

# Designing Smart Graphics “simple scenarios” with IMS Simple Sequencing

DRAFT 1.0

i3M has written this report in order to participate in the work of the AICC's Smart Graphics subcommittee. The AICC is allowed to use it in part or whole and to review it or modify it, provided that the author's and contributors' names are mentioned. Any use of this report outside the scope of the AICC is strictly forbidden without prior written permission of i3M.

The writing of this report has been made possible thanks to the participation of Airbus Training, which has provided a case study to validate the principles laid down in this document.

Author: Sébastien FRAYSSE – i3M

Contributors: Jean DURAND – i3M  
Karine HERVE – Airbus Training  
Jean-Louis BRAVO – Airbus Training

## CONTENTS

OBJECTIVE.....	3
BENEFITS .....	4
PRINCIPLES .....	5
GLOSSARY .....	6
ANALYSIS PHASE.....	8
Activity Tree .....	8
Resources.....	10
Sequencing.....	11
XML FORMALIZATION.....	15
Activity Tree .....	15
Declaration of resources.....	17
Management of sequencing .....	18
SMART GRAPHICS DATA MODEL.....	22
SMG Attribute Value .....	22
SMG Attribute Animation .....	22
SMG Highlight.....	23
SMGs Interaction .....	24
SMGs Configuration.....	24

## OBJECTIVE

The objective of this specification is to provide a standardized approach to describing and formalizing Smart Graphics **simple scenarios**. Simple scenarios are training scenarios that are intended for the learning, understanding and acquisition of aircraft procedures. Smart Graphics are used to display aircraft components and to allow the user to interact with them.

Several training strategies can be developed from simple scenarios. Training scenarios may thus be as follows:

- **Demonstration:** all steps of a linear procedure can be performed by the system or the instructor with comments or instruction given for each step.
- **Path simulation:** all steps of a linear procedure can be performed by the trainee who interacts with the Smart Graphics. The trainee can get feedback on its performance through the system that must always keep him on a main path.
- **Allowed deviations:** all steps of a procedure can be performed by the trainee who interacts with the Smart Graphics around a main linear path. A limited number of clearly defined alternative paths are allowed around the main one.

Simple scenarios exclude complex training strategies based on simulation. For instance, the following training strategies are beyond the scope of simple scenarios:

- "Goal oriented"
- "Intelligent tutoring"
- "Free play"

Please refer to the documents edited by the "Smart Graphics and simulation" subcommittee for further information about training strategies for simulation.

## **BENEFITS**

A standardized approach to describing and formalizing Smart Graphics **simple scenarios** should:

- Facilitate the design phase of procedure-based training by providing instructional designers with a standard framework.
- Allow instructional designers to exchange procedures and scenarios based on a common formalism.
- Facilitate the testing and validating process by using compliant tools.
- Facilitate communication and avoid misunderstanding between instructional designers and content developers.
- Facilitate the development of authoring tools for procedure-based training.

## PRINCIPLES

This specification is based on the following principles:

- The role of the instructional designer consists in analyzing a procedure from the aircraft manual and in extracting all the necessary background information by following some precise recommendations. The "Analysis Phase" chapter of this document examines these recommendations.
- Procedure information has then to be written in XML by using the IMS Content Packaging and IMS Simple Sequencing specifications, which are part of SCORM 2004. The way to convert procedure information into XML by using these specifications is examined in the "XML Formalization" chapter of this document.
- Additional information, associated with Smart Graphics, has also to be formalized and written in XML. It mainly concerns Smart Graphics configurations and interactions. The "Smart Graphics Data Model" chapter provides some useful information in this respect.

## GLOSSARY

**Procedure:** within the scope of this document, this term will refer to the overall aircraft procedure as it appears in the aircraft manual. It will also refer to the overall aircraft procedure that has been formalized in XML, including the hierarchy of tasks, the associated content and the conditions determining when steps of the procedure (elements of the activity tree) should be performed.

**Activity tree:** within the scope of this document, this term will refer to a conceptual structure that represents a hierarchy of tasks and sub-tasks that cover the entire procedure.

**Node:** within the scope of this document, this term will refer to each element of the activity tree.

**Leaf:** within the scope of this document, this term will refer to a node of the activity tree that has no child in the activity tree.

**Resource:** within the scope of this document, this term will refer to any kind of information that can be represented by a single file and be associated with a step of the procedure (a leaf of the activity tree). Smart Graphics, SMGs configurations, informative and instructional content are resources. Resources from a same step of the procedure (a same leaf of the activity tree) are grouped together to form a resource block.

**Resource block:** within the scope of this document, this term will refer to a set of files that is associated with a step of the procedure (to a leaf of the activity tree).

**Smart Graphic (SMG):** within the scope of this document, this term will refer to a configurable and interactive graphic that represents a specific part of an aircraft or a system (cockpit panels, ECAM pages, schematics, etc.).

**SMG attribute:** within the scope of this document, this term will refer to a configurable element of a Smart Graphic. An SMG attribute has either predefined values or a range of possible ones set between a minimum and a maximum.

**SMGs configuration:** within the scope of this document, this term will refer to the overall configuration of a set of Smart Graphics applied for a step of a procedure (for a leaf of the activity tree). SMGs configuration can include SMG attribute values, SMG attribute animations, SMG highlights and SMGs interaction.

**SMG attribute value:** within the scope of this document, this term will refer to the assignation of a value to an SMG attribute.

**SMG attribute animation:** within the scope of this document, this term will refer to the assignation of an animation to an SMG attribute. An animation is a sequence of successive values with continuous or discrete changes.

**SMG highlight:** within the scope of this document, this term will refer to a highlight effect on a Smart Graphic to help the trainee locate components or areas of the Smart Graphic. Highlights can be associated with SMG attributes or not.

**SMGs interaction:** within the scope of this document, this term will refer to the description of interactions between the user and a set of Smart Graphics. This description should include the actions that the user must perform and also those that he must not. It should also include reactions of the system like feedback and navigation consequences.

**Informative content:** within the scope of this document, this term will refer to the content that is associated with a step of the procedure (with a leaf of the activity tree) and that has to be displayed even during a testing or evaluation context. The nature of the content (text, image, sound, rich content...) is outside the scope of this document.

**Instructional content:** within the scope of this document, this term will refer to the content that is associated with a step of the procedure (with a leaf of the activity tree) and that has to be displayed only during a learning context (through explanations, etc.). The nature of the content (text, image, sound, rich content...) is outside the scope of this document.

**Scenario:** within the scope of this document, this term will refer to a specific context that determines how the procedure should apply. A specific scenario is determined by a set of variables that describe the context.

**Variable:** within the scope of this document, this term will refer to a single piece of information which is one of the parameters that determines a specific scenario. Each variable of a procedure should be clearly written in an understandable sentence.

**Condition:** within the scope of this document, this term will refer to an expression that can be evaluated to decide whether specific parts of a procedure (specific elements of an activity tree) will be available for the user. Conditions are based upon comparisons and logical expressions applied to variables.

**Path:** within the scope of this document, this term will refer to a selection of parts of the procedure (of nodes of the activity tree). This selection will refer to either the parts of the procedure that the user should perform or the ones that the user has already covered. Paths are determined by the scenario and by the user's decisions.

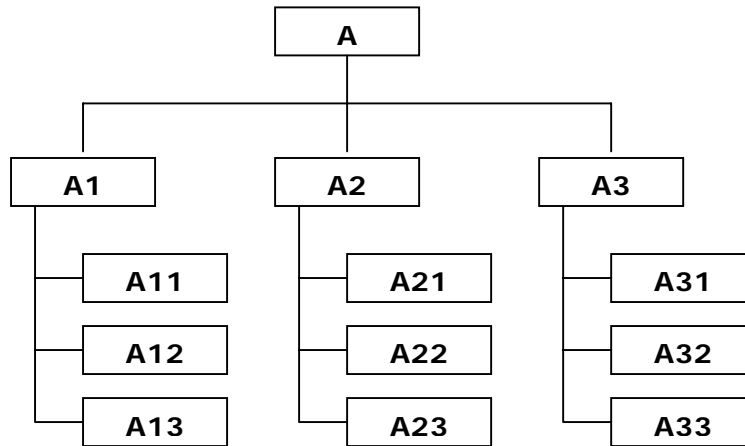
**Sequencing:** within the scope of this document, this term will refer to the overall strategy that is applied to determine the paths that are available for users. It includes the definition of the variables and conditions associated with the activity tree.

# ANALYSIS PHASE

Designing Smart Graphics scenarios with IMS Simple Sequencing should be based upon a preliminary analysis phase and should result in the identification and formalization of all the necessary information. This chapter describes some rules to follow during the analysis phase.

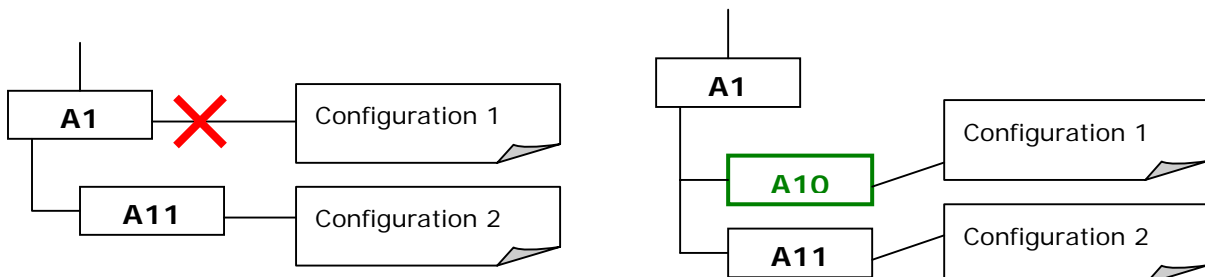
## Activity Tree

1. An **activity tree** should be associated with each procedure to represent its **hierarchy** of tasks and sub-tasks.
2. A unique **identifier** should be associated with each activity tree (with each procedure).
3. A **title** should be associated with each activity tree (with each procedure).
4. A unique **identifier** should be associated with each **node** of an activity tree.
5. A **title** should be associated with each **node** of an activity tree.

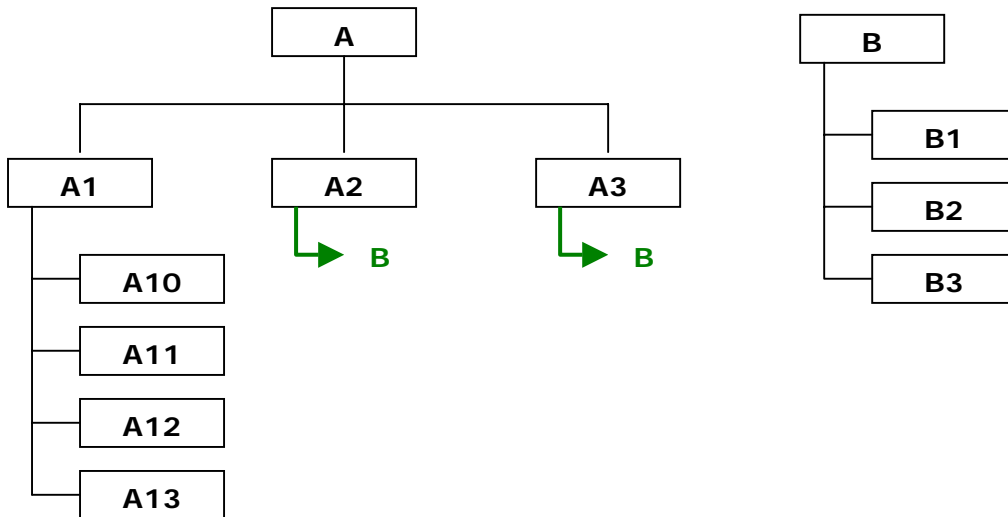


Id	Title
Proc_Example_1	Example of procedure N°1
A	Root node
A1	Preparation phase
A11	Open the door
...	...

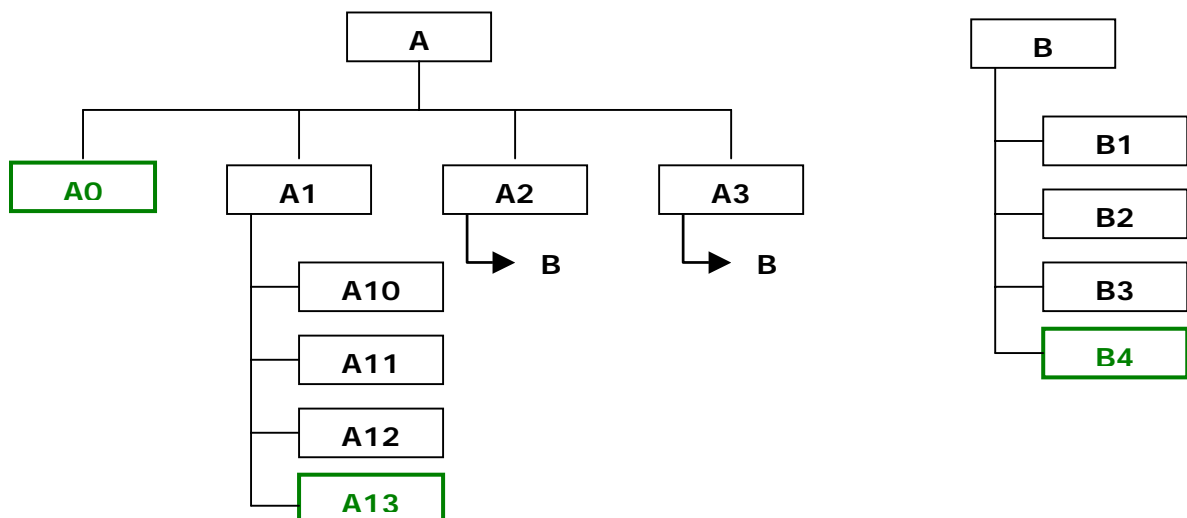
6. A set of **resources** should be associated with each **leaf** of an activity tree (refer to the "resources" section).
7. **Resources** cannot be associated with nodes that are not **leaves**. If necessary, additional leaves may be inserted so as to associate resources.



8. If necessary, **sub-procedures** may be created to avoid repetitions in a main procedure. An **activity tree** should be created for each sub-procedure.
9. Sub-procedures may be **connected** to a main procedure by specifying the **identifiers** on the matching leaves of the main activity tree. No resource can be associated with the connection leaves.

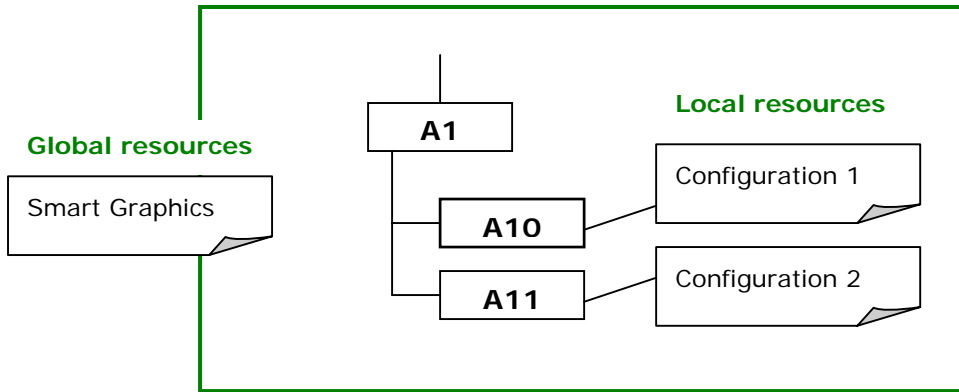


10. Each procedure should be properly initialized with an **initial leaf** in the activity tree. If necessary, an initialization leaf may be added at the beginning of the activity tree.
11. Each branch that precedes a major change in the procedure should appropriately end with a specific **final leaf** in the activity tree. If necessary, final leaves may be added to the activity tree.

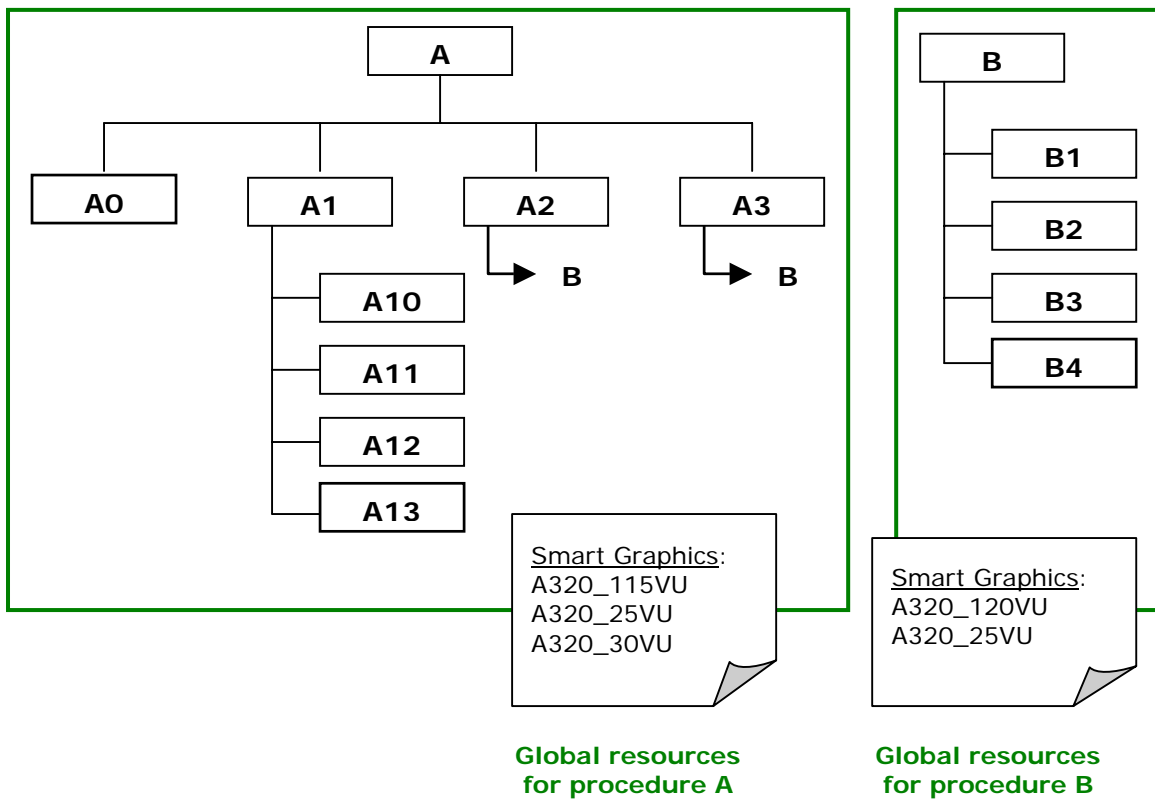


## Resources

1. A set of **global resources** can be declared for each procedure (or sub-procedure). An **identifier** should be associated with the set of resources.
2. A set of **local resources** (or leaf resources) should be declared for each leaf of an activity tree. An **identifier** should be associated with the set of resources.

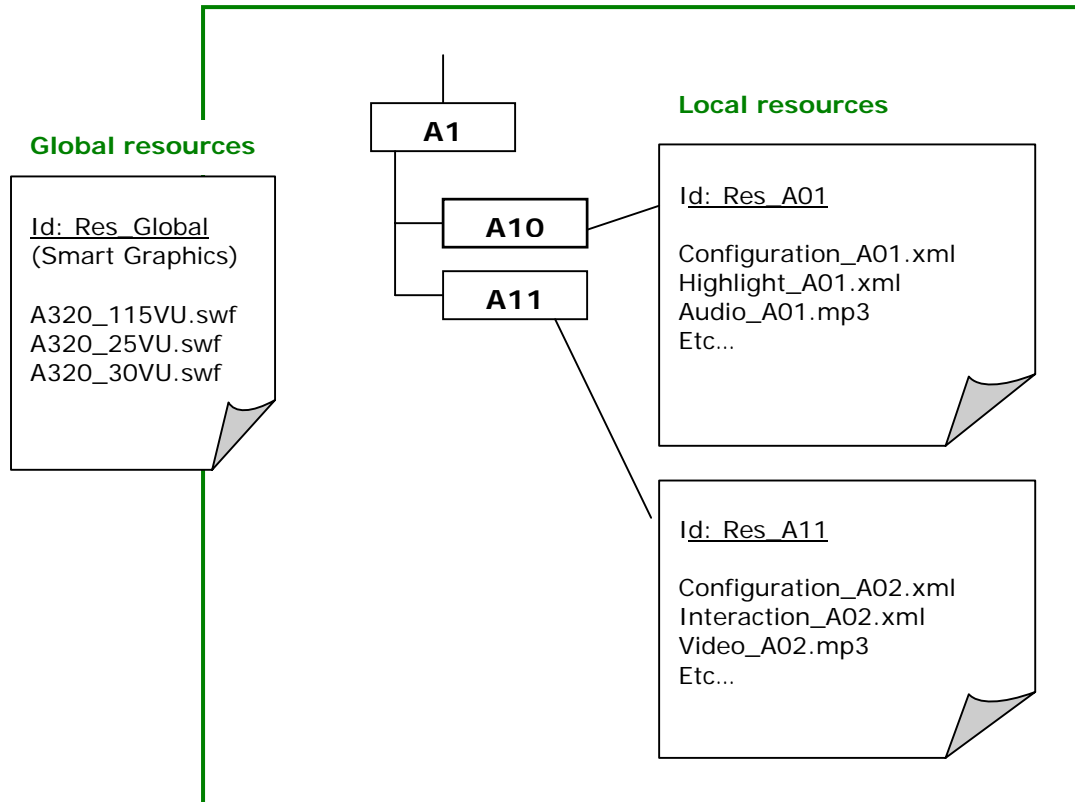


3. If necessary, sub-procedures may be created to associate a set of **resources** with **branches** of an activity tree. In this case, the set of resources becomes the global resources of the sub-procedure.



4. **Smart Graphics** can be declared as global or local resources of an activity tree. In all cases, the **identifier** of each Smart Graphic should be provided.

5. **SMGs configurations** can be associated with a leaf of an activity tree as a local resource. Please refer to the “SMGs data model” section for further information.
6. **Informative content** can be associated with a leaf of an activity tree as local resources. Specification of informative content is outside the scope of this document.
7. **Instructional content** can be associated with a leaf of an activity tree as local resources. Specification of informative content is outside the scope of this document.



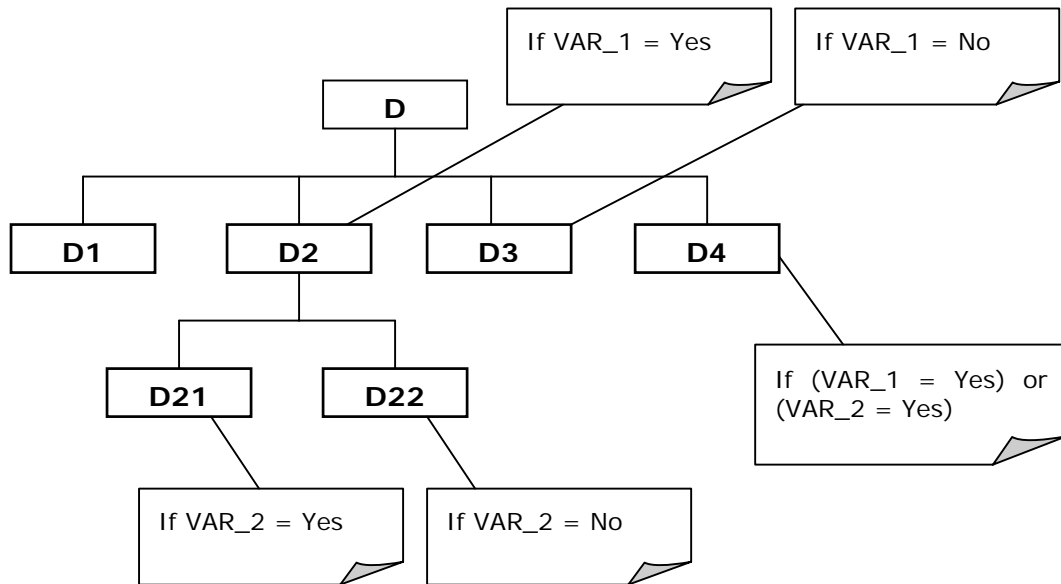
## Sequencing

1. Sequencing in the activity tree can be managed with variables and conditions.
2. A set of **variables** can be declared for each main activity tree (each main procedure).
3. A unique **identifier** should be associated with each variable.
4. A unique **title** should be associated with each variable.
5. A list of **possible values** should be associated with each variable.
6. A **number** and a **title** should be associated with each possible value.

Set of variables:

Identifier	Title	Possible values
VAR_1	Perceptible Smoke	1. Yes 2. No
VAR_2	Intense Smoke	1. Yes 2. No

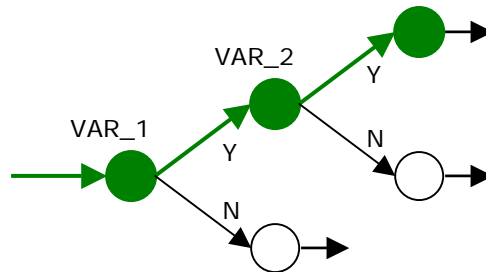
7. A **condition** can be associated with a node of an activity tree. In this case, a matching node is available for the user only if the associated condition evaluates as true.



8. For linear learning experiences, **values** of variables are determined before playing the scenario. For instance, they can be pre-determined by an instructor and recorded as a file that is read by the training application when the scenario is launched.

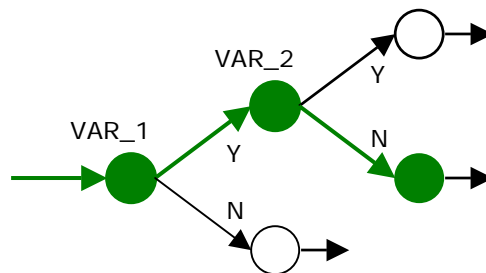
**Configuration1.txt**

VAR\_1 = Yes  
VAR\_2 = Yes

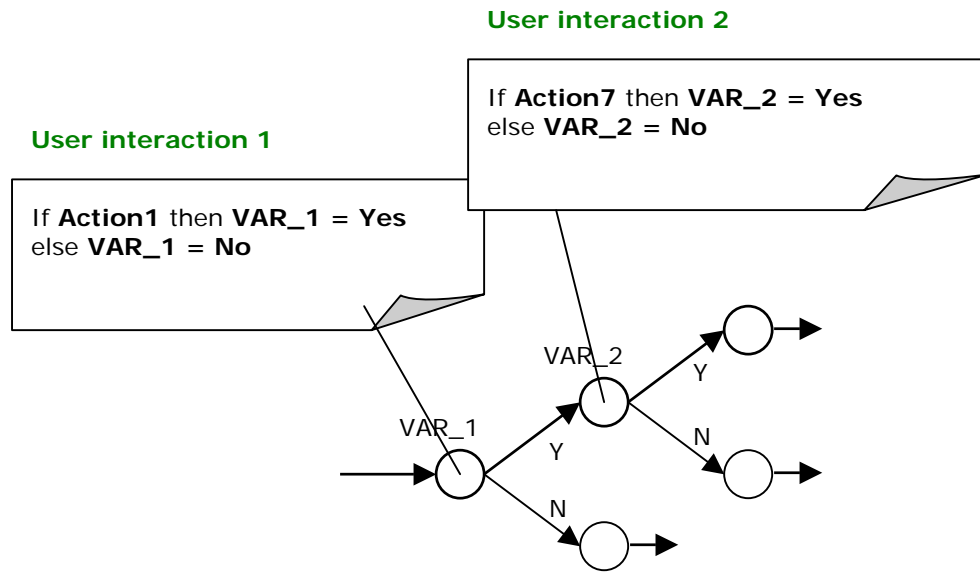


**Configuration2.txt**

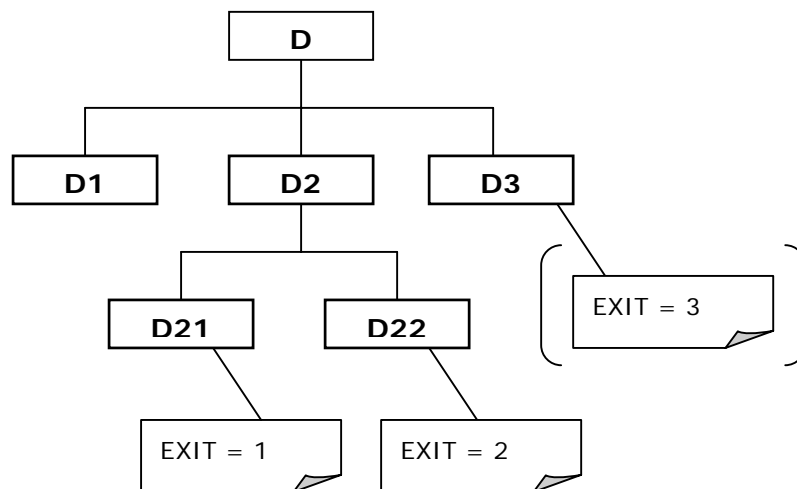
VAR\_1 = Yes  
VAR\_2 = No



- For non-linear learning experiences, **values** of variables are determined when the user interacts with the learning content. Values of variables are consequences of user actions.

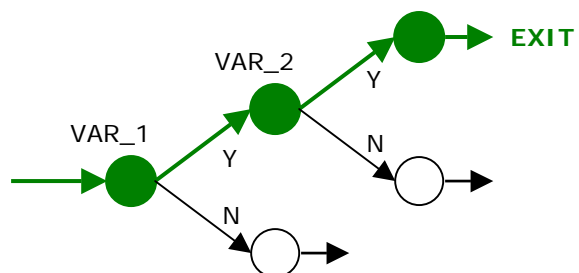


- An **exit condition** can be associated with a node of an activity tree. In this case, the procedure stops when the user has visited the matching node, if the condition is satisfied. An exit condition is implicit on the last leaf of the activity tree after all conditions are applied.

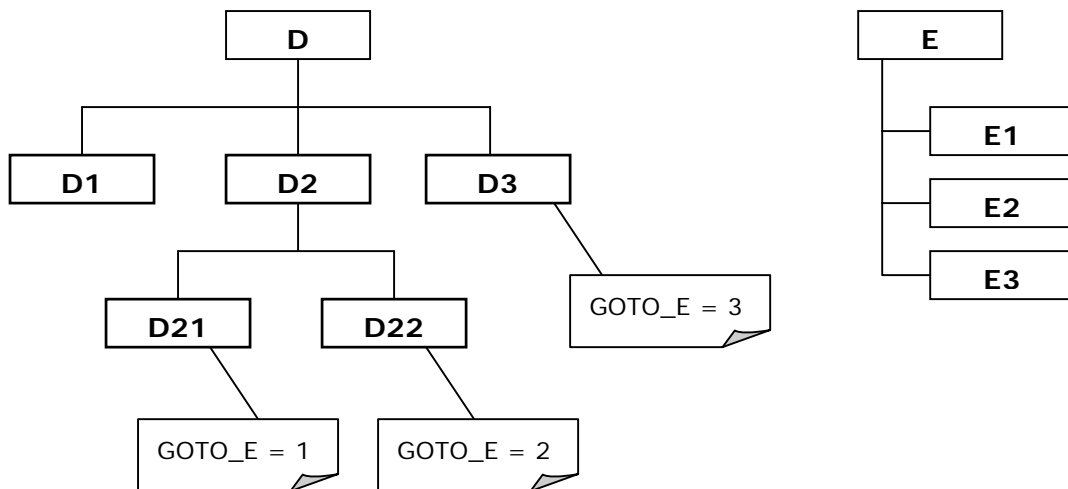


**Configuration1.txt**

VAR\_1 = Yes  
VAR\_2 = Yes  
EXIT = 1

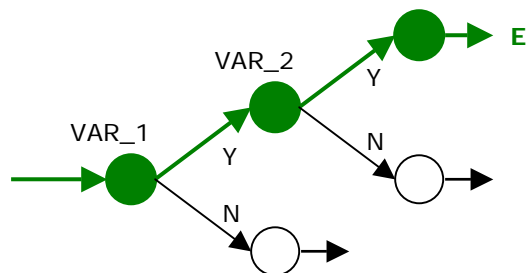


11. A **branching condition** can be associated with a leaf of an activity tree. In this case, an associated sub-procedure starts when the user accesses the matching leaf, if the condition is satisfied.



**Configuration1.txt**

```
VAR_1 = Yes
VAR_2 = Yes
GOTO_E = 1
```



## XML FORMALIZATION

The XML formalization consists in writing an XML file called **imsmanifest.xml**, which is used to describe a full aircraft procedure. This file must comply with the IMS Simple Sequencing and IMS Content Packaging specifications. By extension, this file should be also SCORM 2004 compliant.

This chapter explains how to write the "imsmanifest.xml" file in the context of an aircraft procedure. It does not claim to cover all aspects of IMS Content Packaging and IMS Simple Sequencing. Only some useful elements of these specifications have been retained as part of Smart Graphics simple scenarios.

Please refer to the IMS Content Packaging, IMS Simple Sequencing and ADL SCORM 2004 specifications for further information.

### Activity Tree

#### <manifest>

This is the top-level element of the "imsmanifest.xml" file. It should include several attributes. One can start from the following example:

```
<manifest identifier="SMG_Scenario" version="1.3"
  xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns="http://www.imsglobal.org/xsd/imsss"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlseq_v1p3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
    imscp_v1p1.xsd
    http://www.imsglobal.org/xsd/imsss
    imsss_v1p0.xsd
    http://www.adlnet.org/xsd/adlcp_v1p3
    adlcp_v1p3.xsd"
    http://www.adlnet.org/xsd/adlseq_v1p3
    adlseq_v1p3.xsd">
  (...)
</manifest>
```

#### <metadata>

The use of this element is outside the scope of this document. However, this element is required to be SCORM 2004 compliant. One can use the following example:

```
<manifest identifier="SMG_Scenario" version="1.3"
  (...)
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
  </metadata>
</manifest>
```

## <organizations>

These elements provide the description of the activity tree. In fact, IMS Content Packaging (IMS CP) enables several activity trees to be declared in the same "imsmanifest.xml" file. In the following example, we only declare one **activity tree** by specifying its **identifier** and **title**.

```
<manifest identifier="SMG_Scenario" version="1.3"
(...)
  <organizations default="Proc_Example_1">
    <organization identifier="Proc_Example_1">
      <title>Example of procedure 1</title>
    </organization>
  </organizations>
</manifest>
```

## <item>

Items are used to declare the hierarchy of nodes of the activity tree. An **identifier** and a **title** are specified for each item as well as a reference to a set of resources.

```
<organization identifier="Proc_Example_1">
  <title>Example of procedure 1</title>

  <item identifier="B">
    <title>Root node</title>
    <item identifier="B1" identifierref="Res-B1">
      <title>Step 1</title>
    </item>
    <item identifier="B2" identifierref="Res-B2">
      <title>Step 2</title>
    </item>
    <item identifier="B3" identifierref="Res-B3">
      <title>Step 3</title>
    </item>
    <item identifier="B4" identifierref="Res-B4">
      <title>Step 4</title>
    </item>
  </item>
</organization>
```

## (sub) <manifest>

The manifest> element enables the declaration of **sub-manifests**, which are useful to create **sub-procedures** that can be launched from several locations of a main procedure.

```
<manifest identifier="SMG_Scenario" version="1.3"
(...)
  <organization identifier="Proc_Example_1">
    <title>Example of procedure 1</title>
    <item identifier="A">
      <title>Root node</title>
      <item identifier="A2" identifierref="SMG_Sub_Scenario_1"/>
    </item>
  </organization>
```

```

    < manifest identifier="SMG_Sub_Scenario_1" version="1.3">
      (...)
    </manifest>
</manifest>

```

## Runtime information

The LMS usually parses the "imsmanifest.xml" file in order to display a navigation menu that represents the activity tree. The LMS manages the navigation and launches the associated resources. The LMS also manages the sequencing process. The LMS should be SCORM 2004 compliant.

Several other alternative solutions can possibly be implemented. For instance, a content player can be developed to parse the "imsmanifest.xml" file and to manage the navigation itself.

## Declaration of resources

Several sets of resources can be declared in the "imsmanifest.xml" file with the <resources> element, as defined in the IMS Content Packaging specification. This chapter explains how to write the <resources> section. It only includes the information necessary for designing Smart Graphics scenarios. Please refer to the IMS Content Packaging specification for further information.

### <resources>

The <resources> element is declared at the same level as the <organizations> element in the "imsmanifest.xml" file. It contains several **sets of resources**. Each set is declared with a <resource> element and is associated with a unique **identifier**. A set of leaf resources can be associated with a leaf using the "identifierref" attribute of the <item> element. There is no difference between local and global resources in the "imsmanifest.xml" file, except that global resources are not associated with items.

```

<manifest identifier="SMG_Scenario" version="1.3"
  (...)
  <organization identifier="Proc_Example_1">
    <title>Example of procedure 1</title>
    <item identifier="B">
      <title>Root node</title>
      <item identifier="B1" identifierref="Res_B1">
        <title>Step 1</title>
      </item>
    </item>
  </organization>
  <resources>
    <resource identifier="Res_Global" type="webcontent" href="layout.htm">
      (...)
    </resource>
    <resource identifier="Res_B1" type="webcontent" href="send.htm">
      (...)
    </resource>
  </resources>
</manifest>

```

## <file>

A <file> element is used to declare each **physical resource** of a set of resources. These can be Smart Graphics, configuration files or anything else.

```
<resources>
  <resource identifier="Res_Global" type="webcontent" href="layout.htm">
    <file href="A320_115VU.swf" />
    <file href="A320_25VU.swf" />
    <file href="A320_30VU.swf" />
  </resource>
  <resource identifier="Res_B1" type="webcontent" href="send.htm">
    <file href="B1_config.xml" />
  </resource>
</resources>
```

## Runtime information

The way to use and display resources during runtime is not specified in the "imsmanifest.xml" file. The only indication is the "href" attribute of the <resource> element, which refers to a file that is automatically launched when the user visits the associated leaf (item). This file can be considered as the main learning content of the set of resources. Its function is to manage the set of resources. The implementation of this file is free and outside the scope of this document.

```
<resources>
  <resource identifier="Res_Global" type="webcontent" href="layout.htm">
    <file href="A320_115VU.swf" />
    <file href="A320_25VU.swf" />
    <file href="A320_30VU.swf" />
  </resource>
  <resource identifier="Res_B1" type="webcontent" href="send.htm?B1_conf.xml">
    <file href="B1_conf.xml" />
  </resource>
</resources>
```

In this example, the "layout.htm" file is used to arrange the Smart Graphics into an HTML page. The "send.htm" file is used to send the Smart Graphics configuration ("B1\_config.xml" file) to the Smart Graphics being displayed.

## Management of sequencing

Sequencing can be managed by using the IMS Simple Sequencing specification. This specification is an extension of IMS Content Packaging and consists in adding sequencing information to the "imsmanifest.xml" file. Furthermore, IMS Simple Sequencing has been integrated into the ADL SCORM 2004 specification. This chapter explains how to write sequencing information in the context of Smart Graphics scenarios. Please refer to the IMS Simple Sequencing specification for further information.

## Declaration of variables

Variables can be created and managed with IMS Simple Sequencing by using the concept of "Objectives". A set of objectives can be defined and used to store and retrieve information. Each time that a variable has to be read or written, it must be declared in the "imsmanifest.xml" file at the matching item. A variable is therefore created when it is declared for the first time. The identifier of the variable must be mentioned when it is declared.

```
<item identifier="B1" identifierref="Res-B1">
  <title>Step 1</title>

  <imsss:sequencing>
    <imsss:objectives>
      <imsss:primaryObjective/>

      <imsss:mapInfo targetObjectiveID="VAR_1"
        readSatisfiedStatus="true"
        readNormalizedMeasure="true"
        writeSatisfiedStatus="true"
        writeNormalizedMeasure="true" >

      <imsss:mapInfo targetObjectiveID="VAR_2"
        readSatisfiedStatus="true"
        readNormalizedMeasure="true"
        writeSatisfiedStatus="true"
        writeNormalizedMeasure="true" >
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

## Initialization and modification of variables

IMS and SCORM do not provide a standardized approach to assigning values to variables. This cannot be done directly in the "imsmanifest.xml" file. However, variables can be accessed from any learning content that is able to communicate with an LMS and that uses a standardized communication protocol. The SCORM RunTime Environment enables this communication thanks to an LMS API.

The following example is a learning content that has been developed in HTML and Javascript to initialize variables. The variable "api" refers to the LMS API. Please refer to the SCORM RunTime Environment specification and its examples for further information.

```
api.setValue("cmi.objectives.1.id", "VAR_1");
api.setValue("cmi.objectives.1.score.scaled", "1.0");

api.setValue("cmi.objectives.2.id", "VAR_2");
api.setValue("cmi.objectives.2.score.scaled", "0.5");
```

## Evaluation of conditions and associated actions

When one or several variables have been created and initialized, conditions can be used and attached to nodes of the activity tree. A condition is a logical expression based on comparisons (greater than, less than) and operators (or, and, not). When this expression evaluates as true, some actions can be launched (skip, hide or disable a branch of the activity tree, exit from the procedure, jump to a sub-procedure).

The following example shows how to **disable** a branch of the activity tree after the evaluation of a simple condition. The condition in this example is "VAR\_1 < 0.5" and is described by using the attributes **referencedObjective**, **condition** and **measureThreshold**.

```
<item identifier="D2" identifierref="Res-D2">
  <title>Phase 2</title>
  (...)

  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition
            measureThreshold="0.5"
            referencedObjective="VAR_1"
            condition="objectiveMeasureLessThan"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
    <imsss:objectives>
      (...)
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

The following example shows how to **hide** a branch of the activity tree after the evaluation of a more complex expression. The condition is "VAR\_1 < 0.5 and VAR\_2 > 0.0". The attribute **conditionCombination** is used with an "all" value to create the logical "and" expression.

```
<item identifier="D2" identifierref="Res-D2">
  <title>Phase 2</title>
  (...)

  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule conditionCombination="all">
        <imsss:ruleConditions>
          <imsss:ruleCondition
            measureThreshold="0.5"
            referencedObjective="VAR_1"
            condition="objectiveMeasureLessThan"/>
        </imsss:ruleConditions>
        <imsss:ruleConditions>
          <imsss:ruleCondition
            measureThreshold="0.0"
            referencedObjective="VAR_2"
            condition="objectiveMeasureGreaterThan"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "hiddenFromChoice"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
    <imsss:objectives>
      (...)
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

The following example shows how to jump from a main manifest to a **sub-manifest** after the evaluation of a simple **condition**. It is important to know that a condition cannot be associated directly with the branching item. A parent item should be added to attach the condition. This item will appear in the navigation menu but has no real meaning for the user.

```
<item identifier="A">
  <title>Root node</title>

  <item identifier="A1_Cond">
    <title>Sub-procedure</title>
    <item identifier="A1" identifierref="SMG_Sub_Scenario_1"/>
    <imsss:sequencing>
      <imsss:sequencingRules>
        (...)
      </imsss:sequencingRules>
      <imsss:objectives>
        (...)
      </imsss:objectives>
    </imsss:sequencing>
  </item>
</item>
```

The following example shows how to **exit** from a manifest after the evaluation of a simple condition. The exit condition is a **post condition**. This means that the exit action is being performed after the associated item has been completely visited, even if this item contains sub-items.

```
<item identifier="D2" identifierref="Res-D2">
  <title>Phase 2</title>
  (...)

  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:postConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition
            measureThreshold="0.5"
            referencedObjective="VAR_1"
            condition="objectiveMeasureLessThan"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "exitAll"/>
      </imsss:postConditionRule>
    </imsss:sequencingRules>
    <imsss:objectives>
      (...)
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

## SMART GRAPHICS DATA MODEL

This chapter does not claim to provide a complete and generic data model for Smart Graphics. It only describes what kind of information should be included in a Smart Graphics data model to enable the creation of scenarios. The objective is to provide merely an illustration. The Smart Graphics data model is based on the following principles:

1. The data model should make it possible to describe the configurations of a set of Smart Graphics, whether they are in a single file or in multiple files.
2. SMGs configurations should be able to include the description of attribute values, attribute animations, highlights and interactions.

### SMG Attribute Value

Information	Multiplicity
SMGDataModel	No
SMGDataModel.SmartGraphic	Yes
SMGDataModel.SmartGraphic.Id	No
SMGDataModel.SmartGraphic.Attribute	Yes
SMGDataModel.SmartGraphic.Attribute.Id	No
SMGDataModel.SmartGraphic.Attribute.Value	No

#### Example:

```
<SMGDataModel>
  <smartGraphic id="A320_115VU">
    <attribute id="MASTER_1">
      <value>ON</value>
    </attribute>
    <attribute id="MASTER_2">
      <value>ON</value>
    </attribute>
  </smartGraphic>
  <smartGraphic id="A320_ECAM_EWD">
    <attribute id="ENGINE_1">
      <value>20.0</value>
    </attribute>
  </smartGraphic>
</SMGDataModel>
```

This example initializes attributes on the ENGINE cockpit panel and on the ECAM page.

### SMG Attribute Animation

Smart Graphics animation is very similar to Smart Graphics configuration, except that the attribute value is not a single but a temporal one, that is a step-by-step animation sequence.

Information	Multiplicity
SMGDataModel	No
SMGDataModel.SmartGraphic	Yes
SMGDataModel.SmartGraphic.Id	No
SMGDataModel.SmartGraphic.Attribute	Yes
SMGDataModel.SmartGraphic.Attribute.Id	No
SMGDataModel.SmartGraphic.Attribute.Animation	No
SMGDataModel.SmartGraphic.Attribute.Animation.Step	Yes
SMGDataModel.SmartGraphic.Attribute.Animation.Step.Duration	No
SMGDataModel.SmartGraphic.Attribute.Animation.Step.TargetValue	No

**Example:**

```
<SMGDataModel>
  <smartGraphic id="A320_ECAM_EWD">
    <attribute id="ENGINE_1">
      <animation>
        <step id="step1" duration="2000">30.0</step>
        <step id="step2" duration="1000">50.0</step>
        <step id="step3" duration="2000">70.0</step>
      </animation>
    </attribute>
  </smartGraphic>
</SMGDataModel>
```

This example plays an animation on the ECAM page.

**SMG Highlight**

Smart Graphics highlight is very similar to Smart Graphics configuration, except that it does not specify attribute values but the highlight associated with attributes.

Information	Multiplicity
SMGDataModel	No
SMGDataModel.SmartGraphic	Yes
SMGDataModel.SmartGraphic.Id	No
SMGDataModel.SmartGraphic.Attribute	Yes
SMGDataModel.SmartGraphic.Attribute.Id	No
SMGDataModel.SmartGraphic.Attribute.Highlight	No
SMGDataModel.SmartGraphic.Attribute.Highlight.Visible	No

**Example:**

```
<SMGDataModel>
  <smartGraphic id="A320_115VU">
    <attribute id="MASTER_2">
      <highlight visible="true"/>
    </attribute>
  </smartGraphic>
</SMGDataModel>
```

This example highlights the MASTER\_2 button of the ENGINE cockpit panel.

## SMGs Interaction

Smart Graphics interaction can be a more or less complex process. In this document, we consider the simplest way to describe an SMG interaction, but it lets people free to use descriptions that are more complex.

The simplest way to describe an SMG interaction is to define the SMG configuration that should result from the user actions. The actions are considered as correct when the user has set the described configuration.

More complex interactions could be described by specifying correct and wrong actions, associated feedback, try-out numbers, etc...

Information	Multiplicity
SMGDataModel	No
SMGDataModel.SmartGraphic	Yes
SMGDataModel.SmartGraphic.Id	No
SMGDataModel.SmartGraphic.Attribute	Yes
SMGDataModel.SmartGraphic.Attribute.Id	No
SMGDataModel.SmartGraphic.Attribute.Interaction	No
SMGDataModel.SmartGraphic.Attribute.Interaction.CorrectValue	No

### Example:

```
<SMGDataModel>
  <smartGraphic id="A320_115VU">
    <attribute id="MASTER_2">
      <interaction>
        <correctValue>ON</correctValue>
      </interaction>
    </attribute>
  </smartGraphic>
</SMGDataModel>
```

This example describes an interaction. The trainee has to press the MASTER\_2 button on the ON position.

## SMGs Configuration

Of course, the data models that are described in this chapter can be used to write XML files containing Smart Graphics attribute values, animations, interactions and highlights and can be merged into a single file. The complete data model should be the following:

Information	Multiplicity
SMGDataModel	No
SMGDataModel.SmartGraphic	Yes
SMGDataModel.SmartGraphic.Id	No
SMGDataModel.SmartGraphic.Attribute	Yes
SMGDataModel.SmartGraphic.Attribute.Id	No
SMGDataModel.SmartGraphic.Attribute.Value	No

SMGDataModel.SmartGraphic.Attribute.Animation	No
SMGDataModel.SmartGraphic.Attribute.Animation.Step	Yes
SMGDataModel.SmartGraphic.Attribute.Animation.Step.Duration	No
SMGDataModel.SmartGraphic.Attribute.Animation.Step.TargetValue	No
SMGDataModel.SmartGraphic.Attribute.Interaction	No
SMGDataModel.SmartGraphic.Attribute.Interaction.CorrectValue	No
SMGDataModel.SmartGraphic.Attribute.Highlight	No
SMGDataModel.SmartGraphic.Attribute.Highlight.Visible	No

**Example:**

```

<SMGDataModel>
  <smartGraphic id="A320_115VU">
    <attribute id="MASTER_1">
      <value>ON</value>
    </attribute>
    <attribute id="MASTER_2">
      <value>OFF</value>
      <highlight visible="true"/>
      <interaction>
        <correctValue>ON</correctValue>
      </interaction>
    </attribute>
  </smartGraphic>
  <smartGraphic id="A320_ECAM_EWD">
    <attribute id="ENGINE_1">
      <animation>
        <step id="step1" duration="2000">30.0</step>
        <step id="step2" duration="1000">50.0</step>
        <step id="step3" duration="2000">70.0</step>
      </animation>
    </attribute>
  </smartGraphic>
</SMGDataModel>

```

In this example, the Smart Graphics configuration describes an animation on the ECAM page, initializes the ENGINE cockpit panel, highlights the MASTER\_2 button and waits for an interaction on the MASTER\_2 button. The trainee has to press the MASTER\_2 button on the ON position.