChangedDate
AmendmentSource
AmendmentType

CECAOccupation

New elements
MessageSignature

Deleted elements
Lease
EffectiveDate
ChangedDate
AmendmentSource
AmendmentType

Other changes

CECA Addressing and BS7666

The specification currently implements an architecture schema CECAAddressStructure which defines a composite structure for a combined CECA Address(semi structured UK address format) and BS7666 address format. Duue to onging discussions regarding address formats and the Government data standards catalog, a decision has been made to implement a "frozen" bs7666 version 3 schema. Due to the modular nature of the address implementation we could use the government data standards address proposal which is under consideration at present. This is dependant upon timescales, so options are open at present but only for the duration of the next RFC process.

XML Namespacing and import declarations

Due to inadequacies with the govtalk site there is no formal physical structure for schema storage. This makes the implementation of UNC based namespaces and import declarations difficult. Version 1.0 of the specification has local file path UNC declarations which need to be changed to reflect the e-Gif standards. Discussions with govtalk have implied that a "repository" could be setup to accommodate the schemas, but at present this has not been done.

As a contingency the schemas have been published to the Property and Construction directorate website (http://www.property.gov.uk/schemas/ceca/cecaxml.htm). The schemas will be located at *http://www.property.gov.uk/schemas/ceca/xml* and namespace and import declarations will be updated.

An outstanding issue still relates to BS7666 (see above) and the related reference path. Office of e-Envoy have indicated that this will be sorted out via the schema being published to a repository for reference.

CECA XML Message Schema

Implement a simple schema to allow the recording of XML document processing messages from CECAXMLServices component. Schema to include the following elements.

Date/Time
Error code (optional)
Object Type identifier (E, P,B,H,L,O)
OGC object identifier (optional)
Departmental identifier (optional)
ElementName (optional)
ElementValue (optional)
Message

Appendix B: Required updates upon approval

B.1 bs7666

Currently in the CECAAddressStructure schema, the bs7666 schema is being referenced as follows:

```
<xsd:schema targetNamespace="http://www.property.gov.uk" xmlns="http://www.property.gov.uk"
xmlns:bs7666="http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0" id="CECAAddressStructure">

        <xsd:import namespace="http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd" schemaLocation="
http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd "/>
```

This reference is based on a draft version of the BS7666 schemas which has not yet been published to a central schema repository. A version of the schema has been derived from the document *UK Online – Information Architecture – BS7666 Address and Geographic Location Structures Fragment version 0.1c* and dated 17th July 2001.

This version has been published to a temporary repository at http://www.property.gov.uk/schemas/bs7666/BS7666Schema.xsd until the schema has passed through RFC and is made available through Govtalk.

This issue should be resolved before the CECA schemas are finalised, but is not an important issue for now.

B.2 CECA schemas and xsd:include

At present, all of the `<xsd:include …/>` statements and namespaces declarations are using www.property.gov.uk due to non existence of a central repository within Govtalk for the storage of schemas. Upon publications and if a central repository for schemas is set up then this will be amended.