

# Schema documentation for pfc\_scenario.xsd

october 23, 2020

## Table of Contents

Namespace: "com:ara:pfc:training:1" .....	4
Schema(s) .....	4
Main schema pfc_scenario.xsd .....	4
Included schema pfc_scenario_complex_types.xsd .....	4
Element(s) .....	4
Element Scenario .....	4
Complex Type(s) .....	5
Complex Type ScenarioSchema .....	5
Complex Type summary .....	7
Complex Type author .....	8
Complex Type equipment-definition-list .....	9
Complex Type equipment .....	9
Complex Type citation-ref-list .....	10
Complex Type equipment-properties-list .....	10
Complex Type equipment-property .....	11
Complex Type property-field-list .....	11
Complex Type field-type .....	12
Complex Type trauma-definition-list .....	12
Complex Type trauma .....	12
Complex Type trauma-severity-range .....	13
Complex Type numeric-range .....	14
Complex Type token-range .....	14
Complex Type treatment-plan-definition-list .....	15
Complex Type treatment-plan .....	15
Complex Type equipment-ref-list .....	16
Complex Type medical-reference-list .....	16
Complex Type cpg-ref-list .....	17
Complex Type trauma-profile-list .....	17
Complex Type trauma-profile .....	17
Complex Type trauma-occurrence-list .....	18
Complex Type trauma-occurrence .....	18
Complex Type treatment-plan-ref-list .....	19
Complex Type medical-scenario-syllabus .....	20
Complex Type learning-objective-list .....	20
Complex Type learning-objective .....	20
Complex Type trauma-profile-ref-list .....	21
Complex Type assessment-list .....	21
Complex Type assessment .....	22
Complex Type medical-scenario .....	22
Complex Type role-list .....	23
Complex Type role .....	24
Complex Type scene-list .....	24
Complex Type scene .....	25
Complex Type event-list .....	26
Complex Type event .....	26
Complex Type item-list .....	27
Complex Type item .....	28
Complex Type property-value-list .....	28
Complex Type property-value .....	29
Complex Type role-ref-list .....	29
Complex Type citation-list .....	29
Complex Type citation .....	30
Complex Type cpg-list .....	30
Complex Type cpg .....	31
Complex Type learning-objective-reference-list .....	31
Simple Type(s) .....	32
Simple Type event-category-enum .....	32
Simple Type event-fidelity-enum .....	32
Namespace: "" .....	33
Element(s) .....	33
Element ScenarioSchema / summary .....	33
Element summary / title .....	34
Element summary / description .....	34

Element summary / version .....	34
Element summary / classification .....	35
Element summary / keywords .....	35
Element summary / domain .....	35
Element summary / limitations .....	35
Element ScenarioSchema / author .....	36
Element author / id .....	36
Element author / first_name .....	36
Element author / last_name .....	37
Element author / organization .....	37
Element author / phone_number .....	37
Element author / email .....	37
Element author / zip .....	38
Element author / state .....	38
Element author / country .....	38
Element ScenarioSchema / equipment .....	38
Element equipment-definition-list / equipment .....	39
Element equipment / id .....	39
Element equipment / name .....	39
Element equipment / type .....	40
Element equipment / description .....	40
Element equipment / citations .....	40
Element citation-ref-list / citation-ref .....	40
Element equipment / image .....	41
Element equipment / properties .....	41
Element equipment-properties-list / property .....	41
Element equipment-property / name .....	42
Element equipment-property / type .....	42
Element equipment-property / fields .....	42
Element property-field-list / field .....	42
Element field-type / name .....	43
Element field-type / type .....	43
Element ScenarioSchema / trauma-definitions .....	43
Element trauma-definition-list / trauma .....	44
Element trauma / id .....	44
Element trauma / common-name .....	44
Element trauma / medical-name .....	44
Element trauma / citations .....	45
Element trauma / description .....	45
Element trauma / severity-range .....	45
Element trauma-severity-range / numeric-range .....	46
Element numeric-range / lower_bound .....	46
Element numeric-range / upper_bound .....	46
Element numeric-range / median .....	46
Element numeric-range / unit .....	47
Element trauma-severity-range / token-range .....	47
Element token-range / token .....	47
Element ScenarioSchema / treatment-plans .....	47
Element treatment-plan-definition-list / treatment-plan .....	48
Element treatment-plan / id .....	48
Element treatment-plan / common-name .....	49
Element treatment-plan / medical-name .....	49
Element treatment-plan / description .....	49
Element treatment-plan / required-equipment .....	49
Element equipment-ref-list / equipment-refs .....	50
Element treatment-plan / references .....	50
Element medical-reference-list / citations .....	50
Element medical-reference-list / cpgs .....	50
Element cpg-ref-list / cpg-ref .....	51
Element ScenarioSchema / trauma-sets .....	51
Element trauma-profile-list / trauma-profile .....	51
Element trauma-profile / id .....	52
Element trauma-profile / name .....	52
Element trauma-profile / physiology-state .....	52
Element trauma-profile / injuries .....	53
Element trauma-occurrence-list / trauma .....	53
Element trauma-occurrence / id .....	53
Element trauma-occurrence / location .....	54
Element trauma-occurrence / description .....	54
Element trauma-occurrence / severity .....	54
Element trauma-profile / treatments .....	54
Element treatment-plan-ref-list / treatment-plan .....	55

Element ScenarioSchema / syllabus .....	55
Element medical-scenario-syllabus / learning-objectives .....	55
Element learning-objective-list / objective .....	55
Element learning-objective / id .....	56
Element learning-objective / name .....	56
Element learning-objective / description .....	56
Element learning-objective / references .....	57
Element learning-objective / relates-to .....	57
Element learning-objective / relates-to / treatment-plans .....	57
Element learning-objective / relates-to / trauma-profiles .....	58
Element trauma-profile-ref-list / trauma-profile .....	58
Element medical-scenario-syllabus / learning-assessments .....	58
Element assessment-list / total-points .....	58
Element assessment-list / assessment .....	59
Element assessment / id .....	59
Element assessment / objective-id .....	59
Element assessment / name .....	60
Element assessment / description .....	60
Element assessment / points-available .....	60
Element assessment / criteria .....	60
Element ScenarioSchema / medical-scenario .....	60
Element medical-scenario / id .....	61
Element medical-scenario / roles .....	61
Element role-list / role .....	61
Element role / id .....	62
Element role / name .....	62
Element role / short-name .....	62
Element role / trauma-profile-ref .....	62
Element role / description .....	63
Element medical-scenario / training-script .....	63
Element scene-list / scene .....	63
Element scene / id .....	64
Element scene / location-id .....	64
Element scene / name .....	64
Element scene / description .....	64
Element scene / details .....	65
Element scene / weather .....	65
Element scene / time-of-day .....	65
Element scene / time-in-simulation .....	65
Element scene / events .....	65
Element event-list / event .....	66
Element event / id .....	66
Element event / name .....	67
Element event / category .....	67
Element event / fidelity .....	67
Element event / actor_1 .....	68
Element event / actor_2 .....	68
Element event / equipment .....	68
Element event / details .....	68
Element event / description .....	68
Element scene / items .....	68
Element item-list / item .....	69
Element item / short-name .....	69
Element item / equipment-id .....	69
Element item / description .....	70
Element item / properties .....	70
Element property-value-list / value .....	70
Element property-value / name .....	71
Element property-value / value .....	71
Element scene / roles .....	71
Element role-ref-list / role-ref .....	71
Element ScenarioSchema / works-cited .....	71
Element citation-list / citation .....	72
Element citation / uuid .....	72
Element citation / key .....	73
Element citation / authors .....	73
Element citation / title .....	73
Element citation / date .....	73
Element citation / page .....	74
Element citation / accessed .....	74
Element cpg-list / cpg .....	74
Element cpg / name .....	74

---

Element cpg / description .....	75
Element cpg / citation-ref .....	75
Element learning-objective-reference-list / objective .....	75

## Namespace: "com:ara:pfc:training:1"

### Schema(s)

#### Main schema pfc\_scenario.xsd

Namespace	com:ara:pfc:training:1
Properties	attribute form default: unqualified element form default: unqualified version: 0.3.0

#### Included schema pfc\_scenario\_complex\_types.xsd

Namespace	com:ara:pfc:training:1
Properties	attribute form default: unqualified element form default: unqualified version: 0.3

### Element(s)

#### Element Scenario

Namespace	com:ara:pfc:training:1

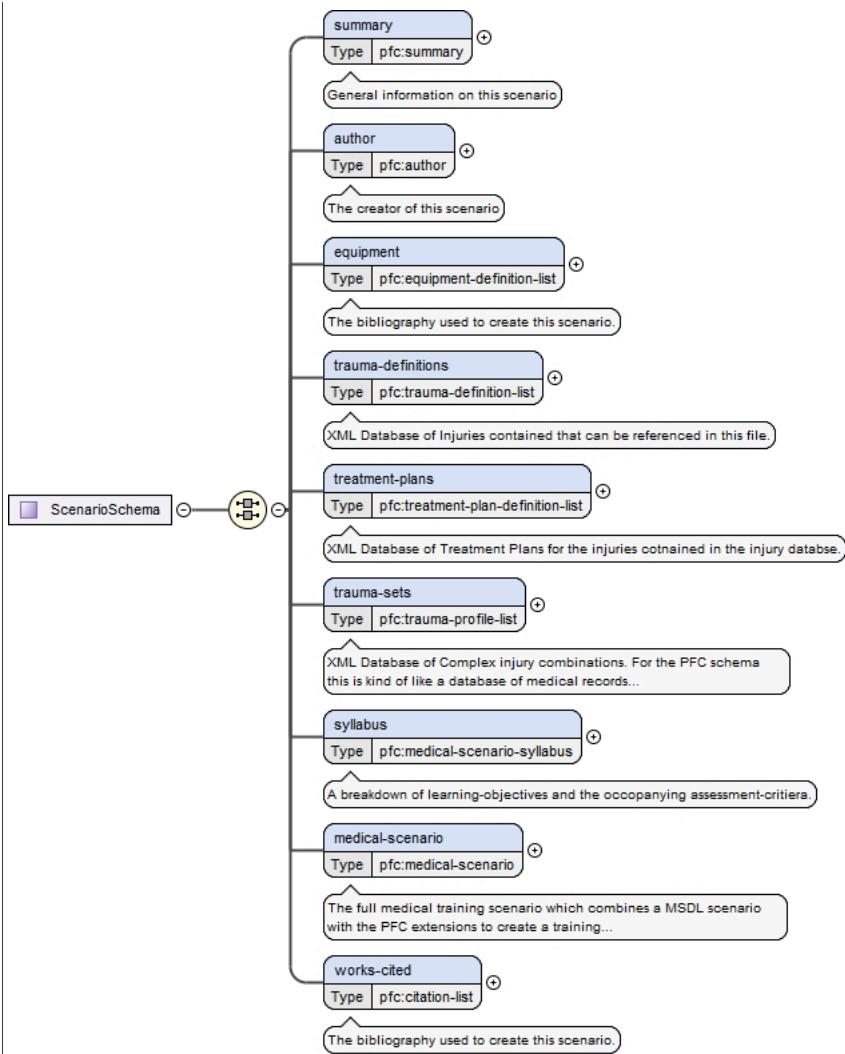
Diagram	<pre> classDiagram     class ScenarioSchema {         summary         author         equipment         trauma-definitions         treatment-plans         trauma-sets         syllabus         medical-scenario         works-cited     }     class Scenario {         &lt; -- ScenarioSchema     }     Scenario &lt; -- ScenarioSchema   </pre> <p>The diagram illustrates the structure of the <code>ScenarioSchema</code> element. It is defined as a complex type (<code>complex</code>) and contains several child elements: <code>summary</code>, <code>author</code>, <code>equipment</code>, <code>trauma-definitions</code>, <code>treatment-plans</code>, <code>trauma-sets</code>, <code>syllabus</code>, <code>medical-scenario</code>, and <code>works-cited</code>. Each child element is associated with a specific type: <code>pfc:summary</code>, <code>pfc:author</code>, <code>pfc:equipment-definition-list</code>, <code>pfc:trauma-definition-list</code>, <code>pfc:treatment-plan-definition-list</code>, <code>pfc:trauma-profile-list</code>, <code>pfc:medical-scenario-syllabus</code>, <code>pfc:medical-scenario</code>, and <code>pfc:citation-list</code> respectively. The <code>Scenario</code> element is shown to inherit from <code>ScenarioSchema</code>.</p>
Type	ScenarioSchema
Properties	content: complex
Model	ALL(summary author equipment trauma-definitions treatment-plans trauma-sets syllabus medical-scenario works-cited)
Children	author, equipment, medical-scenario, summary, syllabus, trauma-definitions, trauma-sets, treatment-plans, works-cited
Instance	<pre> &lt;Scenario xmlns="com:ara:pfc:training:1"&gt;   &lt;summary&gt;{1,1}&lt;/summary&gt;   &lt;author&gt;{1,1}&lt;/author&gt;   &lt;equipment&gt;{1,1}&lt;/equipment&gt;   &lt;trauma-definitions&gt;{1,1}&lt;/trauma-definitions&gt;   &lt;treatment-plans&gt;{1,1}&lt;/treatment-plans&gt;   &lt;trauma-sets&gt;{1,1}&lt;/trauma-sets&gt;   &lt;syllabus&gt;{1,1}&lt;/syllabus&gt;   &lt;medical-scenario&gt;{1,1}&lt;/medical-scenario&gt;   &lt;works-cited&gt;{1,1}&lt;/works-cited&gt; &lt;/Scenario&gt;   </pre>
Source	<code>&lt;xss:element name="Scenario" type="ScenarioSchema" /&gt;</code>

## Complex Type(s)

### Complex Type Scenarioschema

Namespace	com:ara:pfc:training:1
-----------	------------------------

## Diagram



## Used by

Element Scenario

## Model

ALL(summary author equipment trauma-definitions treatment-plans trauma-sets syllabus medical-scenario works-cited)

## Children

author, equipment, medical-scenario, summary, syllabus, trauma-definitions, trauma-sets, treatment-plans, works-cited

## Source

```

<xs:complexType name="ScenarioSchema">
  <xs:all>
    <xs:element name="summary" type="pfc:summary">
      <xs:annotation>
        <xs:documentation>General information on this scenario</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="author" type="pfc:author">
      <xs:annotation>
        <xs:documentation>The creator of this scenario</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="equipment" type="pfc:equipment-definition-list">
      <xs:annotation>
        <xs:documentation>The bibliography used to create this scenario.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="trauma-definitions" type="pfc:trauma-definition-list">
      <xs:annotation>
        <xs:documentation>XML Database of Injuries contained that can be referenced in this file.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="treatment-plans" type="pfc:treatment-plan-definition-list">
      <xs:annotation>
        <xs:documentation>XML Database of Treatment Plans for the injuries cotnained in the injury database.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="trauma-sets" type="pfc:trauma-profile-list">
      <xs:annotation>
        <xs:documentation>XML Database of Complex injury combinations. For the PFC schema this is kind of like a database of medical records...</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="syllabus" type="pfc:medical-scenario-syllabus">
      <xs:annotation>
        <xs:documentation>A breakdown of learning-objectives and the occopanying assessment-criteria.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="medical-scenario" type="pfc:medical-scenario">
      <xs:annotation>
        <xs:documentation>The full medical training scenario which combines a MSDL scenario with the PFC extensions to create a training...</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="works-cited" type="pfc:citation-list">
      <xs:annotation>
        <xs:documentation>The bibliography used to create this scenario.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:all>
</xs:complexType>

```

```

<xs:element name="trauma-sets" type="pfc:trauma-profile-list">
  <xs:annotation>
    <xs:documentation>XML Database of Complex injury combinations. For the PFC schema this is kind of like a database of medical records which can be applied to patients. It allows the same set of injuries to be applied to multiple patients with minimal documentation.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="syllabus" type="pfc:medical-scenario-syllabus">
  <xs:annotation>
    <xs:documentation>A breakdown of learning-objectives and the accompanying assessment-criteria.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="medical-scenario" type="pfc:medical-scenario">
  <xs:annotation>
    <xs:documentation>The full medical training scenario which combines a MSDL scenario with the PFC extensions to create a training narrative which accomplishes the learning objectives described in the syllabus.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="works-cited" type="pfc:citation-list">
  <xs:annotation>
    <xs:documentation>The bibliography used to create this scenario.</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:all>
</xs:complexType>

```

## Complex Type summary

Namespace	com:ara:pfc:training:1
Annotations	<p>Summary stores all general information about the current scenario file.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>Title: xs:string : The name of the scenario contained in the file</li> <li>Description: xs:string : A brief summary of the contents of the scenario and its purpose. Can be multiple paragraphs in format.</li> <li>Version: xs:string : Internal version number of the current scenario file. Ideally this would be maintained over multiple official releases.</li> <li>Classification: xs:string : Intended as a way of marking the classified nature of the contents contained in the scenario file. (e.g Non-Classified, Confidential, Secret, Top Secret, ...)</li> <li>Keywords: xs:string : Set of comma delimited keywords that can be used for organizing multiple scenario files into organized sets.</li> <li>Domain: xs:string : The area of study which is most relevant to the the training scenario</li> <li>Limitations: xs:string : Any known limitations in the scenario with regards to its use.</li> </ul> <p>Example:</p> <p>Title: Gun Shot Wound Treatment    Description: Simple emergency training on treatment of clean GSW wounds.    Version: MIL-GSW-2019    Classification: NON-CLASSIFIED    Keywords: First Responder, Puncture Wounds, Gun Shot    Domain: Emergency Response, Trauma Ward    Limitations: Not Suited for resuscitation exercises.</p>
Diagram	<pre> classDiagram     class summary {         +title         +description         +version         +classification         +keywords         +domain         +limitations     }     note left of summary: Summary stores all general information about the current scenario file. Fields: Title: xs:string : The name of the scenario contained in the file Description: xs:string : A brief summary of the contents of the scenario and its purpose. Can be multiple paragraphs in format. </pre>
Used by	Element ScenarioSchema/summary
Model	ALL(title{0,1} description{0,1} version{0,1} classification{0,1} keywords{0,1} domain{0,1} limitations{0,1})
Children	classification, description, domain, keywords, limitations, title, version
Source	<pre> &lt;xs:complexType name="summary"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Summary stores all general information about the current scenario file. Fields: Title: xs:string : The name of the scenario contained in the file Description: xs:string : A brief summary of the contents of the scenario and its purpose. Can be multiple paragraphs in   &lt;/xs:documentation&gt; </pre>

```

format. Version: xs:string : Internal version number of the current scenario file. Ideally this would be maintained over multiple official releases. Classification: xs:string : Intended as away of marking the classified nature of the contents contained in the scenario file. (e.g Non-Classified, Confidential, Secret, Top Secret, ...) Keywords: xs:string : Set of comma delimited keywords that can be used for organizing multiple scenario files in to organized sets. Domain: xs:string : The area of study which is most relevant to the the training scenario Limitations: xs:string : Any known limitations in the scenario with regards to its use. Example: Title: Gun Shot Wound Treatment Description: Simple emergency training on treatment of clean GSW wounds. Version: MIL-GSW-2019 Classification: NON-CLASSIFIED Keywords: First Responder, Puncture Wounds, Gun Shot Domain: Emergency Response, Trauma Ward Limitations: Not Suited for resuscitation exercises.</xs:documentation>
</xs:annotation>
<xs:all>
<xs:element name="title" type="xs:string" minOccurs="0"/>
<xs:element name="description" type="xs:string" minOccurs="0"/>
<xs:element name="version" type="xs:string" minOccurs="0"/>
<xs:element name="classification" type="xs:string" minOccurs="0"/>
<xs:element name="keywords" type="xs:string" minOccurs="0"/>
<xs:element name="domain" type="xs:string" minOccurs="0"/>
<xs:element name="limitations" type="xs:string" minOccurs="0"/>
</xs:all>
</xs:complexType>

```

## Complex Type author

Namespace	com:ara:pfc:training:1
Annotations	<p>Important data regarding a scenario's creator.</p> <p>Fields:</p> <p>First Name: xs:string : The author's first name.      Last Name: xs:string : The author's last name.      Organization: xs:string : The organization that the author belongs to.      Phone Number: xs:string : Preferred phone number for contact.      Email: xs:string : Preferred email for contact.      ZIP: xs:string : Author/organization's ZIP code.      State: xs:string : Author/organization's state of residence.      Country: xs:string : Author/organization's country of residence.</p> <p>Example:</p> <p>First Name: John      Last Name: Smith      Organization: ARA      Phone Number: (123)456-7890      Email: johnsmith@ara.com      ZIP: 12345      State: North Carolina      Country: USA</p>
Diagram	<p>Important data regarding a scenario's creator. Fields: First Name: xs:string : The author's first name. Last Name: xs:string : The author's last name. Organization: xs:string : The organization that the author belongs to. Phone Number: xs:string : Preferred phone number for contact. Email: xs:string : Preferred email for contact. ZIP: xs:string : Author/organization's ZIP code. State: xs:string : Author/organization's state of residence. Country: xs:string : Author/organization's country of residence. Example: First Name: John Last Name: Smith Organization: ARA</p>
Used by	Element ScenarioSchema/author
Model	ALL(id{0,1} first_name{0,1} last_name{0,1} organization{0,1} phone_number{0,1} email{0,1} zip{0,1} state{0,1} country{0,1})
Children	country, email, first_name, id, last_name, organization, phone_number, state, zip
Source	<pre> &lt;xs:complexType name="author"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Important data regarding a scenario's creator. Fields: First Name: xs:string : The author's first name. Last Name: xs:string : The author's last name. Organization: xs:string : The organization that the author belongs to. Phone Number: xs:string : Preferred phone number for contact. Email: xs:string : Preferred email for contact. ZIP: xs:string : Author/organization's ZIP code. State: xs:string : Author/organization's state of residence. Country: xs:string : Author/organization's country of residence. Example: First Name: John Last Name: Smith Organization: ARA     &lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:complexType&gt; </pre>

```

Phone Number: (123)456-7890 Email: johnsmith@ara.com ZIP: 12345 State: North Carolina Country:
USA</xs:documentation>
</xs:annotation>
<xs:all>
<xs:element name="id" type="xs:string" minOccurs="0"/>
<xs:element name="first_name" type="xs:string" minOccurs="0"/>
<xs:element name="last_name" type="xs:string" minOccurs="0"/>
<xs:element name="organization" type="xs:string" minOccurs="0"/>
<xs:element name="phone_number" type="xs:string" minOccurs="0"/>
<xs:element name="email" type="xs:string" minOccurs="0"/>
<xs:element name="zip" type="xs:string" minOccurs="0"/>
<xs:element name="state" type="xs:string" minOccurs="0"/>
<xs:element name="country" type="xs:string" minOccurs="0"/>
</xs:all>
</xs:complexType>

```

## Complex Type equipment-definition-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:equipment types.
Diagram	<pre> graph LR     A[equipment-definition-list] -- "0..∞" --&gt; B[equipment]     subgraph Callout [List of pfc:equipment types.]         B     end </pre>
Used by	Element ScenarioSchema/equipment
Model	equipment*
Children	equipment
Source	<pre> &lt;xs:complexType name="equipment-definition-list"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;List of pfc:equipment types.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:sequence&gt;         &lt;xs:element name="equipment" type="equipment" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type equipment

Namespace	com:ara:pfc:training:1
Annotations	<p>A PFC representation of any equipment or inventory that might be used in the scenario. This could be a medical device or a any common item which is of significance to the scenario narrative.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Name: xs:string : Name of the equipment</li> <li>Type: xs:integer :</li> <li>Description: xs:string : Brief summary of the equipment</li> <li>Citations: citation-ref-list : List of reference material for additional information</li> <li>Image: xs:string : Visual reference</li> <li>Properties : property-list : Name/Value Pair of Implementation Details</li> <li>Example:</li> <li>ID: 0001</li> <li>Name: Syringe</li> <li>Type: Medical Device</li> <li>Description: A tube with a nozzle and piston or bulb for sucking in and ejecting liquid in a thin stream, used for cleaning wounds or body cavities, or fitted with a hollow needle for injecting or withdrawing fluids.</li> <li>Citations: <a href="https://en.wikipedia.org/wiki/Syringe">https://en.wikipedia.org/wiki/Syringe</a></li> <li>Image:</li> </ul>
Diagram	<pre> graph LR     A[equipment] -- "0..∞" --&gt; B[id]     B -- "+" --&gt; C[name]     C -- "+" --&gt; D[type]     D -- "+" --&gt; E[description]     E -- "+" --&gt; F[citations]     F -- "+" --&gt; G[image]     G -- "+" --&gt; H[properties]     subgraph Callout [A PFC representation of any equipment or inventory that might be used in the scenario. This could be a medical device...]         A     end </pre>

Used by	Element equipment-definition-list/equipment
Model	ALL(id name type{0,1} description citations image{0,1} properties)
Children	citations, description, id, image, name, properties, type
Source	<pre> &lt;xss:complexType name="equipment"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;A PFC representation of any equipment or inventory that might be used in the scenario. This could be a medical device or a any common item which is of significance to the scenario narrative. Fields: ID: xs:string : Identification Name: xs:string : Name of the equipment Type: xs:integer : Description: xs:string : Brief summary of the equipment Citations: citation-ref-list : List of reference material for additional information Image: xs:string : Visual reference Properties : property-list : Name/Value Pair of Implementation Details Example: ID: 0001 Name: Syringe Type: Medical Device Description: A tube with a nozzle and piston or bulb for sucking in and ejecting liquid in a thin stream, used for cleaning wounds or body cavities, or fitted with a hollow needle for injecting or withdrawing fluids. Citations: https://en.wikipedia.org/wiki/Syringe Image:&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:all&gt;     &lt;xss:element name="id" type="xs:string"/&gt;     &lt;xss:element name="name" type="xs:string" minOccurs="1"/&gt;     &lt;xss:element name="type" type="xs:integer" minOccurs="0"/&gt;     &lt;xss:element name="description" type="xs:string" minOccurs="1"/&gt;     &lt;xss:element name="citations" type="citation-ref-list" minOccurs="1"/&gt;     &lt;xss:element name="image" type="xs:string" minOccurs="0"/&gt;     &lt;xss:element name="properties" type="equipment-properties-list" minOccurs="1"/&gt;   &lt;/xss:all&gt; &lt;/xss:complexType&gt;</pre>

## Complex Type citation-ref-list

Namespace	com:ara:pfc:training:1
Annotations	List of strings which should be citations for source material. This could be expanded to be a more stringent field combination, but lets see what happens
Diagram	<pre> sequenceDiagram     participant A as citation-ref-list     participant B as citation-ref     A-&gt;&gt;B: 0..∞   </pre> <p>A UML sequence diagram illustrating the relationship between 'citation-ref-list' and 'citation-ref'. A lifeline labeled 'citation-ref-list' sends a message to a lifeline labeled 'citation-ref' 0..∞ times. A callout box above the lifeline 'citation-ref-list' contains the annotation: 'List of strings which should be citations for source material. This could be expanded to be a more stringent field...'.</p>
Used by	Elements equipment/citations, medical-reference-list/citations, trauma/citations
Model	citation-ref*
Children	citation-ref
Source	<pre> &lt;xss:complexType name="citation-ref-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of strings which should be citations for source material. This could be expanded to be a more stringent field combination, but lets see what happens&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:sequence&gt;     &lt;xss:element name="citation-ref" type="xs:string" minOccurs="0" maxOccurs="unbounded" /&gt;   &lt;/xss:sequence&gt; &lt;/xss:complexType&gt;</pre>

## Complex Type equipment-properties-list

Namespace	com:ara:pfc:training:1
Annotations	List of Citation Information that a citation-ref-list refers to
Diagram	<pre> sequenceDiagram     participant A as equipment-properties-list     participant B as property     A-&gt;&gt;B: 0..∞   </pre> <p>A UML sequence diagram illustrating the relationship between 'equipment-properties-list' and 'property'. A lifeline labeled 'equipment-properties-list' sends a message to a lifeline labeled 'property' 0..∞ times. A callout box above the lifeline 'equipment-properties-list' contains the annotation: 'List of Citation Information that a citation-ref-list refers to'.</p>
Used by	Element equipment/properties
Model	property*
Children	property
Source	<pre> &lt;xss:complexType name="equipment-properties-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of Citation Information that a citation-ref-list refers to&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;</pre>

```

</xs:annotation>
<xs:sequence>
  <xs:element name="property" type="equipment-property" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>

```

## Complex Type equipment-property

Namespace	com:ara:pfc:training:1
Annotations	<p>A property is a configurable value of a piece of equipment. Each equipment can have any number of properties</p> <p>The specific values of each property will be set when it is assigned to medical-scenario as a prop in a specific scen</p> <p>Properties consist of a name,type pair with one or more fields.</p> <p>A field is a name,type pair an item in a scene should have corrisponding property-value list which contains the instance values for all fields of the type.</p> <p>The default field is implied by the property type so a Scalar representing volume would have no Scalar field but would have unit field of type string.</p> <p>Future extensions of this spec will aim to standardize field types</p>
Diagram	
Used by	Element equipment-properties-list/property
Model	name , type , fields
Children	fields, name, type
Source	<pre> &lt;xs:complexType name="equipment-property"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A property is a configurable value of a piece of equipment. Each equipment can have any number of properties The specific values of each property will be set when it is assigned to medical-scenario as a prop in a specific scen Properties consist of a name,type pair with one or more fields. A field is a name,type pair an item in a scene should have corrisponding property-value list which contains the instance values for all fields of the type. The default field is implied by the property type so a Scalar representing volume would have no Scalar field but would have unit field of type string. Future extensions of this spec will aim to standardize field types&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="name" type="xs:string"/&gt;     &lt;xs:element name="type" type="xs:string"/&gt;     &lt;xs:element name="fields" type="property-field-list"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type property-field-list

Namespace	com:ara:pfc:training:1
Annotations	List of property fields for defining the properties of equipment in the syllabus
Diagram	
Used by	Element equipment-property/fields
Model	field*
Children	field
Source	<pre> &lt;xs:complexType name="property-field-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of property fields for defining the properties of equipment in the syllabus&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="field" type="field-type" minOccurs="0" maxOccurs="unbounded" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

<pre>&lt;/xs:complexType&gt;</pre>
------------------------------------

## Complex Type field-type

Namespace	com:ara:pfc:training:1
Annotations	The value for a property. They are broken into two types because properties are generally defined in definition types while values appear in instance types;
Diagram	<p>The value for a property. They are broken into two types because properties are generally defined in definition types...</p>
Used by	Element property-field-list/field
Model	ALL(name type)
Children	name, type
Source	<pre>&lt;xs:complexType name="field-type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The value for a property. They are broken into two types because properties are generally defined in definition types while values appear in instance types;&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="name" type="xs:string"/&gt;     &lt;xs:element name="type" type="xs:string"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type trauma-definition-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:trauma types.
Diagram	<p>List of pfc:trauma types.</p>
Used by	Element ScenarioSchema/trauma-definitions
Model	trauma*
Children	trauma
Source	<pre>&lt;xs:complexType name="trauma-definition-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of pfc:trauma types.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="trauma" type="trauma" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type trauma

Namespace	com:ara:pfc:training:1
Annotations	<p>A PFC representation of a simple trauma. Injuries are associated to a Injury-Profile where severity and trauma-site are applied.</p> <p>An trauma must have a medical-name, but can also have a common-name for use in documentation. It is good to provide a list of citations which describe the trauma in more traditional terms in the references section, but is not required.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Common Name: xs:string : Alternative name for trauma.</li> <li>Medical Name: xs:string : Formal name for trauma.</li> <li>Citations: citation-ref-list : List of reference material for additional information.</li> <li>Description: xs:string : Brief summary of the trauma in general.</li> <li>Severity Range: trauma-severity-range : Numerical range for mapping injury occurrence to a physiology engine.</li> </ul>

	<p>Example:</p> <p>ID: 0002</p> <p>Common Name: Cut</p> <p>Medical Name: Laceration wound</p> <p>Citations: <a href="https://en.wikipedia.org/wiki/Wound#Open">https://en.wikipedia.org/wiki/Wound#Open</a></p> <p>Description: A small laceration located on the patient's left calf.</p> <p>Severity Range: 4</p>
Diagram	<pre> classDiagram     class trauma {         id         common-name         medical-name         citations         description         severity-range     }     note over trauma: A PFC representation of a simple trauma. Injuries are associated to a Injury-Profile where severity and trauma-site are...   </pre>
Used by	Element      trauma-definition-list/trauma
Model	ALL(id common-name{0,1} medical-name citations description severity-range)
Children	citations, common-name, description, id, medical-name, severity-range
Source	<pre> &lt;xs:complexType name="trauma"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A PFC representation of a simple trauma. Injuries are associated to a Injury-Profile where severity and trauma-site are applied. An trauma must have a medical-name, but can also have a common-name for use in documentation. It is good to provide a list of citations which describe the trauma in more traditional terms in the references section, but is not required. Fields: ID: xs:string : Identification Common Name: xs:string : Alternative name for trauma. Medical Name: xs:string : Formal name for trauma. Citations: citation-ref-list : List of reference material for additional information. Description: xs:string : Brief summary of the trauma in general. Severity Range: trauma-severity-range : Numerical range for mapping injury occurrence to a physiology engine. Example: ID: 0002 Common Name: Cut Medical Name: Laceration wound Citations: https://en.wikipedia.org/wiki/Wound#Open Description: A small laceration located on the patient's left calf. Severity Range: 4&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="id" type="xs:string"/&gt;     &lt;xs:element name="common-name" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="medical-name" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="citations" type="citation-ref-list" minOccurs="1"/&gt;     &lt;xs:element name="description" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="severity-range" type="trauma-severity-range" minOccurs="1"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type trauma-severity-range

Namespace	com:ara:pfc:training:1
Annotations	Every trauma has a severity range which can either be presented as a numeric-range or a token range. Token ranges are a list of enums that describe the trauma, some are provided in the base XSD. numeric-ranges are either a scalar value or a unit-scalar which include a min and max value that represent some measure of scale.
Diagram	<pre> classDiagram     class trauma-severity-range {         numeric-range         token-range     }     note over trauma-severity-range: Every trauma has a severity range which can either be presented as a numeric-range or a token range. Token ranges are a...   </pre>
Used by	Element      trauma/severity-range
Model	numeric-range   token-range
Children	numeric-range, token-range
Source	<pre> &lt;xs:complexType name="trauma-severity-range"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Every trauma has a severity range which can either be presented as a numeric-range or a token range. Token ranges are a list of enums that describe the trauma, some are provided in the base XSD. numeric-ranges are either a scalar value or a unit-scalar which include a min and max value that represent some measure of scale.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:choice&gt;   </pre>

```

<xs:element name="numeric-range" type="numeric-range"/>
<xs:element name="token-range" type="token-range"/>
</xs:choice>
</xs:complexType>

```

## Complex Type numeric-range

Namespace	com:ara:pfc:training:1
Annotations	Numerical range of two doubles with an optional median value and unit type. The trauma definition mostly uses this for documentation. Scenario UIs may try to use it for visualization of trauma sets, but that is implementation defined. A physiology engine might use severity to set its own internal insults/injuries where valid values may be [0.0,1.0], [0.0, .5 l/s]
Diagram	<p>Numerical range of two doubles with an optional median value and unit type. The trauma definition mostly uses this for...</p>
Used by	Element trauma-severity-range/numeric-range
Model	ALL(lower_bound upper_bound median{0,1} unit{0,1})
Children	lower_bound, median, unit, upper_bound
Source	<pre> &lt;xs:complexType name="numeric-range"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Numerical range of two doubles with an optional median value and unit type. The trauma definition mostly uses this for documentation. Scenario UIs may try to use it for visualization of trauma sets, but that is implementation defined. A physiology engine might use severity to set its own internal insults/injuries where valid values may be [0.0,1.0], [0.0, .5 l/s]&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="lower_bound" type="xs:double" minOccurs="1"/&gt;     &lt;xs:element name="upper_bound" type="xs:double" minOccurs="1"/&gt;     &lt;xs:element name="median" type="xs:double" minOccurs="0"/&gt;     &lt;xs:element name="unit" type="xs:string" minOccurs="0"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type token-range

Namespace	com:ara:pfc:training:1
Annotations	Some ranges are better described with English words [low,medium,high] token-range allows a trauma definition to specify a custom enumeration that a UI would then convert to a valid dropdown list when constructing a UI. trauma-profiles can not validate against the trauma definition with out restricting the possible ranges so this is really for internal documentation.
Diagram	<p>Some ranges are better described with English words [low,medium,high] token-range allows a trauma definition to...</p>
Used by	Element trauma-severity-range/token-range
Model	token {2,unbounded}
Children	token
Source	<pre> &lt;xs:complexType name="token-range"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Some ranges are better described with English words [low,medium,high] token-range allows a trauma definition to specify a custom enumeration that a UI would then convert to a valid dropdown list when constructing a UI. trauma-profiles can not validate against the trauma definition with out restricting the possible ranges so this is really for internal documentation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt; </pre>

```

<xs:element name="token" type="xs:NCName" minOccurs="2" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>

```

## Complex Type treatment-plan-definition-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:treatment-plan types.
Diagram	<pre> sequenceDiagram     participant A as treatment-plan-definition-list     participant B as treatment-plan     A-&gt;&gt;B:      activate B     B--&gt;&gt;A: 1..∞     deactivate B </pre> <p>List of pfc:treatment-plan types.</p>
Used by	Element ScenarioSchema/treatment-plans
Model	treatment-plan+
Children	treatment-plan
Source	<pre> &lt;xs:complexType name="treatment-plan-definition-list"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;List of pfc:treatment-plan types.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:sequence&gt;         &lt;xs:element name="treatment-plan" type="treatment-plan" minOccurs="1" maxOccurs="unbounded" /&gt;     &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type treatment-plan

Namespace	com:ara:pfc:training:1
Annotations	<p>A Treatment Plan is intended to be a break down of a clinical practice guideline in to a series of discrete steps. The simplicity of the treatment plan breakdown is matched to the intended training for the scenario and not the true complexity of the full CPG.</p> <p>Future versions may allow logical branching but currently the representation is a linear flow of steps.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Common Name: xs:string : Alternative name for a treatment plan.</li> <li>Medical Name: xs:string : Formal name for a treatment plan.</li> <li>Description: xs:string : Brief summary of the treatment plan.</li> <li>Required Equipment: equipment-ref-list : List of the equipment needed to carry out the treatment plan.</li> <li>References: medical-reference-list : List of references used to create the treatment plan.</li> </ul> <p>Example:</p> <p>ID: 0003</p> <p>Common Name: Tylenol</p> <p>Medical Name: Acetaminophen</p> <p>Description: The drug acetaminophen was administered to the patient orally to reduce patient's fever.</p> <p>Required Equipment: One 500 mg Tylenol tablet</p> <p>References: <a href="https://en.wikipedia.org/wiki/Paracetamol">https://en.wikipedia.org/wiki/Paracetamol</a></p>
Diagram	<pre> sequenceDiagram     participant A as treatment-plan     participant B as      A-&gt;&gt;B:      activate B     B--&gt;&gt;A:      deactivate B </pre> <p>A Treatment Plan is intended to be a break down of a clinical practice guideline in to a series of discrete steps. The...</p>
Used by	Element treatment-plan-definition-list/treatment-plan
Model	ALL(id common-name{0,1} medical-name{0,1} description required-equipment references)
Children	common-name, description, id, medical-name, references, required-equipment
Source	<pre> &lt;xs:complexType name="treatment-plan"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;A Treatment Plan is intended to be a break down of a clinical practice guideline in to a series of discrete steps. The simplicity of the treatment plan breakdown is</pre>

matched to the intended training for the scenario and not the true complexity of the full CPG. Future versions may allow logical branching but currently the representation is a linear flow of steps. Fields: ID: xs:string : Identification Common Name: xs:string : Alternative name for a treatment plan. Medical Name: xs:string : Formal name for a treatment plan. Description: xs:string : Brief summary of the treatment plan. Required Equipment: equipment-ref-list : List of the equipment needed to carry out the treatment plan. References: medical-reference-list : List of references used to create the treatment plan. Example: ID: 0003 Common Name: Tylenol Medical Name: Acetaminophen Description: The drug acetaminophen was administered to the patient orally to reduce patient's fever. Required Equipment: One 500 mg Tylenol tablet References: <https://en.wikipedia.org/wiki/Paracetamol>

```

</xs:annotation>
<xs:all>
  <xs:element name="id" type="xs:string"/>
  <xs:element name="common-name" type="xs:string" minOccurs="0"/>
  <xs:element name="medical-name" type="xs:string" minOccurs="0"/>
  <xs:element name="description" type="xs:string" minOccurs="1"/>
  <xs:element name="required-equipment" type="equipment-ref-list" minOccurs="1"/>
  <xs:element name="references" type="medical-reference-list"/>
</xs:all>
</xs:complexType>
```

## Complex Type equipment-ref-list

Namespace	com:ara:pfc:training:1
Annotations	List of UUIDs from the equipment list in the FORCE_SIDE section of an associated MSDL file
Diagram	<p>List of UUIDs from the equipment list in the FORCE_SIDE section of an associated MSDL file</p>
Used by	Element treatment-plan/required-equipment
Model	equipment-refs*
Children	equipment-refs
Source	<pre> &lt;xs:complexType name="equipment-ref-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of UUIDs from the equipment list in the FORCE_SIDE section of an associated MSDL file&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="equipment-refs" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type medical-reference-list

Namespace	com:ara:pfc:training:1
Annotations	Series of citations that the treatment plan was built on. Allows the user to refer to more detailed material beyond the simplified training plan.
Diagram	<p>Series of citations that the treatment plan was built on. Allows the user to refer to more detailed material beyond the...</p>
Used by	Elements learning-objective/references, treatment-plan/references
Model	ALL(citations cpgs)
Children	citations, cpgs
Source	<pre> &lt;xs:complexType name="medical-reference-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Series of citations that the treatment plan was built on. Allows the user to refer to more detailed material beyond the simplified training plan.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="citations" type="citation-ref-list"/&gt;     &lt;xs:element name="cpgs" type="cpg-ref-list"/&gt;     &lt;!-- &lt;xs:element name="related-learning-objectives" type="learning-objective-reference-list" /&gt;   --&gt;</pre>

```

    </xs:all>
</xs:complexType>

```

## Complex Type cpg-ref-list

Namespace	com:ara:pfc:training:1
Annotations	List of CPG specific citations. This tag is differentiated from the citations tag in order to make it easier for user interfaces to parse out all associated CPGs and make mapping widgets.
Diagram	<p>The diagram shows a UML Class Diagram. A class named "cpg-ref-list" is connected via a directed association to another class named "cpg-ref". The multiplicity at the "cpg-ref-list" end is "0..infinity" and at the "cpg-ref" end is "+". A callout box points to the "cpg-ref-list" class with the text: "List of CPG specific citations. This tag is differentiated from the citations tag in order to make it easier for user...".</p>
Used by	Element medical-reference-list/cpgs
Model	cpg-ref*
Children	cpg-ref
Source	<pre> &lt;xs:complexType name="cpg-ref-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of CPG specific citations. This tag is differentiated from the citations tag in order to make it easier for user interfaces to parse out all associated CPGs and make mapping widgets.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="cpg-ref" type="xs:string" minOccurs="0" maxOccurs="unbounded" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type trauma-profile-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:trauma-profiles.
Diagram	<p>The diagram shows a UML Class Diagram. A class named "trauma-profile-list" is connected via a directed association to another class named "trauma-profile". The multiplicity at the "trauma-profile-list" end is "1..infinity" and at the "trauma-profile" end is "+". A callout box points to the "trauma-profile-list" class with the text: "List of pfc:trauma-profiles."</p>
Used by	Element ScenarioSchema/trauma-sets
Model	trauma-profile+
Children	trauma-profile
Source	<pre> &lt;xs:complexType name="trauma-profile-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of pfc:trauma-profiles.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="trauma-profile" type="trauma-profile" minOccurs="1" maxOccurs="unbounded" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type trauma-profile

Namespace	com:ara:pfc:training:1
Annotations	<p>A collection of Injuries which is intended to act as a boot strap medical record for an actor in a medical-scenario.</p> <p>Injury profiles can be appended either to an actor initially or mid scenario to act like a contracted wound. Injury in this case may include medical conditions or diagnosis, but will generally be a combat casualty wound.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Name: xs:string : Name of the collection of injuries.</li> <li>Physiology State: xs:string : The condition or state of the body or bodily functions.</li> <li>Injuries: trauma-occurrence-list : List of the injuries present in the profile.</li> <li>Treatments: treatment-plan-ref-list : Treatments that can be used to treat the trauma profile.</li> </ul>

	<p>Example: ID: 0004 Name: Blunt Force Trauma Physiology State: Awake and aware Injuries: Contusions, abrasions, lacerations, bone fractures, internal hemorrhages Treatments: Compression, disinfection, plaster cast, splint, intravenous injection</p>
Diagram	<pre> classDiagram     class trauma-profile {         id         name         physiology-state         injuries         treatments     }     class id     class name     class physiology-state     class injuries     class treatments     id &lt; -- trauma-profile     name &lt; -- trauma-profile     physiology-state &lt; -- trauma-profile     injuries &lt; -- trauma-profile     treatments &lt; -- trauma-profile   </pre> <p>A collection of Injuries which is intended to act as a boot strap medical record for an actor in a medical-scenario....</p>
Used by	Element      trauma-profile-list/trauma-profile
Model	id , name , physiology-state{0,1} , injuries , treatments
Children	id, injuries, name, physiology-state, treatments
Source	<pre> &lt;xs:complexType name="trauma-profile"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A collection of Injuries which is intended to act as a boot strap medical record for an actor in a medical-scenario. Injury profiles can be appended either to an actor initially or mid scenario to act like a contracted wound. Injury in this case may include medical conditions or diagnosis, but will generally be a combat casualty wound. Fields: ID: xs:string : Identification Name: xs:string : Name of the collection of injuries. Physiology State: xs:string : The condition or state of the body or bodily functions. Injuries: trauma-occurrence-list : List of the injuries present in the profile. Treatments: treatment-plan-ref-list : Treatments that can be used to treat the trauma profile. Example: ID: 0004 Name: Blunt Force Trauma Physiology State: Awake and aware Injuries: Contusions, abrasions, lacerations, bone fractures, internal hemorrhages Treatments: Compression, disinfection, plaster cast, splint, intravenous injection&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="id" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="name" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="physiology-state" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="injuries" type="trauma-occurrence-list" minOccurs="1"/&gt;     &lt;xs:element name="treatments" type="treatment-plan-ref-list"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type trauma-occurrence-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:trauma-occurrence
Diagram	<pre> classDiagram     class trauma-occurrence-list {         traumas     }     class traumas     traumas *-- traumas   </pre> <p>List of pfc:trauma-occurrence</p>
Used by	Element      trauma-profile/injuries
Model	trauma*
Children	trauma
Source	<pre> &lt;xs:complexType name="trauma-occurrence-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of pfc:trauma-occurrence&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="trauma" type="trauma-occurrence" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type trauma-occurrence

Namespace	com:ara:pfc:training:1
Annotations	Injury occurrence is an Injury-Reference with a few additional fields to help associate the trauma with a specific location in the body that it occurred and a severity of the condition. The severity is an abstract term which should be

	<p>annotated in the trauma description. It could be the flow rate for a hemorrhage or a general term like mild,severe for an infection</p> <p>Even first,second,third for a burn trauma. It is left to the implementation to interpret the meaning of the severity. Severity values are mapped to a severity definition by the implementation as xml validation alone can not enforce these rules.</p> <p>Fields:</p> <p>ID: xs:string : Identification Location: xs:string : Where the trauma is located on the body. Description: xs:string : Brief summary of the simple trauma. Severity: xs:string : Numeric value meant to be interpreted by a UI and then visualized in the scenario.</p> <p>Example: ID: 0005 Location: Right thigh Description: Gunshot wound Severity: 70</p>
--	---

Diagram	<pre> classDiagram     class trauma-occurrence {         id         location         description         severity     }     note over trauma-occurrence: Injury occurrence is an Injury-Reference with a few additional fields to help associate the trauma with a specific...   </pre>
Used by	Element trauma-occurrence-list/trauma
Model	ALL(id location description severity)
Children	description, id, location, severity
Source	<pre> &lt;xs:complexType name="trauma-occurrence"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Injury occurrence is an Injury-Reference with a few additional fields to help associate the trauma with a specific location in the body that it occurred and a severity of the condition. The severity is an abstract term which should be annotated in the trauma description. It could be the flow rate for a hemorrhage or a general term like mild,severe for an infection Even first,second,third for a burn trauma. It is left to the implementation to interpret the meaning of the severity. Severity values are mapped to a severity definition by the implementation as xml validation alone can not enforce these rules. Fields: ID: xs:string : Identification Location: xs:string : Where the trauma is located on the body. Description: xs:string : Brief summary of the simple trauma. Severity: xs:string : Numeric value meant to be interpreted by a UI and then visualized in the scenario. Example: ID: 0005 Location: Right thigh Description: Gunshot wound Severity: 70&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="id" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="location" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="description" type="xs:string" minOccurs="1"/&gt;     &lt;xs:element name="severity" type="xs:string" minOccurs="1"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type treatment-plan-ref-list

Namespace	com:ara:pfc:training:1
Annotations	List of UUIDs from the treatment-plans in the treatment-definition-list
Diagram	<pre> classDiagram     class treatment-plan-ref-list {         treatment-plan     }     note over treatment-plan-ref-list: List of UUUIDs from the treatment-plans in the treatment-definition-list   </pre>
Used by	Elements learning-objective/relates-to/treatment-plans, trauma-profile/treatments
Model	treatment-plan*
Children	treatment-plan
Source	<pre> &lt;xs:complexType name="treatment-plan-ref-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of UUUIDs from the treatment-plans in the treatment-definition-list&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="treatment-plan" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

<pre>&lt;/xs:complexType&gt;</pre>
------------------------------------

## Complex Type medical-scenario-syllabus

Namespace	com:ara:pfc:training:1
Diagram	<pre> classDiagram     class medical-scenario-syllabus {         &lt;&lt;complexType&gt;&gt;         &lt;&lt;learning-objectives&gt;&gt;         &lt;&lt;learning-assessments&gt;&gt;     }     medical-scenario-syllabus "1" -- "1" learning-objectives     medical-scenario-syllabus "1" -- "1" learning-assessments   </pre>
Used by	Element ScenarioSchema/syllabus
Model	learning-objectives , learning-assessments
Children	learning-assessments, learning-objectives
Source	<pre> &lt;xs:complexType name="medical-scenario-syllabus"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="learning-objectives" type="learning-objective-list"/&gt;     &lt;xs:element name="learning-assessments" type="assessment-list"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type learning-objective-list

Namespace	com:ara:pfc:training:1
Diagram	<pre> classDiagram     class learning-objective-list {         &lt;&lt;complexType&gt;&gt;         &lt;&lt;objective&gt;&gt;     }     learning-objective-list "1" -- "0..&gt;" objective   </pre>
Used by	Element medical-scenario-syllabus/learning-objectives
Model	objective*
Children	objective
Source	<pre> &lt;xs:complexType name="learning-objective-list"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="objective" type="learning-objective" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type learning-objective

Namespace	com:ara:pfc:training:1
Annotations	<p>Objectives will be entered as goals to be met upon completion of a scenario.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Name: xs:string : Name for the learning objective being created.</li> <li>Description: xs:string : Brief summary that defines the goal of the objective when taught through the scenario.</li> <li>References: medical-reference-list : List of references used to create the learning objective.</li> </ul> <p>Example:</p> <ul style="list-style-type: none"> <li>ID: 0006</li> <li>Name: Burn Wound PFC</li> <li>Description: Through the completion of the scenario, the player will be able to effectively treat a burn wound over a long period of time.</li> <li>References: <a href="https://www.mayoclinic.org/diseases-conditions/burns/diagnosis-treatment/drc-20370545">https://www.mayoclinic.org/diseases-conditions/burns/diagnosis-treatment/drc-20370545</a></li> </ul>
Diagram	<pre> classDiagram     class learning-objective {         &lt;&lt;complexType&gt;&gt;         &lt;&lt;id&gt;&gt;         &lt;&lt;name&gt;&gt;         &lt;&lt;description&gt;&gt;         &lt;&lt;references&gt;&gt;         &lt;&lt;relates-to&gt;&gt;     }     learning-objective "1" -- "1" id     learning-objective "1" -- "1" name     learning-objective "1" -- "1" description     learning-objective "1" -- "1" references     learning-objective "1" -- "1" relates-to   </pre> <p>Objectives will be entered as goals to be met upon completion of a scenario. Fields: ID: xs:string : Identification...</p>
Used by	Element learning-objective-list/objective
Model	ALL(id name description references relates-to)
Children	description, id, name, references, relates-to

Source	<pre> &lt;xss:complexType name="learning-objective"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Objectives will be entered as goals to be met upon completion of a scenario. Fields: ID: xs:string : Identification Name: xs:string : Name for the learning objective being created. Description: xs:string : Brief summary that defines the goal of the objective when taught through the scenario. References: medical-reference-list : List of references used to create the learning objective. Example: ID: 0006 Name: Burn Wound PFC Description: Through the completion of the scenario, the player will be able to effectively treat a burn wound over a long period of time. References: https://www.mayoclinic.org/diseases-conditions/burns/diagnosis-treatment/drc-20370545&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:all&gt;     &lt;xss:element name="id" type="xs:string"/&gt;     &lt;xss:element name="name" type="xs:string"/&gt;     &lt;xss:element name="description" type="xs:string"/&gt;     &lt;xss:element name="references" type="medical-reference-list"/&gt;     &lt;xss:element name="relates-to"&gt;       &lt;xss:complexType&gt;         &lt;xss:all&gt;           &lt;xss:element name="treatment-plans" type="treatment-plan-ref-list"/&gt;           &lt;xss:element name="trauma-profiles" type="trauma-profile-ref-list"/&gt;         &lt;/xss:all&gt;       &lt;/xss:complexType&gt;     &lt;/xss:element&gt;   &lt;/xss:all&gt; &lt;/xss:complexType&gt; </pre>
--------	---

## Complex Type trauma-profile-ref-list

Namespace	com:ara:pfc:training:1
Annotations	List of UUIDs from the trauma-profiles in the profile list
Diagram	<pre> classDiagram     class trauma-profile-ref-list     class trauma-profile     trauma-profile-ref-list "0..infinity" --&gt; "0..infinity" trauma-profile     </pre>
Used by	Element learning-objective/relates-to/trauma-profiles
Model	trauma-profile*
Children	trauma-profile
Source	<pre> &lt;xss:complexType name="trauma-profile-ref-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of UUIDs from the trauma-profiles in the profile list&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:sequence&gt;     &lt;xss:element name="trauma-profile" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xss:sequence&gt; &lt;/xss:complexType&gt; </pre>

## Complex Type assessment-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:assessment with a required total-points field which should equal the sum of all points-available in the assessment fields. Used as a sanity check by implementations.
Diagram	<pre> classDiagram     class assessment-list     class assessment     assessment-list "0..infinity" --&gt; "0..infinity" assessment     assessment-list "0..infinity" --&gt; "0..infinity" total-points     </pre>
Used by	Element medical-scenario-syllabus/learning-assessments
Model	total-points , assessment*
Children	assessment, total-points
Source	<pre> &lt;xss:complexType name="assessment-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of pfc:assessment with a required total-points field which should equal the sum of all points-available in the assessment fields. Used as a sanity check by implementations.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:sequence&gt;   &lt;/xss:sequence&gt; &lt;/xss:complexType&gt; </pre>

```

<xs:element name="total-points" default="0" type="xs:integer" minOccurs="1" maxOccurs="1"/>
<xs:element name="assessment" type="assessment" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

```

## Complex Type assessment

Namespace	com:ara:pfc:training:1
Annotations	<p>An assessment consists of an id and a idref to a lesson-objective. This type is a work in practice and the criteria element is likely to be reworked beyond a string in a future revision</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Name: xs:string : Name for the assessment.</li> <li>Description: xs:string : Brief summary of what the assessment evaluates.</li> <li>Points Available: xs:integer : How many total points are available for the player to earn through the scenario</li> <li>Criteria: xs:string : The standards at which a user will be evaluated by the assessment.</li> </ul> <p>Example:</p> <ul style="list-style-type: none"> <li>ID: 0007</li> <li>Name: Burn Wound Assessment</li> <li>Description: This assessment evaluates the user's capability to treat a burn wound over a long period of time.</li> <li>Points Available: 200</li> <li>Criteria: The player must take the correct steps in the correct sequence to treat the patient's burn for a period of 48 in-game hours.</li> </ul>
Diagram	<pre> classDiagram     class assessment {         id         objective-id         name         description         points-available         criteria     } </pre>
Used by	Element assessment-list/assessment
Model	ALL(id objective-id name description points-available criteria)
Children	criteria, description, id, name, objective-id, points-available
Source	<pre> &lt;xs:complexType name="assessment"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An assessment consists of an id and a idref to a lesson-objective. This type is a work in practice and the criteria element is likely to be reworked beyond a string in a future revision Fields: ID: xs:string : Identification Name: xs:string : Name for the assessment. Description: xs:string : Brief summary of what the assessment evaluates. Points Available: xs:integer : How many total points are available for the player to earn through the scenario Criteria: xs:string : The standards at which a user will be evaluated by the assessment. Example: ID: 0007 Name: Burn Wound Assessment Description: This assessment evaluates the user's capability to treat a burn wound over a long period of time. Points Available: 200 Criteria: The player must take the correct steps in the correct sequence to treat the patient's burn for a period of 48 in-game hours.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="id" type="xs:string"/&gt;     &lt;xs:element name="objective-id" type="xs:string"/&gt;     &lt;xs:element name="name" type="xs:string"/&gt;     &lt;xs:element name="description" type="xs:string"/&gt;     &lt;xs:element name="points-available" type="xs:integer"/&gt;     &lt;!--IF PARTIAL THEN HOW MANY TOTAL POINTS AVAILABLE--&gt;     &lt;xs:element name="criteria" type="xs:string"/&gt;     &lt;!--PASS/FAIL OR PARTIAL--&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type medical-scenario

Namespace	com:ara:pfc:training:1
Annotations	A PFC medical scenario is an extension of the MSDL. The purpose of this extension is to better document micro level sequence of events which are to complex to be described in the original MSDL definition.

	<p>The best metaphor for this extension is a stageplay script which requires a list of cast members, contains a description of the scenery and props to be used within the narrative along with a timeline of events.</p> <p>This metaphor maps to the list of requirements needed to accurately simulate a medical training scenario to provide the immersion needed to properly elevate the training and make learning transfer possible. The breakdown can be as simple or complex as is needed for a narrative to be completed. Various elements also have a fidelity field which is designed to allow additional information to be encoded in the scenario that is ignored by lower fidelity implementations.</p> <p><b>Fields:</b>  <b>ID:</b> xs:string : Identification  <b>Roles:</b> role-list : A list of the roles that are present in the scenario  <b>Training Script:</b> scene-list : A list of the scenes that are to be included in the scenario</p> <p><b>Example:</b>  <b>ID:</b> 0008  <b>Roles:</b> Doctor, burn wound patient, assistant  <b>Training Script:</b> Burn Wound PFC, Burn Wound Aftercare</p>
Diagram	<pre> classDiagram     class medical-scenario {         &lt;&lt;A PFC medical scenario is an extension of the MSDL. The purpose of this extension is to better document micro level sequence of events which are too complex to be described in the original MSDL definition. The best metaphor for this extension is a stageplay script which requires a list of cast members, contains a description of the scenery and props to be used within the narrative along with a timeline of events. This metaphor maps to the list of requirements needed to accurately simulate a medical training scenario to provide the immersion needed to properly elevate the training and make learning transfer possible. The breakdown can be as simple or complex as is needed for a narrative to be completed. Various elements also have a fidelity field which is designed to allow additional information to be encoded in the scenario that is ignored by lower fidelity implementations. Fields: ID: xs:string : Identification Roles: role-list : A list of the roles that are present in the scenario Training Script: scene-list : A list of the scenes that are to be included in the scenario Example: ID: 0008 Roles: Doctor, burn wound patient, assistant Training Script: Burn Wound PFC, Burn Wound Aftercare&gt;&gt;     }     class MSDL {         &lt;&lt;List of pfc:roles&gt;&gt;     }     medical-scenario "1" -- "0..1" id     medical-scenario "1" -- "0..1" roles     medical-scenario "1" -- "0..1" training-script   </pre>
Used by	Element ScenarioSchema/medical-scenario
Model	ALL(id roles training-script)
Children	id, roles, training-script
Source	<pre> &lt;xs:complexType name="medical-scenario"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A PFC medical scenario is an extension of the MSDL. The purpose of this extension is to better document micro level sequence of events which are to complex to be described in the original MSDL definition. The best metaphor for this extension is a stageplay script which requires a list of cast members, contains a description of the scenery and props to be used within the narrative along with a timeline of events. This metaphor maps to the list of requirements needed to accurately simulate a medical training scenario to provide the immersion needed to properly elevate the training and make learning transfer possible. The breakdown can be as simple or complex as is needed for a narrative to be completed. Various elements also have a fidelity field which is designed to allow additional information to be encoded in the scenario that is ignored by lower fidelity implementations. Fields: ID: xs:string : Identification Roles: role-list : A list of the roles that are present in the scenario Training Script: scene-list : A list of the scenes that are to be included in the scenario Example: ID: 0008 Roles: Doctor, burn wound patient, assistant Training Script: Burn Wound PFC, Burn Wound Aftercare&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="id" type="xs:string"/&gt;     &lt;xs:element name="roles" type="role-list"/&gt;     &lt;xs:element name="training-script" type="scene-list"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;   </pre>

## Complex Type role-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:roles
Diagram	<pre> classDiagram     class role-list {         &lt;&lt;List of pfc:roles&gt;&gt;     }     class MSDL {         &lt;&lt;List of pfc:roles&gt;&gt;     }     role-list "0..1" -- "0..&gt;" role   </pre>
Used by	Element medical-scenario/roles
Model	role*
Children	role
Source	<pre> &lt;xs:complexType name="role-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of pfc:roles&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="role" type="role" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

<pre>&lt;/xs:complexType&gt;</pre>
------------------------------------

## Complex Type role

Namespace	com:ara:pfc:training:1
Annotations	<p>A role is a reference to a pfc:role with an optional pfc:trauma-profile reference. An additional description describes the role of the unit in human terms on the level of a cliff-note. The short-name is a unique string to be use when parsing a training-script event to quickly refer to a role</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Name: xs:string : Name for the role</li> <li>Short Name: xs:string : Unique string used when parsing a training-script event to quickly refer to a role</li> <li>Description: xs:string : A brief summary explaining the role</li> </ul> <p>Example:</p> <pre>ID: 0009 Name: Doctor Short Name: 8cxQ5 Description: Doctor role that the player will fill to care for the wounded patient</pre>
Diagram	
Used by	Element      role-list/role
Model	ALL(id name short-name trauma-profile-ref{0,1} description)
Children	description, id, name, short-name, trauma-profile-ref
Source	<pre>&lt;xs:complexType name="role"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A role is a reference to a pfc:role with an optional pfc:trauma-profile reference. An additional description describes the role of the unit in human terms on the level of a cliff-note. The short-name is a unique string to be use when parsing a training-script event to quickly refer to a role Fields: ID: xs:string : Identification Name: xs:string : Name for the role Short Name: xs:string : Unique string used when parsing a training-script event to quickly refer to a role Description: xs:string : A brief summary explaining the role Example: ID: 0009 Name: Doctor Short Name: 8cxQ5 Description: Doctor role that the player will fill to care for the wounded patient&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="id" type="xs:string"/&gt;     &lt;xs:element name="name" type="xs:string"/&gt;     &lt;xs:element name="short-name" type="xs:string"/&gt;     &lt;xs:element name="trauma-profile-ref" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="description" type="xs:string"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type scene-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:scene(s)
Diagram	
Used by	Element      medical-scenario/training-script
Model	scene+
Children	scene

Source	<pre>&lt;xss:complexType name="scene-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of pfc:scene(s)&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:sequence&gt;     &lt;xss:element name="scene" type="scene" minOccurs="1" maxOccurs="unbounded" /&gt;   &lt;/xss:sequence&gt; &lt;/xss:complexType&gt;</pre>
--------	---

## Complex Type scene

Namespace	com:ara:pfc:training:1
Annotations	<p>A scene is a reference msdl:installation with its own independent meteorological data and time of day.</p> <p>Each scene also has a time-in-simulation which represents the number of minutes since the start of the simulation this is used to sort scenes for complex training scenarios</p> <p>An additional description field allows a human readable synopsis to be added. It is likely most training scenarios will be a single scene, but the format supports multiple scenes for mass casualty events when training groups with the hope of allowing teams to train communication of information learned independent of each other and practice medical handoff techniques.</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>Name: xs:string : Name of the scene.</li> <li>Description: xs:string : Brief summary of the scene.</li> <li>Time of day: xs:time : number of seconds from midnight.</li> <li>Time in simulation: xs:integer : Number of seconds since the simulation happened.</li> <li>Weather: xs:string : Summary Information of the Weather</li> <li>Details: Technical Details intended for instructor review</li> </ul> <p>Example:</p> <ul style="list-style-type: none"> <li>Name: Patient and doctor dialogue</li> <li>Description: The injured patient and the doctor discuss plans for how to treat the wound.</li> <li>Time of day: 3600 -- 01:00</li> <li>Time in simulation: 45 -- 00:00::45 second in to simulation</li> <li>Weather: xs:string : Summer;Heavy Rain;Wind</li> <li>Details: Exterior:Desert:Tundra</li> </ul>
Diagram	<pre> classDiagram     class scene {         id         location-id         name         description         details         weather         time-of-day         time-in-simulation         events         items         roles     }     scene "0..1" *-- "1..1" scene     note over scene: A scene is a reference msdl:installation with its own independent meteorological data and time of day. Each scene also...   </pre>
Used by	Element scene-list/scene
Model	ALL(id location-id name description details{0,1} weather{0,1} time-of-day time-in-simulation events items roles)
Children	description, details, events, id, items, location-id, name, roles, time-in-simulation, time-of-day, weather
Source	<pre>&lt;xss:complexType name="scene"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;A scene is a reference msdl:installation with its own independent meteorological data and time of day. Each scene also has a time-in-simulation which represents the number of minutes since the start of the simulation this is used to sort scenes for complex training scenarios An additional description field allows a human readable synopsis to be added. It is likely most training scenarios will be a single scene, but the format supports multiple scenes for mass casualty events when training groups with the hope of allowing teams to train communication of information learned independent of each other and practice medical handoff techniques. Fields: Name: xs:string : Name of the scene. Description: xs:string : Brief summary of the scene. Time of day: xs:time : number of seconds from midnight. Time in simulation: xs:integer : Number of seconds since the simulation happened. Weather: xs:string : Summary Information of the Weather Details: Technical Details intended for instructor review Example: Name: Patient and doctor dialogue Description: The injured patient and the doctor discuss plans for how to treat the wound.</pre>

```

Time of day: 3600 -- 01:00 Time in simulation: 45 -- 00:00::45 second in to simulation Weather:
xs:string : Summer;Heavy Rain;Wind Details: Exterior:Desert:Tundra</xs:documentation>
<xs:annotation>
<xs:all>
  <xs:element name="id" type="xs:string"/>
  <xs:element name="location-id" type="xs:string"/>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="description" type="xs:string"/>
  <xs:element name="details" type="xs:string" minOccurs="0"/>
  <xs:element name="weather" type="xs:string" minOccurs="0"/>
  <xs:element name="time-of-day" type="xs:integer"/>
  <xs:element name="time-in-simulation" type="xs:integer"/>
  <!-- <xs:element name="weather" type="msdl:EnvironmentType" minOccurs="0"/> -->
  <xs:element name="events" type="event-list"/>
  <xs:element name="items" type="item-list"/>
  <xs:element name="roles" type="role-ref-list"/>
</xs:all>
</xs:complexType>

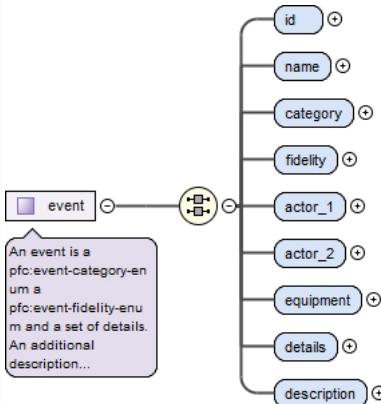
```

## Complex Type event-list

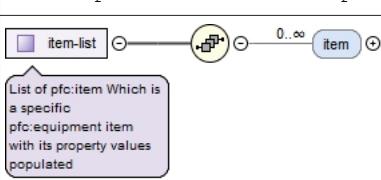
Namespace	com:ara:pfc:training:1
Annotations	List of pfc:event(s)
Diagram	<pre> classDiagram     class eventList {         &lt;&lt;List of pfc:event(s)&gt;&gt;     }     class event     eventList "1..&gt;" o-- "1..&gt;" event   </pre> <p>The diagram shows a class named 'event-list' represented by a rectangle with a purple border. It has a multiplicity of '1..&gt;' at its end and a multiplicity of '1..&gt;' at the other end, connected by a line with an open circle at each end. A rounded rectangle labeled 'List of pfc:event(s)' is positioned below the class.</p>
Used by	Element scene/events
Model	event+
Children	event
Source	<pre> &lt;xs:complexType name="event-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of pfc:event(s)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="event" type="event" minOccurs="1" maxOccurs="unbounded" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type event

Namespace	com:ara:pfc:training:1
Annotations	<p>An event is a pfc:event-category-enum a pfc:event-fidelity-enum and a set of details.</p> <p>An additional description describes the purpose of the prop in the training simulation in human terms on the level of a cliff-note.</p> <p>At this time details is a string field a future update to this spec will define a event-detail-type which will be a choice of the following</p> <ul style="list-style-type: none"> <li>event-action-details</li> <li>event-dialog-details</li> <li>event-movement-details</li> <li>event-acoustic-details</li> <li>event-environment-details</li> <li>event-script-details</li> </ul> <p>Each one with its own set of fields to be manipulated</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>ID: xs:string : Identification</li> <li>Name: xs:string : Name for the event</li> <li>Category: event-category-enum : Describes the type of action that occurs in the event</li> <li>Fidelity: event-fidelity-enum : Intended to be used as a guide by implementations on if the details of this event can be skipped over</li> <li>Actor 1: xs:string : Actor in the event</li> <li>Actor 2: xs:string : Actor in the event</li> <li>Equipment: xs:string : Equipment needed for the event</li> <li>Details: xs:string : Future update will define event-detail-type with own set of fields to be manipulated</li> <li>Description: xs:string : Brief summary of the event</li> </ul> <p>Example:</p> <ul style="list-style-type: none"> <li>ID: 0010</li> <li>Name: Patient Briefing</li> <li>Category: Dialog</li> <li>Fidelity: Low</li> <li>Actor 1: Doctor</li> <li>Actor 2: Burn Wound Patient</li> </ul>

	<p>Equipment: N/A          Details: event-script-details          Description: The patient is briefed on their wounds and the plan to treat them.</p>
Diagram	 <pre> classDiagram     class event {         id         name         category         fidelity         actor_1         actor_2         equipment         details         description     }     actor_1     actor_2     category     description     details     equipment     fidelity     id     name     </pre>
Used by	Element      event-list/event
Model	ALL(id name category fidelity actor_1 actor_2 equipment details description)
Children	actor_1, actor_2, category, description, details, equipment, fidelity, id, name
Source	<pre> &lt;xs:complexType name="event"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An event is a pfc:event-category-enum a pfc:event-fidelity-enum and a set of details. An additional description describes the purpose of the prop in the training simulation in human terms on the level of a cliff-note. At this time details is a string field a future update to this spec will define a event-detail-type which will be a choice of the following event-action-details event-dialog-details event-movement-details event-acoustic-details event-environment-details event-script-details       &lt;!--Complex max fidelity code scripting to be loaded by level creators --&gt;Each one with its own set of fields to be manipulated Fields: ID: xs:string : Identification Name: xs:string : Name for the event Category: event-category-enum : Describes the type of action that occurs in the event Fidelity: event-fidelity-enum : Intended to be used as a guide by implementations on if the details of this event can be skipped over Actor 1: xs:string : Actor in the event Actor 2: xs:string : Actor in the event Equipment: xs:string : Equipment needed for the event Details: xs:string : Future update will define event-detail-type with own set of fields to be manipulated Description: xs:string : Brief summary of the event Example: ID: 0010 Name: Patient Briefing Category: Dialog Fidelity: Low Actor 1: Doctor Actor 2: Burn Wound Patient Equipment: N/A Details: event-script-details Description: The patient is briefed on their wounds and the plan to treat them.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="id" type="xs:string"/&gt;     &lt;xs:element name="name" type="xs:string"/&gt;     &lt;xs:element name="category" type="event-category-enum"/&gt;     &lt;xs:element name="fidelity" type="event-fidelity-enum"/&gt;     &lt;xs:element name="actor_1" type="xs:string"/&gt;     &lt;xs:element name="actor_2" type="xs:string"/&gt;     &lt;xs:element name="equipment" type="xs:string"/&gt;     &lt;xs:element name="details" type="xs:string"/&gt;     &lt;xs:element name="description" type="xs:string"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type item-list

Namespace	com:ara:pfc:training:1
Annotations	List of pfc:item Which is a specific pfc:equipment item with its property values populated
Diagram	 <pre> classDiagram     class item-list {         item*     }     item     </pre>
Used by	Element      scene/items
Model	item*
Children	item
Source	<pre> &lt;xs:complexType name="item-list"&gt;   &lt;xs:annotation&gt;     </pre>

```

<xs:documentation>List of pfc:item Which is a specific pfc:equipment item with its property values populated</xs:documentation>
<xs:annotation>
<xs:sequence>
<xs:element name="item" type="item" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>

```

## Complex Type item

Namespace	com:ara:pfc:training:1
Annotations	<p>A prop is a reference to a pfc:equipment  An additional description describes the purpose of the prop in the training simulation in human terms on the level of a cliff-note  The short-name is a unique string to be use when parsing a training-script event to quickly refer to a prop</p> <p>Fields:  Short Name: xs:string : Unique string to be used when parsing a training-script event to quickly refer to a prop  Equipment ID: xs:string : Equipment identification  Description: xs:string : Brief summary of the item  Properties: property-value-list : Unique characteristics of the item</p> <p>Example:  Short Name: vYi4s  Equipment ID: 01  Description: A cot for the patient to lay on  Properties: Where the patient lays to be treated</p>
Diagram	<pre> classDiagram     class item {         short-name         equipment-id         description         properties     }     note over item: A prop is a reference to a pfc:equipment An additional description describes the purpose of the prop in the training... </pre>
Used by	Element item-list/item
Model	ALL(short-name equipment-id description properties)
Children	description, equipment-id, properties, short-name
Source	<pre> &lt;xs:complexType name="item"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;A prop is a reference to a pfc:equipment An additional description describes the purpose of the prop in the training simulation in human terms on the level of a cliff-note  The short-name is a unique string to be use when parsing a training-script event to quickly refer to a prop  Fields: Short Name: xs:string : Unique string to be used when parsing a training-script event to quickly refer to a prop  Equipment ID: xs:string : Equipment identification  Description: xs:string : Brief summary of the item  Properties: property-value-list : Unique characteristics of the item  Example: Short Name: vYi4s Equipment ID: 01 Description: A cot for the patient to lay on  Properties: Where the patient lays to be treated&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:all&gt; &lt;xs:element name="short-name" type="xs:string" /&gt; &lt;xs:element name="equipment-id" type="xs:string" /&gt; &lt;xs:element name="description" type="xs:string" /&gt; &lt;xs:element name="properties" type="property-value-list" /&gt; &lt;/xs:all&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type property-value-list

Namespace	com:ara:pfc:training:1
Annotations	List of property-values
Diagram	<pre> classDiagram     class property-value-list {         value     }     note over property-value-list: List of property-values </pre>
Used by	Element item/properties

Model	value*
Children	value
Source	<pre>&lt;xs:complexType name="property-value-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of property-values&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="value" type="property-value" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type property-value

Namespace	com:ara:pfc:training:1
Annotations	The value for a property. They are broken into two types because properties are generally defined in definition types while values appear in instance types;
Diagram	<p>The value for a property. They are broken into two types because properties are generally defined in definition types while values appear in instance types...</p>
Used by	Element property-value-list/value
Model	ALL(name value)
Children	name, value
Source	<pre>&lt;xs:complexType name="property-value"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The value for a property. They are broken into two types because properties are generally defined in definition types while values appear in instance types;&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="name" type="xs:string"/&gt;     &lt;xs:element name="value" type="xs:string"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type role-ref-list

Namespace	com:ara:pfc:training:1
Diagram	
Used by	Element scene/roles
Model	role-ref*
Children	role-ref
Source	<pre>&lt;xs:complexType name="role-ref-list"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="role-ref" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type citation-list

Namespace	com:ara:pfc:training:1
Annotations	List of Citation Information that a citation-ref-list refers to
Diagram	<p>List of Citation Information that a citation-ref-list refers to</p>
Used by	Element ScenarioSchema/works-cited
Model	citation*
Children	citation

Source	<pre>&lt;xs:complexType name="citation-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;List of Citation Information that a citation-ref-list refers to&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="citation" type="citation" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>
--------	---

## Complex Type citation

Namespace	com:ara:pfc:training:1
Annotations	<p>General Citation Information for a bib reference</p> <p>Fields:</p> <ul style="list-style-type: none"> <li>UUID: xs:string : A universally unique identifier (UUID) is a 128-bit number used to identify information in computer systems</li> <li>Key: xs:string : Unique identifier for the given reference</li> <li>Authors: xs:string : The authors of the given reference</li> <li>Title: xs:string : The title of the reference</li> <li>Date: xs:string : The date that the reference was published</li> <li>Page: xs:string : The page number of the reference (if applicable)</li> <li>Accessed: xs:string : When the reference was accessed by the user</li> </ul> <p>Example:</p> <pre>UUID: a313dc22-17e6-45fa-a3d5-0adf04993f10 Key: {citekey} Authors: John Doe, Jane Smith Title: Example Reference Date: 01/01/2020 Page: 20-30 Accessed: 05/05/2020</pre>
Diagram	<pre> classDiagram     class citation {         &lt;&lt;General Citation Information for a bib reference&gt;&gt;         &lt;&lt;Fields: UUID: xs:string ; Key: xs:string ; Authors: xs:string ; Title: xs:string ; Date: xs:string ; Page: xs:string ; Accessed: xs:string ;&gt;&gt;         self-referencing association "General Citation Information for a bib reference" to self     }     "General Citation Information for a bib reference" {         &lt;&lt;Fields: UUID: xs:string ; Key: xs:string ; Authors: xs:string ; Title: xs:string ; Date: xs:string ; Page: xs:string ; Accessed: xs:string ;&gt;&gt;         &lt;&lt;Fields: uuid, key, authors[1..oo], title, date, page, accessed&gt;&gt;     }     "General Citation Information for a bib reference" "*" --&gt; "General Citation Information for a bib reference"     "General Citation Information for a bib reference" --&gt; "General Citation Information for a bib reference"     "General Citation Information for a bib reference" --&gt; "General Citation Information for a bib reference"     "General Citation Information for a bib reference" --&gt; "General Citation Information for a bib reference"     "General Citation Information for a bib reference" --&gt; "General Citation Information for a bib reference"     "General Citation Information for a bib reference" --&gt; "General Citation Information for a bib reference" </pre>
Used by	Element citation-list/citation
Model	uuid , key{0,1} , authors+ , title , date , page{0,1} , accessed{0,1}
Children	accessed, authors, date, key, page, title, uuid
Source	<pre>&lt;xs:complexType name="citation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;General Citation Information for a bib reference Fields: UUID: xs:string : A universally unique identifier (UUID) is a 128-bit number used to identify information in computer systems Key: xs:string : Unique identifier for the given reference Authors: xs:string : The authors of the given reference Title: xs:string : The title of the reference Date: xs:string : The date that the reference was published Page: xs:string : The page number of the reference (if applicable) Accessed: xs:string : When the reference was accessed by the user Example: UUID: a313dc22-17e6-45fa-a3d5-0adf04993f10 Key: {citekey} Authors: John Doe, Jane Smith Title: Example Reference Date: 01/01/2020 Page: 20-30 Accessed: 05/05/2020&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="uuid" type="xs:string" minOccurs="1" maxOccurs="1"/&gt;     &lt;xs:element name="key" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="authors" type="xs:string" minOccurs="1" maxOccurs="unbounded"/&gt;     &lt;xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/&gt;     &lt;xs:element name="date" type="xs:string" minOccurs="1" maxOccurs="1"/&gt;     &lt;xs:element name="page" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="accessed" type="xs:string" minOccurs="0" maxOccurs="1"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type cpg-list

Namespace	com:ara:pfc:training:1
-----------	------------------------

Annotations	List of CPG specific citations. This tag is differentated from the citations tag in order to make it easier for user interfaces to parse out all associated CPGs and make mapping widgets.
Diagram	<pre> sequenceDiagram     participant A as cpg-list     participant B as cpg     A-&gt;&gt;B:      activate B     B--&gt;&gt;A: 0..∞     deactivate B   </pre> <p>A UML sequence diagram illustrating the relationship between 'cpg-list' and 'cpg'. A lifeline labeled 'cpg-list' sends a message to a lifeline labeled 'cpg'. The 'cpg' lifeline has a multiplicity of '0..∞' at its end, indicating it can appear multiple times in the sequence.</p>
Model	cpg*
Children	cpg
Source	<pre> &lt;xss:complexType name="cpg-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of CPG specific citations. This tag is differentated from the citations tag in order to make it easier for user interfaces to parse out all associated CPGs and make mapping widgets.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:sequence&gt;     &lt;xss:element name="cpg" type="cpg" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xss:sequence&gt; &lt;/xss:complexType&gt;   </pre>

## Complex Type cpg

Namespace	com:ara:pfc:training:1
Annotations	Clinical Practice Guideline
Diagram	<pre> sequenceDiagram     participant A as cpg     A--&gt;&gt;A: name     A--&gt;&gt;A: description     A--&gt;&gt;A: citation-ref   </pre> <p>A UML class diagram showing the 'cpg' class. It has three associations with itself, each labeled with a multiplicity of '+': one for 'name', one for 'description', and one for 'citation-ref'.</p>
Used by	Element cpg-list/cpg
Model	ALL(name description citation-ref)
Children	citation-ref, description, name
Source	<pre> &lt;xss:complexType name="cpg"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Clinical Practice Guideline&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:all&gt;     &lt;xss:element name="name" type="xs:string"/&gt;     &lt;xss:element name="description" type="xs:string"/&gt;     &lt;xss:element name="citation-ref" type="xs:string"/&gt;   &lt;/xss:all&gt; &lt;/xss:complexType&gt;   </pre>

## Complex Type learning-objective-reference-list

Namespace	com:ara:pfc:training:1
Annotations	List of UUIDs from the lessons list in the learning-objective-list in a pfc:syllabus
Diagram	<pre> sequenceDiagram     participant A as learning-objective-reference-list     participant B as objective     A-&gt;&gt;B:      activate B     B--&gt;&gt;A: 0..∞     deactivate B   </pre> <p>A UML sequence diagram illustrating the relationship between 'learning-objective-reference-list' and 'objective'. A lifeline labeled 'learning-objective-reference-list' sends a message to a lifeline labeled 'objective'. The 'objective' lifeline has a multiplicity of '0..∞' at its end, indicating it can appear multiple times in the sequence.</p>
Model	objective*
Children	objective
Source	<pre> &lt;xss:complexType name="learning-objective-reference-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;List of UUIDs from the lessons list in the learning-objective-list in a pfc:syllabus&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:sequence&gt;     &lt;xss:element name="objective" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xss:sequence&gt; &lt;/xss:complexType&gt;   </pre>

## Simple Type(s)

### Simple Type event-category-enum

Namespace	com:ara:pfc:training:1	
Annotations	Enumerated choice of an pfc:event category each of which describes one type of action that can occur in this event	
Diagram	<p>The diagram shows a UML class named 'event-category-enum' with a hollow diamond symbol indicating it is a derived class. It has a directed association line pointing to another class named 'xs:string'. Two callout boxes provide additional information: one for 'event-category-enum' stating 'Enumerated choice of an pfc:event category each of which describes one type of action that can occur in this event', and another for 'xs:string' stating 'Built-in primitive type. The string datatype represents character strings in XML.'</p>	
Type	restriction of xs:string	
Facets	enumeration	ACTION An Action is the result of a Role acting upon a piece of equipment or another Role. Can be detailed at all fidelity levels
	enumeration	DIALOG Dialog is anytime a Role or Object performs audible dialog in the scene. Typically for Medium or Higher fidelity.
	enumeration	MOVEMENT Any time a Role or Object changes location in the scene .
	enumeration	SOUND Background or ambient sound affects typically at a high fidelity level as most low and medium fedeity implementations do not have sound systems.
	enumeration	ENVIRONMENT Typically used to describe changes in set design or weather events almost always reserved for high fidelity simulations.
Used by	Element	event/category
Source	<pre> &lt;xs:simpleType name="event-category-enum"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Enumerated choice of an pfc:event category each of which describes one type of action that can occur in this event&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="ACTION"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;An Action is the result of a Role acting upon a piece of equipment or another Role. Can be detailed at all fidelity levels&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;     &lt;xs:enumeration value="DIALOG"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Dialog is anytime a Role or Object performs audible dialog in the scene. Typically for Medium or Higher fidelity.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;     &lt;xs:enumeration value="MOVMENT"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Any time a Role or Object changes location in the scene .&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;     &lt;xs:enumeration value="SOUND"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Background or ambient sound affects typically at a high fidelity level as most low and medium fedeity implementations do not have sound systems.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;     &lt;xs:enumeration value="ENVIRONMENT"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Typically used to describe changes in set design or weather events almost always reserved for high fidelity simulations.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>	

### Simple Type event-fidelity-enum

Namespace	com:ara:pfc:training:1
-----------	------------------------

Annotations	Enumerated choice of an pfc:event fidelity is intended to be used as a guide by implementations on if the details of this event can be skipped over. Additional fidelity levels may be included in the future	
Diagram		
Type	restriction of xs:string	
Facets	enumeration	LOW Low fidelity implementations are intended for traditional medical simulators which use live actors are need only to know the high level details like initial injuries and placement of equipment in the scene.
	enumeration	MEDIUM Medium fidelity implementations are higher level simulations which include some scripted dialog and movement. The details included at this level give non trainees instructions on how to act when not interacting directly with the trainee
	enumeration	HIGH High fidelity simulations are typically fully virtual simulations which must create atmospheric and background effects to enhance immersion We are truly in the weeds at this level of scripting.
Used by	Element	event/fidelity
Source	<pre>&lt;xs:simpleType name="event-fidelity-enum"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Enumerated choice of an pfc:event fidelity is intended to be used as a guide by implementations on if the details of this event can be skipped over. Additional fidelity levels may be included in the future&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="LOW"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Low fidelity implementations are intended for traditional medical simulators which use live actors are need only to know the high level details like initial injuries and placement of equipment in the scene.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;     &lt;xs:enumeration value="MEDIUM"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Medium fidelity implementations are higher level simulations which include some scripted dialog and movement. The details included at this level give non trainees instructions on how to act when not interacting directly with the trainee&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;     &lt;xs:enumeration value="HIGH"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;High fidelity simulations are typically fully virtual simulations which must create atmospheric and background effects to enhance immersion We are truly in the weeds at this level of scripting.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:enumeration&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	

## Namespace: ""

### Element(s)

#### Element ScenarioSchema / summary

Namespace	No namespace
Annotations	General information on this scenario

Diagram	<pre> classDiagram     class summary {         &lt;&lt;General information on this scenario&gt;&gt;     }     class pfc:summary {         title         description         version         classification         keywords         domain         limitations     }     summary "1" -- "*" pfc:summary     note over pfc:summary: Summary stores all general information about the current scenario file. Fields: Title: xs:string : The name of the...   </pre>
Type	summary
Properties	content: complex
Model	ALL(title{0,1} description{0,1} version{0,1} classification{0,1} keywords{0,1} domain{0,1} limitations{0,1})
Children	classification, description, domain, keywords, limitations, title, version
Instance	<pre> &lt;summary&gt;   &lt;title&gt;{0..1}&lt;/title&gt;   &lt;description&gt;{0..1}&lt;/description&gt;   &lt;version&gt;{0..1}&lt;/version&gt;   &lt;classification&gt;{0..1}&lt;/classification&gt;   &lt;keywords&gt;{0..1}&lt;/keywords&gt;   &lt;domain&gt;{0..1}&lt;/domain&gt;   &lt;limitations&gt;{0..1}&lt;/limitations&gt; &lt;/summary&gt;   </pre>
Source	<pre> &lt;xss:element name="summary" type="pfc:summary"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;General information on this scenario&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;   </pre>

### Element summary / title

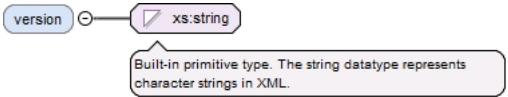
Namespace	No namespace
Diagram	<pre> classDiagram     class summary {         title     }     class xs:string     summary "1" -- "*" xs:string     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>
Type	xs:string
Properties	content: simple
Properties	minOccurs: 0
Source	<pre> &lt;xss:element name="title" type="xs:string" minOccurs="0" /&gt;   </pre>

### Element summary / description

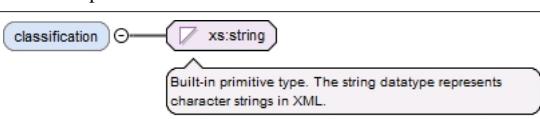
Namespace	No namespace
Diagram	<pre> classDiagram     class summary {         description     }     class xs:string     summary "1" -- "*" xs:string     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>
Type	xs:string
Properties	content: simple
Properties	minOccurs: 0
Source	<pre> &lt;xss:element name="description" type="xs:string" minOccurs="0" /&gt;   </pre>

### Element summary / version

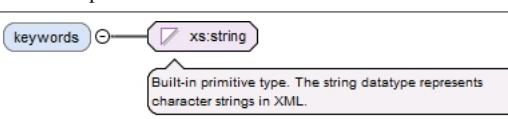
Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="version" type="xs:string" minOccurs="0" />

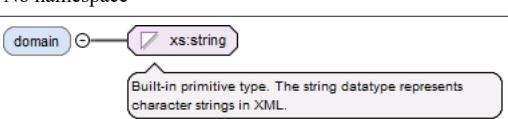
## Element summary / classification

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="classification" type="xs:string" minOccurs="0" />

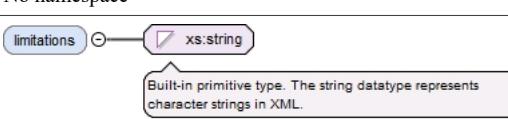
## Element summary / keywords

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="keywords" type="xs:string" minOccurs="0" />

## Element summary / domain

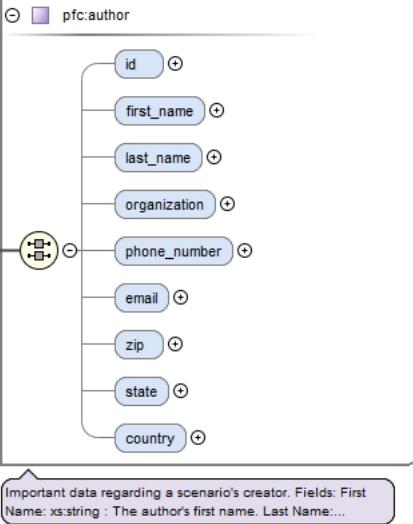
Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="domain" type="xs:string" minOccurs="0" />

## Element summary / limitations

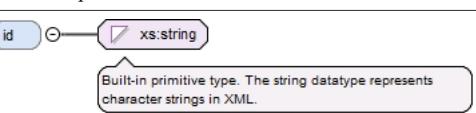
Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p>

	minOccurs:	0
Source	<xs:element name="limitations" type="xs:string" minOccurs="0"/>	

**Element ScenarioSchema / author**

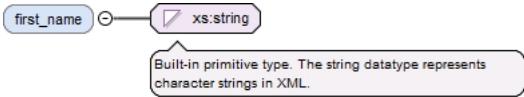
Namespace	No namespace
Annotations	The creator of this scenario
Diagram	 <pre> classDiagram     class author {         &lt;&lt;author         Type pfc:author&gt;&gt;         &lt;&lt;The creator of this scenario&gt;&gt;     }     author &lt; -- pfc:author     pfc:author {         id         first_name         last_name         organization         phone_number         email         zip         state         country     }     note over pfc:author: Important data regarding a scenario's creator. Fields: First Name: xs:string : The author's first name. Last Name:...   </pre>
Type	author
Properties	content: complex
Model	ALL(id{0,1} first_name{0,1} last_name{0,1} organization{0,1} phone_number{0,1} email{0,1} zip{0,1} state{0,1} country{0,1})
Children	country, email, first_name, id, last_name, organization, phone_number, state, zip
Instance	<pre> &lt;author&gt;   &lt;id&gt;{0,1}&lt;/id&gt;   &lt;first_name&gt;{0,1}&lt;/first_name&gt;   &lt;last_name&gt;{0,1}&lt;/last_name&gt;   &lt;organization&gt;{0,1}&lt;/organization&gt;   &lt;phone_number&gt;{0,1}&lt;/phone_number&gt;   &lt;email&gt;{0,1}&lt;/email&gt;   &lt;zip&gt;{0,1}&lt;/zip&gt;   &lt;state&gt;{0,1}&lt;/state&gt;   &lt;country&gt;{0,1}&lt;/country&gt; &lt;/author&gt;   </pre>
Source	<pre> &lt;xs:element name="author" type="pfc:author"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The creator of this scenario&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

**Element author / id**

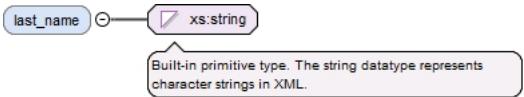
Namespace	No namespace
Diagram	 <pre> classDiagram     class id {         &lt;&lt;xs:string&gt;&gt;     }     id &lt; -- xs:string     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>
Type	xs:string
Properties	content: simple
	minOccurs: 0
Source	<xs:element name="id" type="xs:string" minOccurs="0"/>

**Element author / first\_name**

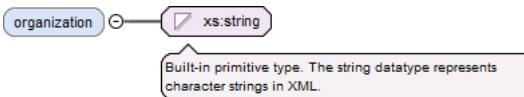
Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xss:element name="first_name" type="xs:string" minOccurs="0"/>

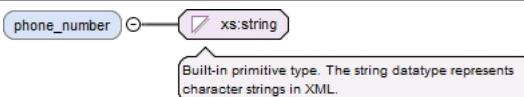
**Element author / last\_name**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xss:element name="last_name" type="xs:string" minOccurs="0"/>

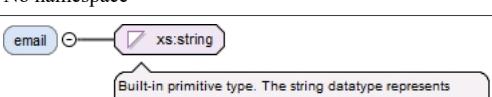
**Element author / organization**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xss:element name="organization" type="xs:string" minOccurs="0"/>

**Element author / phone\_number**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xss:element name="phone_number" type="xs:string" minOccurs="0"/>

**Element author / email**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>

Source <xs:element name="email" type="xs:string" minOccurs="0" />

## Element author / zip

Namespace	No namespace				
Diagram	<pre> classDiagram     class zip {         &lt;&lt;xs:string&gt;&gt;     }     zip "1" -- "0..&gt;" xs:string     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>				
Type	xs:string				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre>&lt;xs:element name="zip" type="xs:string" minOccurs="0"/&gt;</pre>				

### **Element author / state**

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<pre>&lt;xs:element name="state" type="xs:string" minOccurs="0" /&gt;</pre>

### **Element author / country**

Namespace	No namespace				
Diagram	 <pre> classDiagram     class country {         &lt;&lt;xs:string&gt;&gt;     }     country "1" -- "*" xs:string   </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>				
Type	xs:string				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre>&lt;xs:element name="country" type="xs:string" minOccurs="0" /&gt;</pre>				

**Element scenarioSchema / equipment**

Namespace	No namespace
Annotations	The bibliography used to create this scenario.
Diagram	<pre> classDiagram     class equipment {         &lt;&lt;pfc:equipment-definition-list&gt;&gt;     }     equipment "0..∞" *--&gt; "0..∞" equipment     note over equipment: The bibliography used to create this scenario.     note over equipment --&gt; equipment: List of pfc:equipment types.   </pre>
Type	equipment-definition-list
Properties	content: complex
Model	equipment*
Children	equipment
Instance	<pre> &lt;equipment&gt;   &lt;equipment&gt;{0,unbounded}&lt;/equipment&gt; &lt;/equipment&gt;   </pre>

Source	<pre>&lt;xss:element name="equipment" type="pfc:equipment-definition-list"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;The bibliography used to create this scenario.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>
--------	--

## Element equipment-definition-list / equipment

Namespace	No namespace						
Diagram	<p>A UML class diagram representing the 'equipment' element. The class is named 'equipment'. It has several attributes: 'id', 'name', 'type', 'description', 'citations', 'image', and 'properties'. The 'description' attribute is marked with a solid line to its right, indicating it is a required element.</p>						
Type	equipment						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	ALL(id name type{0,1} description citations image{0,1} properties)						
Children	citations, description, id, image, name, properties, type						
Instance	<pre>&lt;equipment&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;name&gt;{1,1}&lt;/name&gt;   &lt;type&gt;{0,1}&lt;/type&gt;   &lt;description&gt;{1,1}&lt;/description&gt;   &lt;citations&gt;{1,1}&lt;/citations&gt;   &lt;image&gt;{0,1}&lt;/image&gt;   &lt;properties&gt;{1,1}&lt;/properties&gt; &lt;/equipment&gt;</pre>						
Source	<pre>&lt;xss:element name="equipment" type="pfc:equipment-definition-list" minOccurs="0" maxOccurs="unbounded" /&gt;</pre>						

## Element equipment / id

Namespace	No namespace
Diagram	<p>A UML class diagram showing the 'id' attribute. It is associated with the 'xs:string' type via a line with a hollow arrowhead.</p>
Type	xs:string
Properties	content: simple
Source	<pre>&lt;xss:element name="id" type="xs:string" /&gt;</pre>

## Element equipment / name

Namespace	No namespace
Diagram	<p>A UML class diagram showing the 'name' attribute. It is associated with the 'xs:string' type via a line with a hollow arrowhead.</p>
Type	xs:string

Properties	content: simple minOccurs: 1
Source	<xs:element name="name" type="xs:string" minOccurs="1"/>

**Element equipment / type**

Namespace	No namespace
Diagram	
Type	xs:integer
Properties	content: simple minOccurs: 0
Source	<xs:element name="type" type="xs:integer" minOccurs="0"/>

**Element equipment / description**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xs:element name="description" type="xs:string" minOccurs="1"/>

**Element equipment / citations**

Namespace	No namespace
Diagram	
Type	citation-ref-list
Properties	content: complex minOccurs: 1
Model	citation-ref*
Children	citation-ref
Instance	<citations> <citation-ref>{0,unbounded}</citation-ref> </citations>
Source	<xs:element name="citations" type="citation-ref-list" minOccurs="1"/>

**Element citation-ref-list / citation-ref**

Namespace	No namespace
Diagram	
Type	xs:string

Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Source	<xss:element name="citation-ref" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>

**Element equipment / image**

Namespace	No namespace
Diagram	 A UML class diagram fragment. An 'image' node is connected by a line to an 'xs:string' node. A callout box points to the 'xs:string' node with the text: 'Built-in primitive type. The string datatype represents character strings in XML.'.
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xss:element name="image" type="xs:string" minOccurs="0"/>

**Element equipment / properties**

Namespace	No namespace
Diagram	 A UML class diagram fragment. A 'properties' node is connected by a line to a 'property' node. A callout box points to the 'property' node with the text: 'List of Citation Information that a citation-ref-list refers to'.
Type	equipment-properties-list
Properties	content: complex minOccurs: 1
Model	property*
Children	property
Instance	<properties> <property>{0,unbounded}</property> </properties>
Source	<xss:element name="properties" type="equipment-properties-list" minOccurs="1"/>

**Element equipment-properties-list / property**

Namespace	No namespace
Diagram	 A UML class diagram fragment. A 'property' node is connected by a line to an 'equipment-property' node. The 'equipment-property' node has three attributes: 'name', 'type', and 'fields'. A callout box points to the 'equipment-property' node with the text: 'A property is a configurable value of a piece of equipment. Each equipment can have any number of properties. The...'.
Type	equipment-property
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	name , type , fields
Children	fields, name, type
Instance	<property>

	<pre>&lt;name&gt;{1,1}&lt;/name&gt; &lt;type&gt;{1,1}&lt;/type&gt; &lt;fields&gt;{1,1}&lt;/fields&gt; &lt;/property&gt;</pre>
Source	<code>&lt;xss:element name="property" type="equipment-property" minOccurs="0" maxOccurs="unbounded" /&gt;</code>

### Element equipment-property / name

Namespace	No namespace
Diagram	<p>The diagram shows the 'name' element connected to an 'xs:string' type. A callout box indicates that 'xs:string' is a built-in primitive type representing character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="name" type="xs:string"/&gt;</code>

### Element equipment-property / type

Namespace	No namespace
Diagram	<p>The diagram shows the 'type' element connected to an 'xs:string' type. A callout box indicates that 'xs:string' is a built-in primitive type representing character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="type" type="xs:string"/&gt;</code>

### Element equipment-property / fields

Namespace	No namespace
Diagram	<p>The diagram shows the 'fields' element connected to a 'property-field-list' type. This list contains multiple 'field' elements, indicated by a multiplicity of 0..∞. A callout box specifies that this list defines properties for equipment in the syllabus.</p>
Type	property-field-list
Properties	content: complex
Model	field*
Children	field
Instance	<code>&lt;fields&gt;   &lt;field&gt;{0,unbounded}&lt;/field&gt; &lt;/fields&gt;</code>
Source	<code>&lt;xss:element name="fields" type="property-field-list"/&gt;</code>

### Element property-field-list / field

Namespace	No namespace
Diagram	<p>The diagram shows the 'field' element connected to a 'field-type' type. This type includes attributes for 'name' and 'type'. A callout box notes that these values represent property types, which are broken into two types based on definition.</p>

Type	field-type
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	ALL(name type)
Children	name, type
Instance	<pre>&lt;field&gt;   &lt;name&gt;{1,1}&lt;/name&gt;   &lt;type&gt;{1,1}&lt;/type&gt; &lt;/field&gt;</pre>
Source	<code>&lt;xs:element name="field" type="field-type" minOccurs="0" maxOccurs="unbounded"/&gt;</code>

## Element field-type / name

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="name" type="xs:string"/&gt;</code>

## Element field-type / type

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="type" type="xs:string"/&gt;</code>

## Element ScenarioSchema / trauma-definitions

Namespace	No namespace
Annotations	XML Database of Injuries contained that can be referenced in this file.
Diagram	
Type	trauma-definition-list
Properties	content: complex
Model	trauma*
Children	trauma
Instance	<pre>&lt;trauma-definitions&gt;   &lt;trauma&gt;{0,unbounded}&lt;/trauma&gt; &lt;/trauma-definitions&gt;</pre>
Source	<code>&lt;xs:element name="trauma-definitions" type="pfc:trauma-definition-list"&gt;</code> <code>&lt;xs:annotation&gt;</code> <code>&lt;xs:documentation&gt;XML Database of Injuries contained that can be referenced in this file.&lt;/xs:documentation&gt;</code> <code>&lt;/xs:annotation&gt;</code> <code>&lt;/xs:element&gt;</code>

**Element trauma-definition-list / trauma**

Namespace	No namespace						
Diagram	<pre> classDiagram     class trauma {         id         common-name         medical-name         citations         description         severity-range     }     trauma &lt; -- traumas     note over traumas: A PFC representation of a simple trauma. Injuries are associated to a Injury-Profile where severity and trauma-site are...   </pre>						
Type	trauma						
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	ALL(id common-name{0,1} medical-name citations description severity-range)						
Children	citations, common-name, description, id, medical-name, severity-range						
Instance	<pre> &lt;trauma&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;common-name&gt;{0,1}&lt;/common-name&gt;   &lt;medical-name&gt;{1,1}&lt;/medical-name&gt;   &lt;citations&gt;{1,1}&lt;/citations&gt;   &lt;description&gt;{1,1}&lt;/description&gt;   &lt;severity-range&gt;{1,1}&lt;/severity-range&gt; &lt;/trauma&gt;   </pre>						
Source	<code>&lt;xss:element name="trauma" type="trauma" minOccurs="0" maxOccurs="unbounded"/&gt;</code>						

**Element trauma / id**

Namespace	No namespace		
Diagram	<pre> classDiagram     attribute id     attribute xs:string     id &lt;--&gt; xs:string     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>		
Type	xs:string		
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<code>&lt;xss:element name="id" type="xs:string"/&gt;</code>		

**Element trauma / common-name**

Namespace	No namespace				
Diagram	<pre> classDiagram     attribute common-name     attribute xs:string     common-name &lt;--&gt; xs:string     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>				
Type	xs:string				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code>&lt;xss:element name="common-name" type="xs:string" minOccurs="0"/&gt;</code>				

**Element trauma / medical-name**

Namespace	No namespace
-----------	--------------

Diagram	A UML class diagram fragment. An oval labeled "medical-name" has a directed association line pointing to a rectangle labeled "xs:string". A callout box below the association line states: "Built-in primitive type. The string datatype represents character strings in XML."/>
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xs:element name="medical-name" type="xs:string" minOccurs="1"/>

**Element trauma / citations**

Namespace	No namespace
Diagram	A UML class diagram fragment. An oval labeled "citations" has a directed association line pointing to a rectangle labeled "citation-ref-list". This rectangle contains another association line pointing to an oval labeled "citation-ref" with a multiplicity of "0..∞". A callout box below the "citation-ref-list" box states: "List of strings which should be citations for source material. This could be expanded to be a more stringent field..."
Type	citation-ref-list
Properties	content: complex minOccurs: 1
Model	citation-ref*
Children	citation-ref
Instance	<citations> <citation-ref>{0,unbounded}</citation-ref> </citations>
Source	<xs:element name="citations" type="citation-ref-list" minOccurs="1"/>

**Element trauma / description**

Namespace	No namespace
Diagram	A UML class diagram fragment. An oval labeled "description" has a directed association line pointing to a rectangle labeled "xs:string". A callout box below the association line states: "Built-in primitive type. The string datatype represents character strings in XML."/>
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xs:element name="description" type="xs:string" minOccurs="1"/>

**Element trauma / severity-range**

Namespace	No namespace
Diagram	A UML class diagram fragment. An oval labeled "severity-range" has a directed association line pointing to a rectangle labeled "trauma-severity-range". This rectangle contains two other ovals: "numeric-range" and "token-range", connected by a line with a junction point. A callout box below the "trauma-severity-range" box states: "Every trauma has a severity range which can either be presented as a numeric-range or a token range. Token ranges are a..."
Type	trauma-severity-range
Properties	content: complex minOccurs: 1
Model	numeric-range   token-range
Children	numeric-range, token-range

Instance	<pre>&lt;severity-range&gt;   &lt;numeric-range&gt;{1,1}&lt;/numeric-range&gt;   &lt;token-range&gt;{1,1}&lt;/token-range&gt; &lt;/severity-range&gt;</pre>
Source	<pre>&lt;xs:element name="severity-range" type="trauma-severity-range" minOccurs="1" /&gt;</pre>

## Element trauma-severity-range / numeric-range

Namespace	No namespace
Diagram	<pre> classDiagram     class numeric-range {         lower_bound         upper_bound         median         unit     }     numeric-range &lt; -- numeric-range   </pre> <p>Numerical range of two doubles with an optional median value and unit type. The trauma definition mostly uses this for...</p>
Type	numeric-range
Properties	content: complex
Model	ALL(lower_bound upper_bound median{0,1} unit{0,1})
Children	lower_bound, median, unit, upper_bound
Instance	<pre>&lt;numeric-range&gt;   &lt;lower_bound&gt;{1,1}&lt;/lower_bound&gt;   &lt;upper_bound&gt;{1,1}&lt;/upper_bound&gt;   &lt;median&gt;{0,1}&lt;/median&gt;   &lt;unit&gt;{0,1}&lt;/unit&gt; &lt;/numeric-range&gt;</pre>
Source	<pre>&lt;xs:element name="numeric-range" type="numeric-range" /&gt;</pre>

## Element numeric-range / lower\_bound

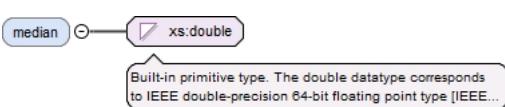
Namespace	No namespace
Diagram	<pre> classDiagram     class lower_bound {         &lt; -- xs:double     }     lower_bound &lt; -- xs:double   </pre> <p>Built-in primitive type. The double datatype corresponds to IEEE double-precision 64-bit floating point type [IEEE...]</p>
Type	xs:double
Properties	content: simple minOccurs: 1
Source	<pre>&lt;xs:element name="lower_bound" type="xs:double" minOccurs="1" /&gt;</pre>

## Element numeric-range / upper\_bound

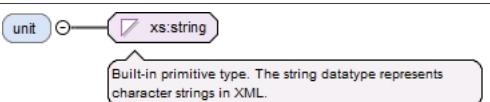
Namespace	No namespace
Diagram	<pre> classDiagram     class upper_bound {         &lt; -- xs:double     }     upper_bound &lt; -- xs:double   </pre> <p>Built-in primitive type. The double datatype corresponds to IEEE double-precision 64-bit floating point type [IEEE...]</p>
Type	xs:double
Properties	content: simple minOccurs: 1
Source	<pre>&lt;xs:element name="upper_bound" type="xs:double" minOccurs="1" /&gt;</pre>

## Element numeric-range / median

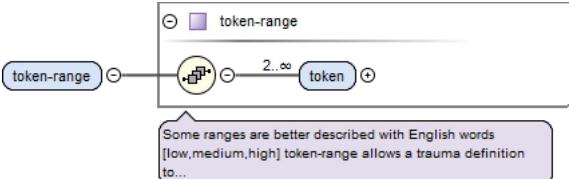
Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:double
Properties	content: simple minOccurs: 0
Source	<xs:element name="median" type="xs:double" minOccurs="0" />

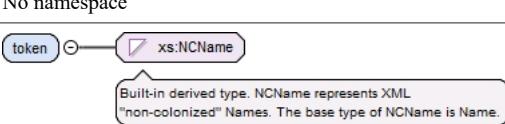
**Element numeric-range / unit**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xs:element name="unit" type="xs:string" minOccurs="0" />

**Element trauma-severity-range / token-range**

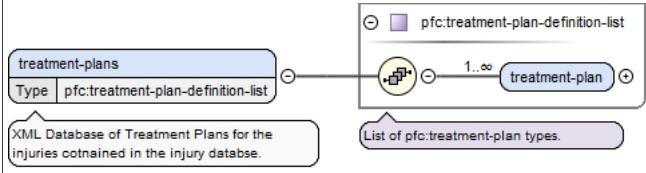
Namespace	No namespace
Diagram	
Type	token-range
Properties	content: complex
Model	token{2,unbounded}
Children	token
Instance	<token-range> <token>{2,unbounded}</token> </token-range>
Source	<xs:element name="token-range" type="token-range" />

**Element token-range / token**

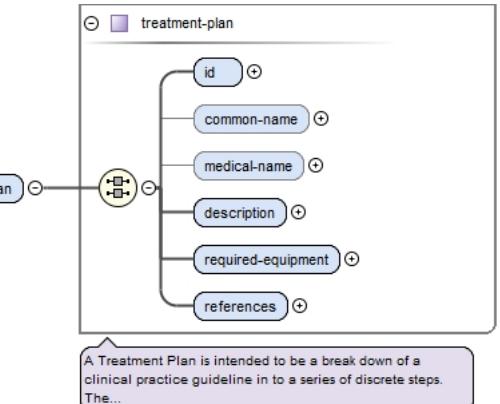
Namespace	No namespace
Diagram	
Type	xs:NCName
Properties	content: simple minOccurs: 2 maxOccurs: unbounded
Source	<xs:element name="token" type="xs:NCName" minOccurs="2" maxOccurs="unbounded" />

**Element ScenarioSchema / treatment-plans**

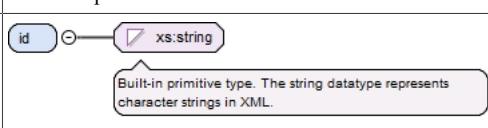
Namespace	No namespace
Annotations	XML Database of Treatment Plans for the injuries contained in the injury database.

Diagram	
Type	treatment-plan-definition-list
Properties	content: complex
Model	treatment-plan+
Children	treatment-plan
Instance	<pre>&lt;treatment-plans&gt;   &lt;treatment-plan&gt;{1,unbounded}&lt;/treatment-plan&gt; &lt;/treatment-plans&gt;</pre>
Source	<pre>&lt;xs:element name="treatment-plans" type="pfc:treatment-plan-definition-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;XML Database of Treatment Plans for the injuries contained in the injury database.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element treatment-plan-definition-list / treatment-plan

Namespace	No namespace
Diagram	
Type	treatment-plan
Properties	<p>content: complex</p> <p>minOccurs: 1</p> <p>maxOccurs: unbounded</p>
Model	ALL(id common-name{0,1} medical-name{0,1} description required-equipment references)
Children	common-name, description, id, medical-name, references, required-equipment
Instance	<pre>&lt;treatment-plan&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;common-name&gt;{0,1}&lt;/common-name&gt;   &lt;medical-name&gt;{0,1}&lt;/medical-name&gt;   &lt;description&gt;{1,1}&lt;/description&gt;   &lt;required-equipment&gt;{1,1}&lt;/required-equipment&gt;   &lt;references&gt;{1,1}&lt;/references&gt; &lt;/treatment-plan&gt;</pre>
Source	<pre>&lt;xs:element name="treatment-plan" type="treatment-plan" minOccurs="1" maxOccurs="unbounded" /&gt;</pre>

### Element treatment-plan / id

Namespace	No namespace
Diagram	
Type	xs:string

Properties	content: simple
Source	<xss:element name="id" type="xs:string"/>

**Element treatment-plan / common-name**

Namespace	No namespace
Diagram	<p>The diagram shows the element 'common-name' with a multiplicity of 0..1. It is connected to a box labeled 'xs:string'. A callout bubble indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xss:element name="common-name" type="xs:string" minOccurs="0"/>

**Element treatment-plan / medical-name**

Namespace	No namespace
Diagram	<p>The diagram shows the element 'medical-name' with a multiplicity of 0..1. It is connected to a box labeled 'xs:string'. A callout bubble indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xss:element name="medical-name" type="xs:string" minOccurs="0"/>

**Element treatment-plan / description**

Namespace	No namespace
Diagram	<p>The diagram shows the element 'description' with a multiplicity of 0..1. It is connected to a box labeled 'xs:string'. A callout bubble indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xss:element name="description" type="xs:string" minOccurs="1"/>

**Element treatment-plan / required-equipment**

Namespace	No namespace
Diagram	<p>The diagram shows the element 'required-equipment' with a multiplicity of 0..1. It is connected to a box labeled 'equipment-ref-list'. Inside 'equipment-ref-list', there is a box labeled 'equipment-refs' with a multiplicity of 0..∞. A callout bubble indicates: 'List of UIDs from the equipment list in the FORCE_SIDE section of an associated MSDL file'</p>
Type	equipment-ref-list
Properties	content: complex minOccurs: 1
Model	equipment-refs*
Children	equipment-refs
Instance	<pre>&lt;required-equipment&gt;   &lt;equipment-refs&gt;{0,unbounded}&lt;/equipment-refs&gt; &lt;/required-equipment&gt;</pre>

## Source

```
<xss:element name="required-equipment" type="equipment-ref-list" minOccurs="1"/>
```

**Element equipment-ref-list / equipment-refs**

Namespace	No namespace						
Diagram	<p>The diagram shows the element 'equipment-refs' represented as a box containing the text 'xs:string'. A callout box indicates that this is a built-in primitive type representing character strings in XML.</p>						
Type	xs:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Source	<pre>&lt;xss:element name="equipment-refs" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;</pre>						

**Element treatment-plan / references**

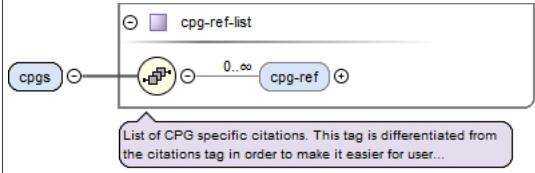
Namespace	No namespace
Diagram	<p>The diagram shows the element 'references' represented as a box containing a 'medical-reference-list' box. This list contains 'citations' and 'cpgs' elements, each with a multiplicity of 0..1. A callout box states that 'citations' are a series of citations built on, allowing users to refer to more detailed material.</p>
Type	medical-reference-list
Properties	content: complex
Model	ALL(citations cpgs)
Children	citations, cpgs
Instance	<pre>&lt;references&gt;   &lt;citations&gt;{1,1}&lt;/citations&gt;   &lt;cpgs&gt;{1,1}&lt;/cpgs&gt; &lt;/references&gt;</pre>
Source	<pre>&lt;xss:element name="references" type="medical-reference-list"/&gt;</pre>

**Element medical-reference-list / citations**

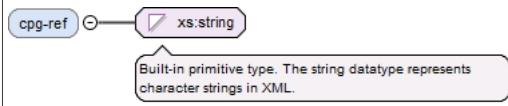
Namespace	No namespace
Diagram	<p>The diagram shows the element 'citations' represented as a box containing a 'citation-ref-list' box. This list contains 'citation-ref' elements, each with a multiplicity of 0..1. A callout box notes that this is a list of strings for source material and could be expanded.</p>
Type	citation-ref-list
Properties	content: complex
Model	citation-ref*
Children	citation-ref
Instance	<pre>&lt;citations&gt;   &lt;citation-ref&gt;{0, unbounded}&lt;/citation-ref&gt; &lt;/citations&gt;</pre>
Source	<pre>&lt;xss:element name="citations" type="citation-ref-list"/&gt;</pre>

**Element medical-reference-list / cpgs**

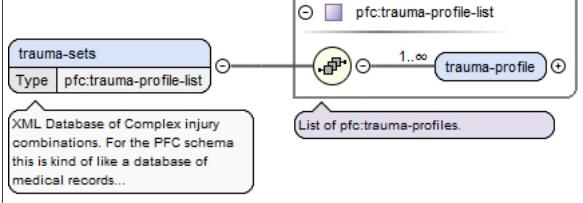
Namespace	No namespace
-----------	--------------

Diagram	 <p>List of CPG specific citations. This tag is differentiated from the citations tag in order to make it easier for user...</p>
Type	cpg-ref-list
Properties	content: complex
Model	cpg-ref*
Children	cpg-ref
Instance	<cpgs> <cpg-ref>{0 ,unbounded}</cpg-ref> </cpgs>
Source	<xss:element name="cpgs" type="cpg-ref-list"/>

### Element cpg-ref-list / cpg-ref

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Source	<xss:element name="cpg-ref" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>

### Element ScenarioSchema / trauma-sets

Namespace	No namespace
Annotations	XML Database of Complex injury combinations. For the PFC schema this is kind of like a database of medical records which can be applied to patients. It allows the same set of injuries to be applied to multiple patients with minimal documentation.
Diagram	 <p>XML Database of Complex injury combinations. For the PFC schema this is kind of like a database of medical records...</p> <p>List of pfc:trauma-profiles.</p>
Type	trauma-profile-list
Properties	content: complex
Model	trauma-profile+
Children	trauma-profile
Instance	<trauma-sets> <trauma-profile>{1,unbounded}</trauma-profile> </trauma-sets>
Source	<xss:element name="trauma-sets" type="pfc:trauma-profile-list"> <xss:annotation> <xss:documentation>XML Database of Complex injury combinations. For the PFC schema this is kind of like a database of medical records which can be applied to patients. It allows the same set of injuries to be applied to multiple patients with minimal documentation.</xss:documentation> </xss:annotation> </xss:element>

### Element trauma-profile-list / trauma-profile

Namespace	No namespace
-----------	--------------

Diagram	
Type	trauma-profile
Properties	<p>content: complex</p> <p>minOccurs: 1</p> <p>maxOccurs: unbounded</p>
Model	id , name , physiology-state{0,1} , injuries , treatments
Children	id, injuries, name, physiology-state, treatments
Instance	<pre>&lt;trauma-profile&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;name&gt;{1,1}&lt;/name&gt;   &lt;physiology-state&gt;{0,1}&lt;/physiology-state&gt;   &lt;injuries&gt;{1,1}&lt;/injuries&gt;   &lt;treatments&gt;{1,1}&lt;/treatments&gt; &lt;/trauma-profile&gt;</pre>
Source	<code>&lt;xss:element name="trauma-profile" type="trauma-profile" minOccurs="1" maxOccurs="unbounded"/&gt;</code>

**Element trauma-profile / id**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 1</p>
Source	<code>&lt;xss:element name="id" type="xs:string" minOccurs="1"/&gt;</code>

**Element trauma-profile / name**

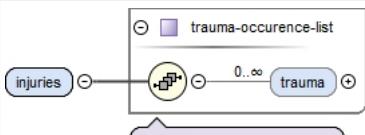
Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 1</p>
Source	<code>&lt;xss:element name="name" type="xs:string" minOccurs="1"/&gt;</code>

**Element trauma-profile / physiology-state**

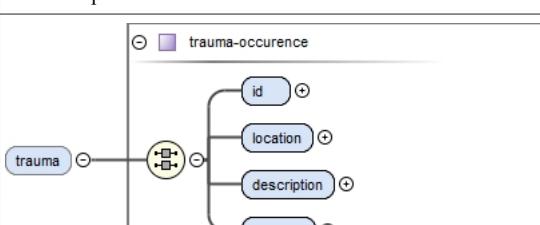
Namespace	No namespace
Diagram	
Type	xs:string

Properties	content: simple minOccurs: 0
Source	<xs:element name="physiology-state" type="xs:string" minOccurs="0" />

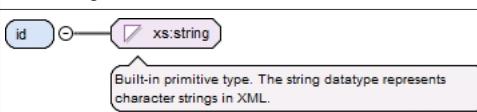
**Element trauma-profile / injuries**

Namespace	No namespace
Diagram	 <p>The diagram illustrates the relationship between the 'injuries' element and the 'trauma-occurrence-list' element. An association line connects 'injuries' to 'trauma-occurrence-list'. The multiplicity '0..∞' is indicated at the 'trauma-occurrence-list' end, and a plus sign (+) is at the 'injuries' end. The 'trauma-occurrence-list' element is shown as a box containing a list of 'trauma' elements, with a callout 'List of pfc:trauma-occurrence' pointing to it.</p>
Type	trauma-occurrence-list
Properties	content: complex minOccurs: 1
Model	trauma*
Children	trauma
Instance	<injuries> <trauma>{0,unbounded}</trauma> </injuries>
Source	<xs:element name="injuries" type="trauma-occurrence-list" minOccurs="1" />

**Element trauma-occurrence-list / trauma**

Namespace	No namespace
Diagram	 <p>The diagram shows the 'trauma' element associated with the 'trauma-occurrence' element. The multiplicity '0' is at the 'trauma' end, and a plus sign (+) is at the 'trauma-occurrence' end. The 'trauma-occurrence' element is detailed as a box containing fields: 'id', 'location', 'description', and 'severity'. A callout box states: 'Injury occurrence is an Injury-Reference with a few additional fields to help associate the trauma with a specific...'.</p>
Type	trauma-occurrence
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(id location description severity)
Children	description, id, location, severity
Instance	<trauma> <id>{1,1}</id> <location>{1,1}</location> <description>{1,1}</description> <severity>{1,1}</severity> </trauma>
Source	<xs:element name="trauma" type="trauma-occurrence" minOccurs="0" maxOccurs="unbounded" />

**Element trauma-occurrence / id**

Namespace	No namespace
Diagram	 <p>The diagram shows the 'id' element associated with the 'xs:string' primitive type. A callout box states: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string

Properties	content: simple minOccurs: 1
Source	<xs:element name="id" type="xs:string" minOccurs="1"/>

**Element trauma-occurrence / location**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xs:element name="location" type="xs:string" minOccurs="1"/>

**Element trauma-occurrence / description**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xs:element name="description" type="xs:string" minOccurs="1"/>

**Element trauma-occurrence / severity**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple minOccurs: 1
Source	<xs:element name="severity" type="xs:string" minOccurs="1"/>

**Element trauma-profile / treatments**

Namespace	No namespace
Diagram	<p>List of UIDs from the treatment-plans in the treatment-definition-list</p>
Type	treatment-plan-ref-list
Properties	content: complex
Model	treatment-plan*
Children	treatment-plan
Instance	<treatments> <treatment-plan>{0,unbounded}</treatment-plan> </treatments>

Source	<code>&lt;xss:element name="treatments" type="treatment-plan-ref-list"/&gt;</code>
--------	--

## Element treatment-plan-ref-list / treatment-plan

Namespace	No namespace						
Diagram	<pre> classDiagram     class treatment-plan {         &lt;&lt;xs:string&gt;&gt;     }     xs:string &lt; -- treatment-plan   </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	xs:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Source	<code>&lt;xss:element name="treatment-plan" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;</code>						

## Element ScenarioSchema / syllabus

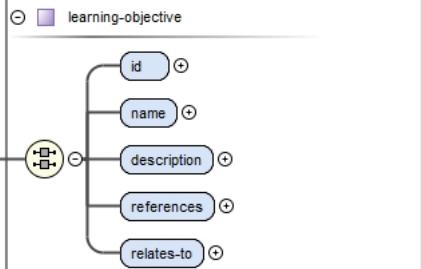
Namespace	No namespace
Annotations	A breakdown of learning-objectives and the accompanying assessment-criteria.
Diagram	<pre> classDiagram     class syllabus {         &lt;&lt;pfc:medical-scenario-syllabus&gt;&gt;     }     pfc:medical-scenario-syllabus &lt; -- syllabus     syllabus --&gt; learning-objectives     syllabus --&gt; learning-assessments   </pre> <p>A breakdown of learning-objectives and the accompanying assessment-criteria.</p>
Type	medical-scenario-syllabus
Properties	content: complex
Model	learning-objectives , learning-assessments
Children	learning-assessments, learning-objectives
Instance	<pre> &lt;syllabus&gt;   &lt;learning-objectives&gt;{1,1}&lt;/learning-objectives&gt;   &lt;learning-assessments&gt;{1,1}&lt;/learning-assessments&gt; &lt;/syllabus&gt;   </pre>
Source	<code>&lt;xss:element name="syllabus" type="pfc:medical-scenario-syllabus"&gt;</code> <code>&lt;xss:annotation&gt;</code> <code>&lt;xss:documentation&gt;A breakdown of learning-objectives and the accompanying assessment-criteria.&lt;/xss:documentation&gt;</code> <code>&lt;/xss:annotation&gt;</code> <code>&lt;/xss:element&gt;</code>

## Element medical-scenario-syllabus / learning-objectives

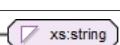
Namespace	No namespace
Diagram	<pre> classDiagram     class learning-objective-list {         &lt;&lt;learning-objective&gt;&gt;     }     learning-objective &lt; -- objective     learning-objectives --&gt; learning-objective-list   </pre>
Type	learning-objective-list
Properties	content: complex
Model	objective*
Children	objective
Instance	<pre> &lt;learning-objectives&gt;   &lt;objective&gt;{0,unbounded}&lt;/objective&gt; &lt;/learning-objectives&gt;   </pre>
Source	<code>&lt;xss:element name="learning-objectives" type="learning-objective-list"/&gt;</code>

## Element learning-objective-list / objective

Namespace	No namespace
-----------	--------------

Diagram	 <p>Objectives will be entered as goals to be met upon completion of a scenario. Fields: ID: xs:string : Identification...</p>
Type	learning-objective
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	ALL(id name description references relates-to)
Children	description, id, name, references, relates-to
Instance	<pre>&lt;objective&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;name&gt;{1,1}&lt;/name&gt;   &lt;description&gt;{1,1}&lt;/description&gt;   &lt;references&gt;{1,1}&lt;/references&gt;   &lt;relates-to&gt;{1,1}&lt;/relates-to&gt; &lt;/objective&gt;</pre>
Source	<code>&lt;xss:element name="objective" type="learning-objective" minOccurs="0" maxOccurs="unbounded"/&gt;</code>

### Element learning-objective / id

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="id" type="xs:string"/&gt;</code>

### Element learning-objective / name

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="name" type="xs:string"/&gt;</code>

### Element learning-objective / description

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="description" type="xs:string"/&gt;</code>

## Element learning-objective / references

Namespace	No namespace
Diagram	<pre> graph TD     A([medical-reference-list]) --&gt; B([references])     A --&gt; C([citations])     A --&gt; D([cpgs])     </pre> <p>Series of citations that the treatment plan was built on. Allows the user to refer to more detailed material beyond the...</p>
Type	medical-reference-list
Properties	content: complex
Model	ALL(citations cpgs)
Children	citations, cpgs
Instance	<pre> &lt;references&gt;     &lt;citations&gt;{1,1}&lt;/citations&gt;     &lt;cpgs&gt;{1,1}&lt;/cpgs&gt; &lt;/references&gt; </pre>
Source	<pre>&lt;xss:element name="references" type="medical-reference-list"/&gt;</pre>

## Element learning-objective / relates-to

Namespace	No namespace
Diagram	<pre> graph TD     A([relates-to]) --&gt; B([treatment-plans])     A --&gt; C([trauma-profiles])     </pre> <p>List of UIDs from the treatment-plans in the treatment-definition-list</p>
Properties	content: complex
Model	ALL(treatment-plans trauma-profiles)
Children	treatment-plans, trauma-profiles
Instance	<pre> &lt;relates-to&gt;     &lt;treatment-plans&gt;{1,1}&lt;/treatment-plans&gt;     &lt;trauma-profiles&gt;{1,1}&lt;/trauma-profiles&gt; &lt;/relates-to&gt; </pre>
Source	<pre>&lt;xss:element name="relates-to"&gt;     &lt;xss:complexType&gt;         &lt;xss:all&gt;             &lt;xss:element name="treatment-plans" type="treatment-plan-ref-list"/&gt;             &lt;xss:element name="trauma-profiles" type="trauma-profile-ref-list"/&gt;         &lt;/xss:all&gt;     &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>

## Element learning-objective / relates-to / treatment-plans

Namespace	No namespace
Diagram	<pre> graph TD     A([treatment-plan-ref-list]) --&gt; B([treatment-plans])     A --&gt; C([treatment-plan])     </pre> <p>0..∞</p> <p>List of UIDs from the treatment-plans in the treatment-definition-list</p>
Type	treatment-plan-ref-list
Properties	content: complex
Model	treatment-plan*
Children	treatment-plan
Instance	<pre> &lt;treatment-plans&gt;     &lt;treatment-plan&gt;{0,unbounded}&lt;/treatment-plan&gt; &lt;/treatment-plans&gt; </pre>
Source	<pre>&lt;xss:element name="treatment-plans" type="treatment-plan-ref-list"/&gt;</pre>

**Element learning-objective / relates-to / trauma-profiles**

Namespace	No namespace
Diagram	<p>Diagram illustrating the relationship between <code>trauma-profiles</code> and <code>trauma-profile-ref-list</code>. The <code>trauma-profiles</code> element is connected to the <code>trauma-profile-ref-list</code> element via a many-to-many relationship. The <code>trauma-profile-ref-list</code> element contains a list of UUIDs from the <code>trauma-profiles</code> in the profile list.</p>
Type	trauma-profile-ref-list
Properties	content: complex
Model	trauma-profile*
Children	trauma-profile
Instance	<pre>&lt;trauma-profiles&gt;   &lt;trauma-profile&gt;{0,unbounded}&lt;/trauma-profile&gt; &lt;/trauma-profiles&gt;</pre>
Source	<code>&lt;xss:element name="trauma-profiles" type="trauma-profile-ref-list"/&gt;</code>

**Element trauma-profile-ref-list / trauma-profile**

Namespace	No namespace
Diagram	<p>Diagram illustrating the relationship between <code>trauma-profile</code> and <code>xs:string</code>. The <code>trauma-profile</code> element is connected to the <code>xs:string</code> element via a simple relationship.</p>
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Source	<code>&lt;xss:element name="trauma-profile" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;</code>

**Element medical-scenario-syllabus / learning-assessments**

Namespace	No namespace
Diagram	<p>Diagram illustrating the relationship between <code>learning-assessments</code> and <code>assessment-list</code>. The <code>learning-assessments</code> element is connected to the <code>assessment-list</code> element via a many-to-many relationship. The <code>assessment-list</code> element contains a list of <code>pfc:assessment</code> with a required <code>total-points</code> field which should equal the sum of all points available in the...</p>
Type	assessment-list
Properties	content: complex
Model	total-points , assessment*
Children	assessment, total-points
Instance	<pre>&lt;learning-assessments&gt;   &lt;total-points&gt;{1,1}&lt;/total-points&gt;   &lt;assessment&gt;{0,unbounded}&lt;/assessment&gt; &lt;/learning-assessments&gt;</pre>
Source	<code>&lt;xss:element name="learning-assessments" type="assessment-list"/&gt;</code>

**Element assessment-list / total-points**

Namespace	No namespace
Diagram	<p>Diagram illustrating the relationship between <code>total-points</code> and <code>xs:integer</code>. The <code>total-points</code> element is connected to the <code>xs:integer</code> element via a simple relationship.</p>
Type	xs:integer

Properties	content: simple minOccurs: 1 maxOccurs: 1 default: 0
Source	<xs:element name="total-points" default="0" type="xs:integer" minOccurs="1" maxOccurs="1"/>

**Element assessment-list / assessment**

Namespace	No namespace
Diagram	
Type	assessment
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(id objective-id name description points-available criteria)
Children	criteria, description, id, name, objective-id, points-available
Instance	<assessment> <id>{1,1}</id> <objective-id>{1,1}</objective-id> <name>{1,1}</name> <description>{1,1}</description> <points-available>{1,1}</points-available> <criteria>{1,1}</criteria> </assessment>
Source	<xs:element name="assessment" type="assessment" minOccurs="0" maxOccurs="unbounded"/>

**Element assessment / id**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="id" type="xs:string"/>

**Element assessment / objective-id**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple

Source	<code>&lt;xs:element name="objective-id" type="xs:string"/&gt;</code>
--------	---

**Element assessment / name**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="name" type="xs:string"/&gt;</code>

**Element assessment / description**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="description" type="xs:string"/&gt;</code>

**Element assessment / points-available**

Namespace	No namespace
Diagram	<p>Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...</p>
Type	xs:integer
Properties	content: simple
Source	<code>&lt;xs:element name="points-available" type="xs:integer"/&gt;</code>

**Element assessment / criteria**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="criteria" type="xs:string"/&gt;</code>

**Element ScenarioSchema / medical-scenario**

Namespace	No namespace
Annotations	The full medical training scenario which combines a MSDL scenario with the PFC extensions to create a training narrative which accomplishes the learning objectives described in the syllabus.
Diagram	<p>The full medical training scenario which combines a MSDL scenario with the PFC extensions to create a training...</p> <p>A PFC medical scenario is an extension of the MSDL. The purpose of this extension is to better document micro level...</p>

Type	medical-scenario
Properties	content: complex
Model	ALL(id roles training-script)
Children	id, roles, training-script
Instance	<pre>&lt;medical-scenario&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;roles&gt;{1,1}&lt;/roles&gt;   &lt;training-script&gt;{1,1}&lt;/training-script&gt; &lt;/medical-scenario&gt;</pre>
Source	<pre>&lt;xs:element name="medical-scenario" type="pfc:medical-scenario"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The full medical training scenario which combines a MSDL scenario with the PFC extensions to create a training narrative which accomplishes the learning objectives described in the syllabus.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element medical-scenario / id

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<pre>&lt;xs:element name="id" type="xs:string"/&gt;</pre>

### Element medical-scenario / roles

Namespace	No namespace
Diagram	<p>List of pfc:roles</p>
Type	role-list
Properties	content: complex
Model	role*
Children	role
Instance	<pre>&lt;roles&gt;   &lt;role&gt;{0,unbounded}&lt;/role&gt; &lt;/roles&gt;</pre>
Source	<pre>&lt;xs:element name="roles" type="role-list"/&gt;</pre>

### Element role-list / role

Namespace	No namespace
Diagram	<p>A role is a reference to a pfc:role with an optional pfc:trauma-profile reference. An additional description describes...</p>

Type	role
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	ALL(id name short-name trauma-profile-ref{0,1} description)
Children	description, id, name, short-name, trauma-profile-ref
Instance	<pre>&lt;role&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;name&gt;{1,1}&lt;/name&gt;   &lt;short-name&gt;{1,1}&lt;/short-name&gt;   &lt;trauma-profile-ref&gt;{0,1}&lt;/trauma-profile-ref&gt;   &lt;description&gt;{1,1}&lt;/description&gt; &lt;/role&gt;</pre>
Source	<code>&lt;xss:element name="role" type="role" minOccurs="0" maxOccurs="unbounded" /&gt;</code>

### Element role / id

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="id" type="xs:string" /&gt;</code>

### Element role / name

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="name" type="xs:string" /&gt;</code>

### Element role / short-name

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="short-name" type="xs:string" /&gt;</code>

### Element role / trauma-profile-ref

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>

	maxOccurs:	1
Source	<xs:element name="trauma-profile-ref" type="xs:string" minOccurs="0" maxOccurs="1"/>	

### Element **role** / **description**

Namespace	No namespace
Diagram	<p>The diagram shows the 'role' element as a class with a multiplicity of 0..1. It has a directed association labeled 'xs:string' pointing to the built-in primitive type 'xs:string'. A callout box indicates that 'xs:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple
Source	<xs:element name="role" type="xs:string" />

### Element **medical-scenario** / **training-script**

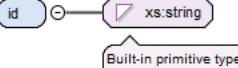
Namespace	No namespace
Diagram	<p>The diagram shows the 'medical-scenario' element as a class named 'scene-list'. It has a directed association labeled 'training-script' with multiplicity 0..1. This association points to another class named 'training-script'. From 'training-script', there is a directed association labeled '1..∞' pointing to a class named 'scene'. A callout box indicates that this represents a 'List of pfc:scene(s)'.</p>
Type	scene-list
Properties	content: complex
Model	scene+
Children	scene
Instance	<training-script><scene>{1,unbounded}</scene></training-script>
Source	<xs:element name="training-script" type="scene-list" />

### Element **scene-list** / **scene**

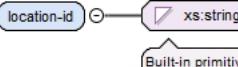
Namespace	No namespace
Diagram	<p>The diagram shows the 'scene-list' element as a class named 'scene'. It has a directed association labeled 'scene' with multiplicity 0..1. This association points to another class named 'scene'. The 'scene' class is shown with a large list of attributes: id, location-id, name, description, details, weather, time-of-day, time-in-simulation, events, items, and roles. Each attribute has a multiplicity of 0..1. A callout box indicates that 'A scene is a reference msdl:installation with its own independent meteorological data and time of day. Each scene also...'.</p>
Type	scene
Properties	content: complex

	minOccurs: 1 maxOccurs: unbounded
Model	ALL(id location-id name description details{0,1} weather{0,1} time-of-day time-in-simulation events items roles)
Children	description, details, events, id, items, location-id, name, roles, time-in-simulation, time-of-day, weather
Instance	<pre>&lt;scene&gt;   &lt;id&gt;{1,1}&lt;/id&gt;   &lt;location-id&gt;{1,1}&lt;/location-id&gt;   &lt;name&gt;{1,1}&lt;/name&gt;   &lt;description&gt;{1,1}&lt;/description&gt;   &lt;details&gt;{0,1}&lt;/details&gt;   &lt;weather&gt;{0,1}&lt;/weather&gt;   &lt;time-of-day&gt;{1,1}&lt;/time-of-day&gt;   &lt;time-in-simulation&gt;{1,1}&lt;/time-in-simulation&gt;   &lt;events&gt;{1,1}&lt;/events&gt;   &lt;items&gt;{1,1}&lt;/items&gt;   &lt;roles&gt;{1,1}&lt;/roles&gt; &lt;/scene&gt;</pre>
Source	<code>&lt;xs:element name="scene" type="scene" minOccurs="1" maxOccurs="unbounded" /&gt;</code>

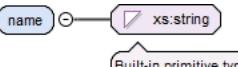
## Element scene / id

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="id" type="xs:string" /&gt;</code>

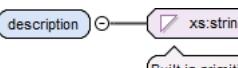
## Element scene / location-id

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="location-id" type="xs:string" /&gt;</code>

## Element scene / name

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="name" type="xs:string" /&gt;</code>

## Element scene / description

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string

Properties	content: simple
Source	<xs:element name="description" type="xs:string" />

### Element scene / details

Namespace	No namespace				
Diagram	<pre> graph LR     details[details] --&gt; xsString[xs:string]     subgraph Info [ ]         direction TB         B["Built-in primitive type. The string datatype represents character strings in XML."]     end     </pre>				
Type	xs:string				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xs:element name="details" type="xs:string" minOccurs="0" />				

### Element scene / weather

Namespace	No namespace				
Diagram	<pre> graph LR     weather[weather] --&gt; xsString[xs:string]     subgraph Info [ ]         direction TB         B["Built-in primitive type. The string datatype represents character strings in XML."]     end     </pre>				
Type	xs:string				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xs:element name="weather" type="xs:string" minOccurs="0" />				

### Element scene / time-of-day

Namespace	No namespace		
Diagram	<pre> graph LR     timeOfDay[time-of-day] --&gt; xsInteger[xs:integer]     subgraph Info [ ]         direction TB         B["Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This..."]     end     </pre>		
Type	xs:integer		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<xs:element name="time-of-day" type="xs:integer" />		

### Element scene / time-in-simulation

Namespace	No namespace		
Diagram	<pre> graph LR     timeInSimulation[time-in-simulation] --&gt; xsInteger[xs:integer]     subgraph Info [ ]         direction TB         B["Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This..."]     end     </pre>		
Type	xs:integer		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<xs:element name="time-in-simulation" type="xs:integer" />		

### Element scene / events

Namespace	No namespace
Diagram	<pre> graph LR     events[events] --&gt; eventList[event-list]     eventList --&gt; event[event]     subgraph Info [ ]         direction TB         B["List of pfc:event(s)"]     end     </pre>

Type	event-list
Properties	content: complex
Model	event+
Children	event
Instance	<events> <event>{1,unbounded}</event> </events>
Source	<xs:element name="events" type="event-list"/>

## Element event-list / event

Namespace	No namespace
Diagram	<p>An event is a pfc:event-category-enum a pfc:event-fidelity-enum and a set of details. An additional description...</p>
Type	event
Properties	content: complex minOccurs: 1 maxOccurs: unbounded
Model	ALL(id name category fidelity actor_1 actor_2 equipment details description)
Children	actor_1, actor_2, category, description, details, equipment, fidelity, id, name
Instance	<event> <id>{1,1}</id> <name>{1,1}</name> <category>{1,1}</category> <fidelity>{1,1}</fidelity> <actor_1>{1,1}</actor_1> <actor_2>{1,1}</actor_2> <equipment>{1,1}</equipment> <details>{1,1}</details> <description>{1,1}</description> </event>
Source	<xs:element name="event" type="event" minOccurs="1" maxOccurs="unbounded"/>

## Element event / id

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<xs:element name="id" type="xs:string"/>

## Element event / name

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<xs:element name="name" type="xs:string"/>

## Element event / category

Namespace	No namespace															
Diagram	 <p>Enumerated choice of an pfc:event category each of which describes one type of action that can occur in this event</p>															
Type	event-category-enum															
Properties	content: simple															
Facets	<table> <tr> <td>enumeration</td> <td>ACTION</td> <td>An Action is the result of a Role acting upon a piece of equipment or another Role. Can be detailed at all fidelity levels</td> </tr> <tr> <td>enumeration</td> <td>DIALOG</td> <td>Dialog is anytime a Role or Object performs audible dialog in the scene. Typically for Medium or Higher fidelity.</td> </tr> <tr> <td>enumeration</td> <td>MOVEMENT</td> <td>Any time a Role or Object changes location in the scene .</td> </tr> <tr> <td>enumeration</td> <td>SOUND</td> <td>Background or ambient sound affects typically at a high fidelity level as most low and medium fidelity implementations do not have sound systems.</td> </tr> <tr> <td>enumeration</td> <td>ENVIRONMENT</td> <td>Typically used to describe changes in set design or weather events almost always reserved for high fidelity simulations.</td> </tr> </table>	enumeration	ACTION	An Action is the result of a Role acting upon a piece of equipment or another Role. Can be detailed at all fidelity levels	enumeration	DIALOG	Dialog is anytime a Role or Object performs audible dialog in the scene. Typically for Medium or Higher fidelity.	enumeration	MOVEMENT	Any time a Role or Object changes location in the scene .	enumeration	SOUND	Background or ambient sound affects typically at a high fidelity level as most low and medium fidelity implementations do not have sound systems.	enumeration	ENVIRONMENT	Typically used to describe changes in set design or weather events almost always reserved for high fidelity simulations.
enumeration	ACTION	An Action is the result of a Role acting upon a piece of equipment or another Role. Can be detailed at all fidelity levels														
enumeration	DIALOG	Dialog is anytime a Role or Object performs audible dialog in the scene. Typically for Medium or Higher fidelity.														
enumeration	MOVEMENT	Any time a Role or Object changes location in the scene .														
enumeration	SOUND	Background or ambient sound affects typically at a high fidelity level as most low and medium fidelity implementations do not have sound systems.														
enumeration	ENVIRONMENT	Typically used to describe changes in set design or weather events almost always reserved for high fidelity simulations.														
Source	<xs:element name="category" type="event-category-enum"/>															

## Element event / fidelity

Namespace	No namespace									
Diagram	 <p>Enumerated choice of an pfc:event fidelity is intended to be used as a guide by implementations on if the details of...</p>									
Type	event-fidelity-enum									
Properties	content: simple									
Facets	<table> <tr> <td>enumeration</td> <td>LOW</td> <td>Low fidelity implementations are intended for traditional medical simulators which use live actors and need only to know the high level details like initial injuries and placement of equipment in the scene.</td> </tr> <tr> <td>enumeration</td> <td>MEDIUM</td> <td>Medium fidelity implementations are higher level simulations which include some scripted dialog and movement. The details included at this level give non trainees instructions on how to act when not interacting directly with the trainee</td> </tr> <tr> <td>enumeration</td> <td>HIGH</td> <td>High fidelity simulations are typically fully virtual simulations which must create atmospheric and background effects to enhance immersion. We are truly in the weeds at this level of scripting.</td> </tr> </table>	enumeration	LOW	Low fidelity implementations are intended for traditional medical simulators which use live actors and need only to know the high level details like initial injuries and placement of equipment in the scene.	enumeration	MEDIUM	Medium fidelity implementations are higher level simulations which include some scripted dialog and movement. The details included at this level give non trainees instructions on how to act when not interacting directly with the trainee	enumeration	HIGH	High fidelity simulations are typically fully virtual simulations which must create atmospheric and background effects to enhance immersion. We are truly in the weeds at this level of scripting.
enumeration	LOW	Low fidelity implementations are intended for traditional medical simulators which use live actors and need only to know the high level details like initial injuries and placement of equipment in the scene.								
enumeration	MEDIUM	Medium fidelity implementations are higher level simulations which include some scripted dialog and movement. The details included at this level give non trainees instructions on how to act when not interacting directly with the trainee								
enumeration	HIGH	High fidelity simulations are typically fully virtual simulations which must create atmospheric and background effects to enhance immersion. We are truly in the weeds at this level of scripting.								
Source	<xs:element name="fidelity" type="event-fidelity-enum"/>									

**Element event / actor\_1**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="actor_1" type="xs:string"/&gt;</code>

**Element event / actor\_2**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="actor_2" type="xs:string"/&gt;</code>

**Element event / equipment**

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="equipment" type="xs:string"/&gt;</code>

**Element event / details**

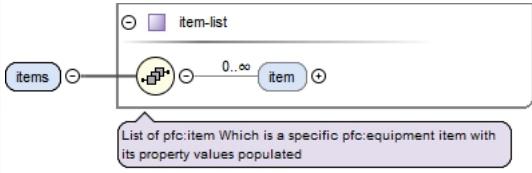
Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="details" type="xs:string"/&gt;</code>

**Element event / description**

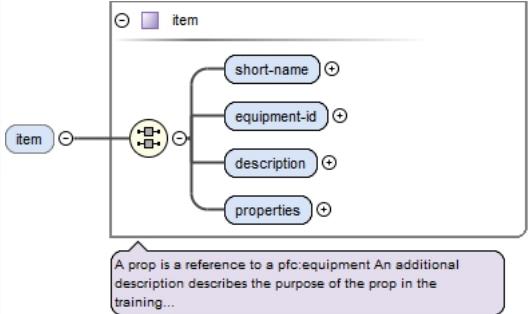
Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<code>&lt;xs:element name="description" type="xs:string"/&gt;</code>

**Element scene / items**

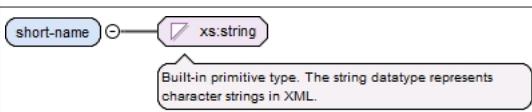
Namespace	No namespace
-----------	--------------

Diagram	
Type	item-list
Properties	content: complex
Model	item*
Children	item
Instance	<items> <item>{0,unbounded}</item> </items>
Source	<xs:element name="items" type="item-list"/>

### Element item-list / item

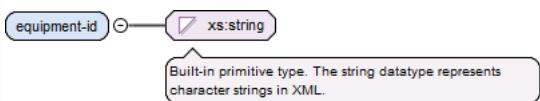
Namespace	No namespace
Diagram	
Type	item
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(short-name equipment-id description properties)
Children	description, equipment-id, properties, short-name
Instance	<item> <short-name>{1,1}</short-name> <equipment-id>{1,1}</equipment-id> <description>{1,1}</description> <properties>{1,1}</properties> </item>
Source	<xs:element name="item" type="item" minOccurs="0" maxOccurs="unbounded"/>

### Element item / short-name

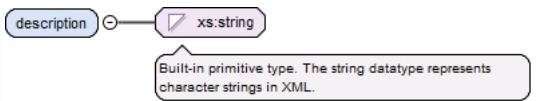
Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="short-name" type="xs:string"/>

### Element item / equipment-id

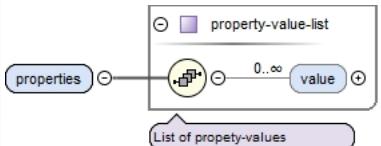
Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="equipment-id" type="xs:string"/>

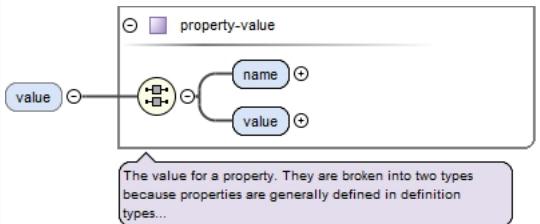
## Element item / description

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="description" type="xs:string"/>

## Element item / properties

Namespace	No namespace
Diagram	
Type	property-value-list
Properties	content: complex
Model	value*
Children	value
Instance	<properties> <value>{0,unbounded}</value> </properties>
Source	<xs:element name="properties" type="property-value-list"/>

## Element property-value-list / value

Namespace	No namespace
Diagram	
Type	property-value
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(name value)
Children	name, value
Instance	<value> <name>{1,1}</name> <value>{1,1}</value> </value>

## Source

```
<xs:element name="value" type="property-value" minOccurs="0" maxOccurs="unbounded" />
```

**Element property-value / name**

Namespace	No namespace
Diagram	<p>A UML class diagram fragment showing an association from a rounded rectangle labeled 'name' to another rounded rectangle labeled 'xs:string'. A callout box indicates that 'xs:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple
Source	<pre>&lt;xs:element name="name" type="xs:string"/&gt;</pre>

**Element property-value / value**

Namespace	No namespace
Diagram	<p>A UML class diagram fragment showing an association from a rounded rectangle labeled 'value' to another rounded rectangle labeled 'xs:string'. A callout box indicates that 'xs:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple
Source	<pre>&lt;xs:element name="value" type="xs:string"/&gt;</pre>

**Element scene / roles**

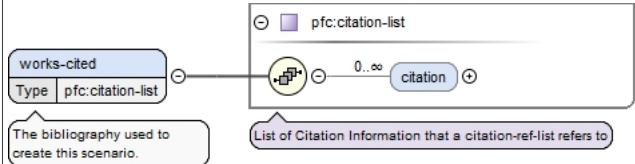
Namespace	No namespace
Diagram	<p>A UML class diagram fragment showing an association from a rounded rectangle labeled 'roles' to a composite structure labeled 'role-ref-list'. This list contains one or more 'role-ref' elements, indicated by a multiplicity of '0..∞' and a self-loop arrow. A callout box indicates that 'role-ref-list' is a 'role-ref*'. A separate callout box indicates that 'role-ref' is a 'role-ref'.</p>
Type	role-ref-list
Properties	content: complex
Model	role-ref*
Children	role-ref
Instance	<pre>&lt;roles&gt;   &lt;role-ref&gt;{0,unbounded}&lt;/role-ref&gt; &lt;/roles&gt;</pre>
Source	<pre>&lt;xs:element name="roles" type="role-ref-list"/&gt;</pre>

**Element role-ref-list / role-ref**

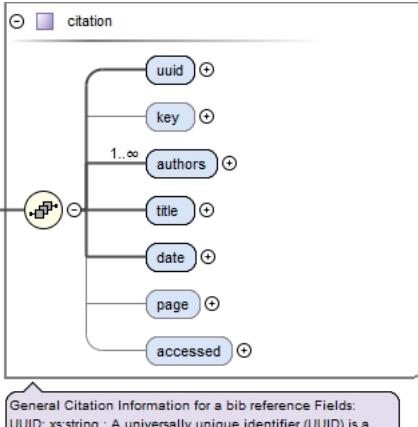
Namespace	No namespace						
Diagram	<p>A UML class diagram fragment showing an association from a rounded rectangle labeled 'role-ref' to another rounded rectangle labeled 'xs:string'. A callout box indicates that 'xs:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>						
Type	xs:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Source	<pre>&lt;xs:element name="role-ref" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;</pre>						

**Element ScenarioSchema / works-cited**

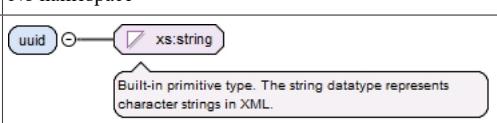
Namespace	No namespace
Annotations	The bibliography used to create this scenario.

Diagram	
Type	citation-list
Properties	content: complex
Model	citation*
Children	citation
Instance	<works-cited> <citation>{0,unbounded}</citation> </works-cited>
Source	<pre>&lt;xs:element name="works-cited" type="pfc:citation-list"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The bibliography used to create this scenario.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element citation-list / citation

Namespace	No namespace
Diagram	 A note below the diagram says 'General Citation Information for a bib reference Fields: UUID: xs:string : A universally unique identifier (UUID) is a...'.
Type	citation
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	uuid , key{0,1} , authors+ , title , date , page{0,1} , accessed{0,1}
Children	accessed, authors, date, key, page, title, uuid
Instance	<citation> <uuid>{1,1}</uuid> <key>{0,1}</key> <authors>{1,unbounded}</authors> <title>{1,1}</title> <date>{1,1}</date> <page>{0,1}</page> <accessed>{0,1}</accessed> </citation>
Source	<pre>&lt;xs:element name="citation" type="citation" minOccurs="0" maxOccurs="unbounded" /&gt;</pre>

### Element citation / uuid

Namespace	No namespace
Diagram	 A note below the diagram says 'Built-in primitive type. The string datatype represents character strings in XML.'

Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 1</p> <p>maxOccurs: 1</p>
Source	<xs:element name="uuid" type="xs:string" minOccurs="1" maxOccurs="1"/>

**Element citation / key**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Source	<xs:element name="key" type="xs:string" minOccurs="0" maxOccurs="1"/>

**Element citation / authors**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 1</p> <p>maxOccurs: unbounded</p>
Source	<xs:element name="authors" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>

**Element citation / title**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 1</p> <p>maxOccurs: 1</p>
Source	<xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/>

**Element citation / date**

Namespace	No namespace
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 1</p>

	maxOccurs:	1
Source	<xs:element name="date" type="xs:string" minOccurs="1" maxOccurs="1"/>	

**Element citation / page**

Namespace	No namespace						
Diagram							
Type	xs:string						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<xs:element name="page" type="xs:string" minOccurs="0" maxOccurs="1"/>						

**Element citation / accessed**

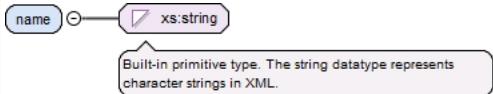
Namespace	No namespace						
Diagram							
Type	xs:string						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<xs:element name="accessed" type="xs:string" minOccurs="0" maxOccurs="1"/>						

**Element cpg-list / cpg**

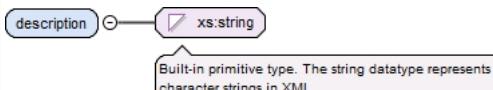
Namespace	No namespace						
Diagram							
Type	cpg						
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	ALL(name description citation-ref)						
Children	citation-ref, description, name						
Instance	<cpg>   <name>{1,1}</name>   <description>{1,1}</description>   <citation-ref>{1,1}</citation-ref> </cpg>						
Source	<xs:element name="cpg" type="cpg" minOccurs="0" maxOccurs="unbounded"/>						

**Element cpg / name**

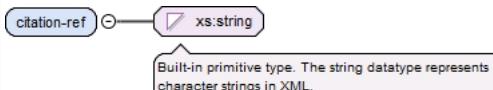
Namespace	No namespace
-----------	--------------

Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	xs:string
Properties	content: simple
Source	<xs:element name="name" type="xs:string"/>

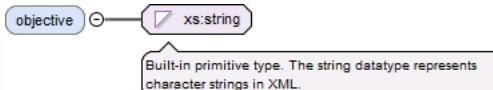
**Element cpg / description**

Namespace	No namespace
Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	xs:string
Properties	content: simple
Source	<xs:element name="description" type="xs:string"/>

**Element cpg / citation-ref**

Namespace	No namespace
Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	xs:string
Properties	content: simple
Source	<xs:element name="citation-ref" type="xs:string"/>

**Element learning-objective-reference-list / objective**

Namespace	No namespace
Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	xs:string
Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Source	<xs:element name="objective" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>