```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
elementFormDefault="qualified" vc:minVersion="1.1">
    <!-- Definition of GUID type -->
    <xs:simpleType name="guidType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- Definition of the root element -->
    <xs:element name="process">
        <xs:complexType>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element ref="action" minOccurs="0" maxOccurs="1"/>
                <xs:element name="appdef" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="endpoint" minOccurs="0">
                    <xs:complexType>
                        <xs:attribute name="narrative" type="xs:string"
use="optional"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="parentobject" minOccurs="0" maxOccurs="1"/>
                <xs:element name="preconditions" minOccurs="0"/>
                <xs:element ref="stage" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="view" minOccurs="0"/>
                <xs:element ref="subsheet" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:choice>
            <xs:attribute name="bpversion" type="xs:string"/>
            <xs:attribute name="byrefcollection" type="xs:string"/>
            <xs:attribute name="disableversioning" type="xs:string"/>
            <xs:attribute name="name" type="xs:string"/>
            <xs:attribute name="narrative" type="xs:string"/>
            <xs:attribute name="preferredid" type="guidType"/>
            <xs:attribute name="processrunningmessage" type="xs:string"/>
            <xs:attribute name="published" type="xs:string"/>
            <xs:attribute name="runmode" type="xs:string"/>
            <xs:attribute name="shared" type="xs:string"/>
            <xs:attribute name="type">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="object"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="version" type="xs:string"/>
```

```
            <!-- Allow any other attributes -->
            <xs:anyAttribute namespace="##any" processContents="skip"/>
            <!--<xs:assert test="count(stage[@type='Start']) >= 1"/>
            <xs:assert test="count(stage[@type='End']) >= 1"/>-->
        </xs:complexType>
    </xs:element>
    <xs:element name="action">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="appdef">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="breakpoint">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="bubbleException">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="calculation">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
```

```
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="choices">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="code">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="collectioninfo">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datatype">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="binary"/>
                <xs:enumeration value="collection"/>
                <xs:enumeration value="date"/>
                <xs:enumeration value="datetime"/>
                <xs:enumeration value="flag"/>
                <xs:enumeration value="image"/>
                <xs:enumeration value="number"/>
                <xs:enumeration value="password"/>
                <xs:enumeration value="text"/>
                <xs:enumeration value="time"/>
                <xs:enumeration value="timespan"/>
                <xs:enumeration value="unknown"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
```

```
    <!--<xs:element name="decision">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>-->
    <xs:element name="decision">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="expression" use="required">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <!--
                                -->
                                <!-- Property Value Comparisons: This regex
matches expressions where a property value is compared to a constant or
another property -->
                                <!--
                                <xs:pattern value="\[[-\w\s]+(\.[-
\w\s]+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;
|\d+|\[[-\w\s]+(\.[-\w\s]+)*\])
"/>
                                -->
                                <!-- Method Call Return Value
Comparisons:Matches method calls (function names followed by
parentheses).Matches comparison operators.Matches boolean
values,            strings, numbers, or other method calls. -->
                                <!--
                                <xs:pattern value="[a-zA-
Z_]\w*\(\s*\)\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&qu
ot;|\d+|[a-zA-Z_]\w*\(\s*\))
"/>
                                -->
                                <!-- Logical Operations (AND): Combines two
property value comparisons with AND between them. -->
                                <!--
                                <xs:pattern value="\[[-\w\s]+(\.[-
\w\s]+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;
|\d+|\[[-\w\s]+(\.[-\w\s]+)*\])\s+AND\s+\[[-\w\s]+(\.[-
\w\s]+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;
|\d+|\[[-\w\s]+(\.[-\w\s]+)*\])
"/>
                                -->
```

```
                                <!-- Logical Operations (OR):Combines two
property value comparisons with OR between them. -->
                                <!--
                                <xs:pattern value="\[[-\w\s]+(\.[-
\w\s]+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;
|\d+|\[[-\w\s]+(\.[-\w\s]+)*\])\s+OR\s+\[[-\w\s]+(\.[-
\w\s]+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;
|\d+|\[[-\w\s]+(\.[-\w\s]+)*\])
"/>
                                -->
                                <!-- Assignments:Matches property assignments
to boolean values, strings, numbers, or other properties. -->
                                <!--
                                <xs:pattern value="\[[-\w\s]+(\.[-
\w\s]+)*\]\s*=\s*(True|False|&quot;[^&quot;]*&quot;|\d+|\[[-\w\s]+(\.[-
\w\s]+)*\])
"/>
                                -->
                                <!-- Combined Pattern. Property Value or
Method Call -->
                                <!--
                                <xs:pattern
value="(\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;
]*&quot;|\d+|\[[^\]]+\]))|([a-zA-Z_]\w*\(\s*\))"/>
                                -->
                                <!--<xs:pattern value="0*"/>        -->
                                <xs:pattern value="\[[^\]]+\]"/>
                                <xs:pattern value="[a-zA-Z_]\w*\(\s*\)"/>
                                <xs:pattern
value="\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]
*&quot;|\d+|\[[^\]]+\])"/>
                                <xs:pattern value="[a-zA-
Z_]\w*\(\s*\)\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&qu
ot;|\d+|[a-zA-Z_]\w*\(\s*\))"/>
                                <xs:pattern
value="\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]
*&quot;|\d+|\[[^\]]+\])\s+AND\s+\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(T
rue|False|&quot;[^&quot;]*&quot;|\d+|\[[^\]]+\])"/>
                                <xs:pattern
value="\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]
*&quot;|\d+|\[[^\]]+\])\s+OR\s+\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(Tr
ue|False|&quot;[^&quot;]*&quot;|\d+|\[[^\]]+\])"/>
                                <xs:pattern
value="\[[^\]]+\]\s*=\s*(True|False|&quot;[^&quot;]*&quot;|\d+|\[[^\]]+\])"/>
                                <xs:pattern value="(\[[^\]]+\])|([a-zA-
Z_]\w*\(\s*\))|(\[[^\]]+\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;
[^&quot;]*&quot;|\d+|\[[^\]]+\]))|([a-zA-
```

```
Z_]\w*\(\s*\)\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;|\d+|[a-zA-Z_]\w*\(\s*\)))"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <!--<xs:element name="decision">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="expression" use="required">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                -->
    <!-- Property Value Comparisons: This regex matches expressions where a property value is compared to a constant or another property -->
    <!--
                                <xs:pattern value=""/>
                                -->
    <!-- Method Call Return Value Comparisons:Matches method calls (function names followed by parentheses).Matches comparison operators.Matches boolean values,            strings, numbers, or other method calls. -->
    <!--
                                <xs:pattern value="[a-zA-Z_]\w*\(\s*\)\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;|\d+|[a-zA-Z_]\w*\(\s*\))"/>
                                -->
    <!-- Logical Operations (AND): Combines two property value comparisons with AND between them. -->
    <!--
                                <xs:pattern value="\[\w+(\.\w+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;|\d+|\[\w+(\.\w+)*\])\s+AND\s+\[\w+(\.\w+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;|\d+|\[\w+(\.\w+)*\])"/>
                                -->
    <!-- Logical Operations (OR):Combines two property value comparisons with OR between them. -->
    <!--
                                <xs:pattern value="\[\w+(\.\w+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;|\d+|\[\w+(\.\w+)*\])\s+OR\s+\[\w+(\.\w+)*\]\s*(=|&lt;&gt;|>|&lt;|&lt;=|>=)\s*(True|False|&quot;[^&quot;]*&quot;|\d+|\[\w+(\.\w+)*\])"/>
                                -->
    <!-- Assignments:Matches property assignments to boolean values, strings, numbers, or other properties. -->
```

```
    <!--
                                <xs:pattern
value="\[\w+(\.\w+)*\]\s*=\s*(True|False|&amp;quot;[^&amp;quot;]*&amp;quot;|\d+|\[\w+(\.\w
+)*\])"/>

                                -->
    <!-- Combined Pattern. Property Value or Method Call -->
    <!--
                                <xs:pattern value="(\[\w+(\.\w+)*\])|([a-zA-
Z_]\w*\(\))"/>

                            -->
    <!--<xs:pattern value="0*"/>        -->
    <!--

                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

-->
<xs:element name="exception">
    <xs:complexType mixed="true">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
</xs:element>
<xs:element name="globalcode">
    <xs:complexType mixed="true">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
    </xs:complexType>
</xs:element>
<xs:element name="imports">
    <xs:complexType mixed="true">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
        </xs:sequence>
```

```
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="initialvalue">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="language" type="xs:string"/>
    <xs:element name="loginhibit">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="loginhibitparameters">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="parentobject">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="postconditions">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
```

```
    </xs:element>
    <xs:element name="preconditions">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="references">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="resource">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="skill">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="step">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="steps">
        <xs:complexType mixed="true">
```

```
            <xs:sequence>
                <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
            </xs:sequence>
            <xs:anyAttribute processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:simpleType name="type">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Action"/>
            <xs:enumeration value="Alert"/>
            <xs:enumeration value="Anchor"/>
            <xs:enumeration value="Block"/>
            <xs:enumeration value="Calculation"/>
            <xs:enumeration value="ChoiceEnd"/>
            <xs:enumeration value="ChoiceStart"/>
            <xs:enumeration value="Code"/>
            <xs:enumeration value="Collection"/>
            <xs:enumeration value="Data"/>
            <xs:enumeration value="Decision"/>
            <xs:enumeration value="End"/>
            <xs:enumeration value="Exception"/>
            <xs:enumeration value="LoopEnd"/>
            <xs:enumeration value="LoopStart"/>
            <xs:enumeration value="MultipleCalculation"/>
            <xs:enumeration value="Navigate"/>
            <xs:enumeration value="Note"/>
            <xs:enumeration value="Process"/>
            <xs:enumeration value="ProcessInfo"/>
            <xs:enumeration value="Read"/>
            <xs:enumeration value="Recover"/>
            <xs:enumeration value="Resume"/>
            <xs:enumeration value="Skill"/>
            <xs:enumeration value="Start"/>
            <xs:enumeration value="SubSheet"/>
            <xs:enumeration value="SubSheetInfo"/>
            <xs:enumeration value="WaitEnd"/>
            <xs:enumeration value="WaitStart"/>
            <xs:enumeration value="Write"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="view">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="camerax" type="xs:decimal" minOccurs="0"/>
                <xs:element name="cameray" type="xs:decimal" minOccurs="0"/>
                <xs:element name="zoom" minOccurs="0">
                    <xs:complexType>
```

```
                        <xs:simpleContent>
                            <xs:extension base="xs:decimal">
                                <xs:attribute name="version" type="xs:string"
use="optional"/>
                            </xs:extension>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="subsheet">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" type="xs:string" minOccurs="0"/>
                <xs:element ref="view" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="published" type="xs:string" use="optional"/>
            <xs:attribute name="subsheetid" type="guidType" use="optional"/>
            <xs:attribute name="type" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="display">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="x" type="xs:decimal" use="optional"/>
                    <xs:attribute name="y" type="xs:decimal" use="optional"/>
                    <xs:attribute name="w" type="xs:decimal" use="optional"/>
                    <xs:attribute name="h" type="xs:decimal" use="optional"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="stage">
        <xs:complexType>
            <xs:sequence>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:element ref="action" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="alert">
                        <xs:complexType>
                            <xs:simpleContent>
                                <xs:extension base="xs:string">
                                    <xs:attribute name="expression"
type="xs:string"/>
                                </xs:extension>
                            </xs:simpleContent>
                        </xs:complexType>
```

```
                    </xs:element>
                    <xs:element name="alwaysinit" type="xs:string"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="attempts" type="xs:integer"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="breakpoint" minOccurs="0"
maxOccurs="1"/>
                    <xs:element ref="bubbleException" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="calculation" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="choices" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="code" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="collectioninfo" minOccurs="0"
maxOccurs="1"/>
                    <xs:element ref="datatype" minOccurs="0" maxOccurs="1"/>
                    <xs:element ref="decision" minOccurs="0" maxOccurs="1"/>
                    <xs:element ref="display" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="displayheight" type="xs:decimal"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="displaywidth" type="xs:decimal"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="displayx" type="xs:decimal"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="displayy" type="xs:decimal"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="exception" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="exposure" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="font" minOccurs="0">
                        <xs:complexType>
                            <xs:attribute name="color" type="xs:string"
use="optional"/>
                            <xs:attribute name="family" type="xs:string"
use="optional"/>
                            <xs:attribute name="size" type="xs:integer"
use="optional"/>
                            <xs:attribute name="style" type="xs:string"
use="optional"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="globalcode" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="groupid" type="guidType" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="imports" minOccurs="0" maxOccurs="1"/>
```

```
                    <xs:element name="initialvalue" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="initialvalueenc" type="xs:string"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="inputs" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="input" minOccurs="0"
maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:attribute name="expr"
type="xs:string" use="optional"/>
                                        <xs:attribute name="friendlyname"
type="xs:string" use="optional"/>
                                        <xs:attribute name="name"
type="xs:string" use="optional"/>
                                        <xs:attribute name="narrative"
type="xs:string" use="optional"/>
                                        <xs:attribute name="stage"
type="xs:string" use="optional"/>
                                        <xs:attribute name="type"
type="xs:string" use="optional"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="language" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="loginhibit" minOccurs="0">
                        <xs:complexType>
                            <xs:simpleContent>
                                <xs:extension base="xs:string">
                                    <xs:attribute name="onsuccess"
type="xs:string"/>
                                </xs:extension>
                            </xs:simpleContent>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="loginhibitparameters" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="loopdata" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="looptype" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="narrative" type="xs:string"
minOccurs="0"/>
                    <xs:element name="noalwaysinit" type="xs:string"
minOccurs="0" maxOccurs="1"/>
```

```
                    <xs:element name="onfalse" type="guidType" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="onsuccess" type="guidType" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="ontrue" type="guidType" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="outputs" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="output" minOccurs="0"
maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:attribute name="friendlyname"
type="xs:string" use="optional"/>
                                        <xs:attribute name="narrative"
type="xs:string" use="optional"/>
                                        <xs:attribute name="name"
type="xs:string" use="optional"/>
                                        <xs:attribute name="stage"
type="xs:string" use="optional"/>
                                        <xs:attribute name="type"
type="xs:string" use="optional"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="parentobject" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="postconditions" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="preconditions" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="private" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="processid" type="guidType" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="references" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="resource" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="skill" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="step" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="steps" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element ref="subsheet" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                    <xs:element name="subsheetid" type="guidType"
minOccurs="0" maxOccurs="1"/>
                    <xs:element name="timeout" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="view" minOccurs="0"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="camerax" type="xs:decimal"
minOccurs="0"/>
                                <xs:element name="cameray" type="xs:decimal"
minOccurs="0"/>
                                <xs:element name="zoom" minOccurs="0">
                                    <xs:complexType>
                                        <xs:simpleContent>
                                            <xs:extension base="xs:decimal">
                                                <xs:attribute name="version"
type="xs:string" use="optional"/>
                                            </xs:extension>
                                        </xs:simpleContent>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:choice>
            </xs:sequence>
            <xs:attribute name="loginhibit" type="xs:string" use="optional"/>
            <xs:attribute name="name" type="xs:string" use="optional"/>
            <xs:attribute name="onfalse" type="guidType" use="optional"/>
            <xs:attribute name="onsuccess" type="guidType" use="optional"/>
            <xs:attribute name="ontrue" type="guidType" use="optional"/>
            <xs:attribute name="stageid" type="guidType" use="optional"/>
            <xs:attribute name="subsheetid" type="guidType" use="optional"/>
            <xs:attribute name="type" type="type" use="required"/>
            <!-- Allow any other attributes -->
            <xs:anyAttribute namespace="##any" processContents="skip"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```