International Standard

ISO 16400-3

First edition
2024-03

Automation systems and integration — Equipment behaviour catalogues for virtual production system —

Part 3: Requirements and recommendations for construction of an equipment instance model

Reference number
ISO 16400-3:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 5, *Interoperability, integration, and architectures for enterprise systems and automation applications*.

A list of all parts in the ISO 16400 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The ISO 16400 series introduces a concept of an equipment behaviour catalogue (EBC), addresses the requirements of an EBC and provides requirements and recommendations to generate an executable model representing the dynamic behaviour of a nominal or a physical instance of an equipment. An equipment instance model is implemented, such as a software agent. Such an executable model plays a vital role when configuring virtual production systems used for simulation and verification of a future process as well as monitoring of a current process. Therefore, EBCs will constitute an important part in the development of smart manufacturing.

An EBC enables an efficient and standardized way for a provider of equipment to communicate its dynamic behaviour.

The ISO 16400 series consists of the following parts, under the general title *"Automation systems and integration — Equipment behaviour catalogues for virtual production system"*:

Part 1: *Overview*

Part 2: *Formal description of a catalogue template*

Part 3 (this document): *Requirements and recommendations for construction of an equipment instance model*

Automation systems and integration — Equipment behaviour catalogues for virtual production system —

Part 3: Requirements and recommendations for construction of an equipment instance model

1 Scope

This document provides requirements and recommendations on how to construct an equipment instance model using an equipment behaviour catalogue (EBC) item.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 16400-1:2020, *Automation systems and integration — Equipment behaviour catalogues for virtual production system — Part 1: Overview*

ISO 16400-2:2024, *Automation systems and integration — Equipment behaviour catalogues for virtual production systems — Part 2: Formal description of a catalogue template*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 16400-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
construction processor
software tool to construct an equipment instance model from an equipment behaviour catalogue (EBC) item refering to construction requirements

3.2
equipment instance model
executable model corresponding to a specification of an equipment behaviour catalogue (EBC) item

4 Abbreviated terms

XML eXtensible Markup Language

XSL eXtensible Stylesheet Language

XSLT eXtensible Stylesheet Language Transformations

5 Required functionalities for an equipment instance model

When the virtual production system of interest is constructed, an equipment instance model shall be constructed from an appropriate EBC item.

An equipment instance model shall have the following functionalities:

— processable description of behaviour flow specified in the EBC item;

NOTE Behaviour flow means a sequence of behaviours performed during the execution of an equipment instance model.

— processable description of formula(s) or mathematical model(s);

— processable description of external interactions.

To fulfil these functionalities, an equipment instance model can be modelled in various ways, e.g. agent-based modelling, discrete event modelling, dynamic system modelling. In this document, an equipment instance model is implemented as a software agent.

6 Role of construction requirements

6.1 Relationship between an EBC item and an equipment instance model

A production system model is constructed using equipment instance models. An equipment instance model is constructed from the selected EBC item. The EBC items and the EBC templates are registered in a shared repository (see ISO 16400-1:2020, Figure 4).

Figure 1 shows three cases for construction of an equipment instance model. All cases shall be treated by the construction requirements.

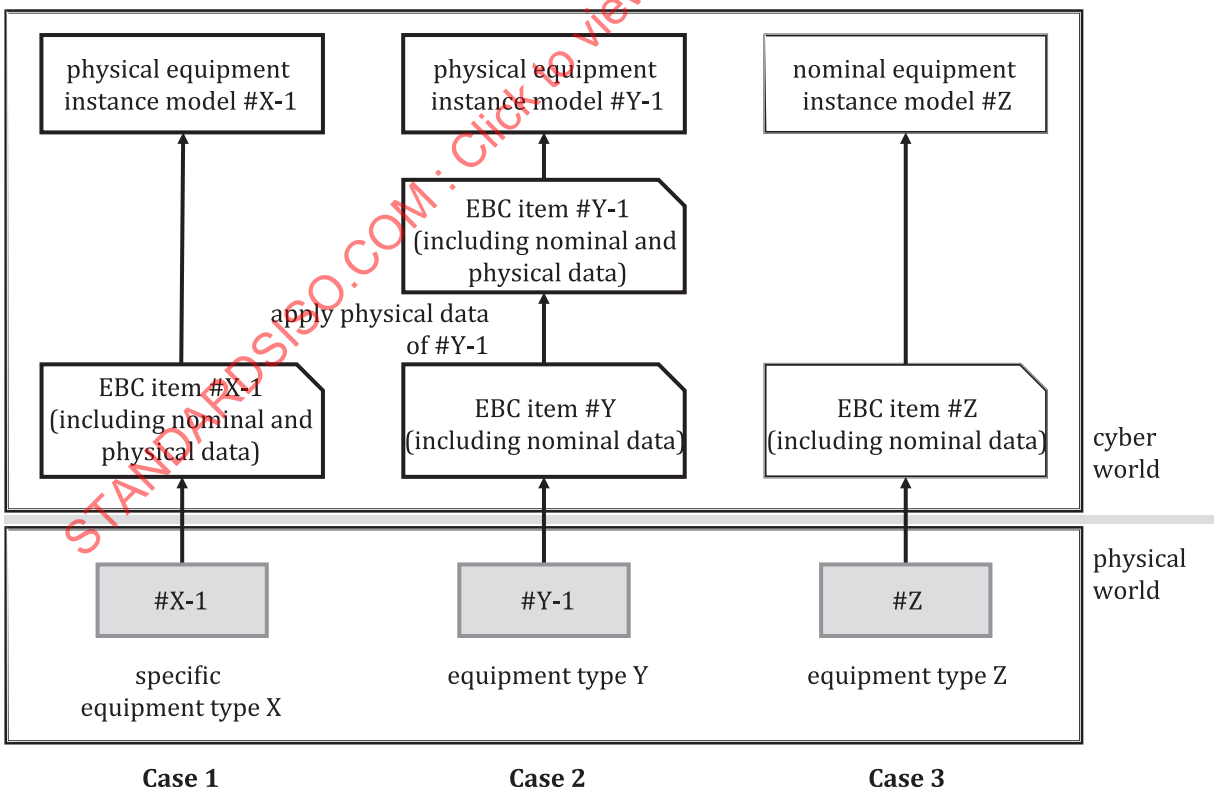


Figure 1 — Three cases for the construction of equipment instance models

— Case 1: If the EBC item contains the physical data of a specific equipment that actually exists, a physical equipment instance model is constructed;

— Case 2: If the EBC item contains only nominal data as designed for a specific equipment, but the specific physical data exists, applying the physical data to the corresponding EBC item, a physical equipment instance model is constructed;

— Case 3: If the EBC item contains only nominal data as designed for a specific equipment, a nominal equipment instance model is constructed.

6.2 Construction processor

An equipment instance model shall be constructed using the construction requirements. These construction requirements shall be applicable to the above three cases.

The construction processor of the target environment where the equipment instance model will be executed takes an EBC item as an input and constructs an equipment instance model.

Depending on whether the EBC item contains nominal data or physical data, there can be three cases in which the equipment instance model is constructed by a construction processor using the construction requirements as shown in Figure 2.

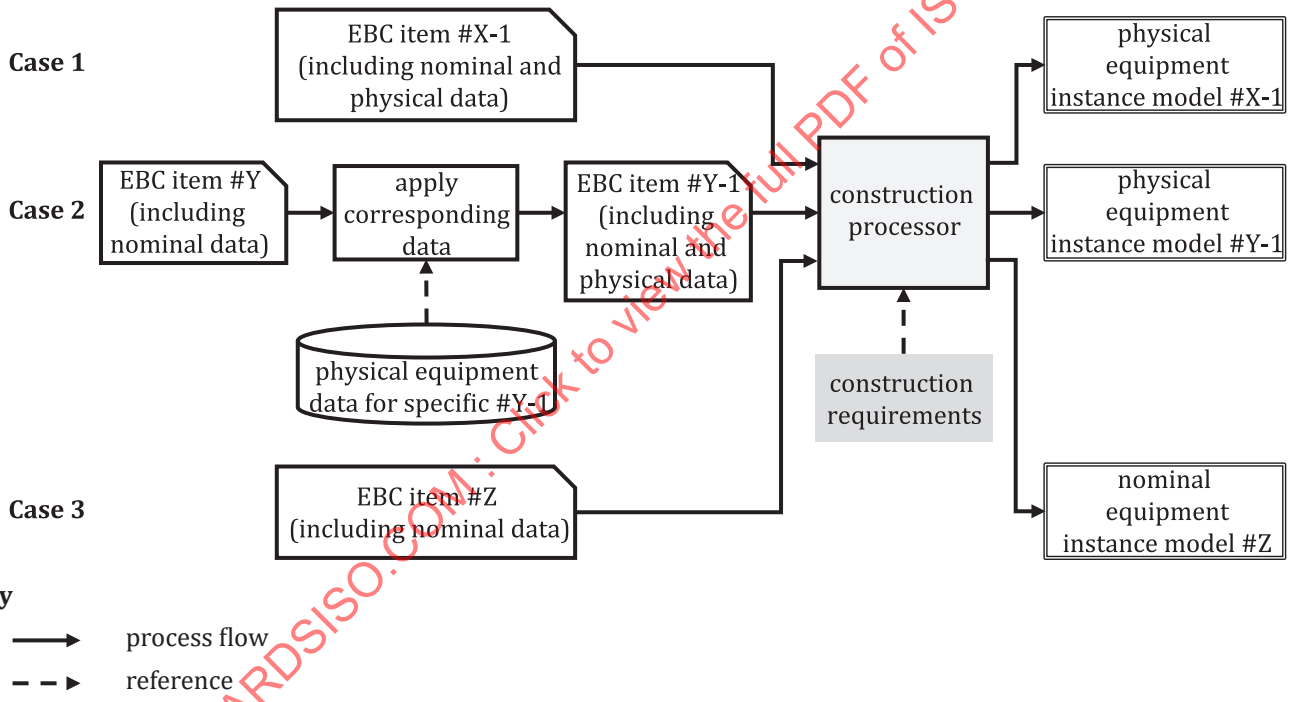
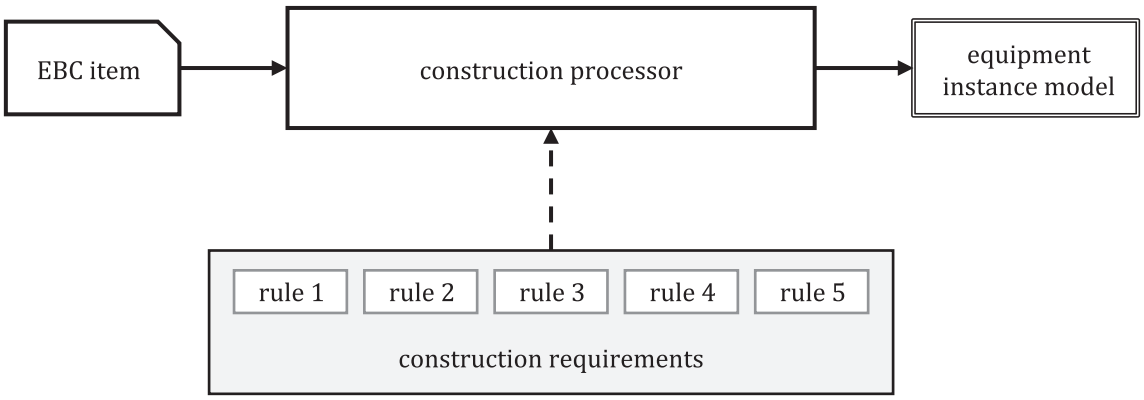


Figure 2 — Construction of an equipment instance model

7 Construction requirements

7.1 General

The construction requirements are represented by the construction rules. Details of construction rules are described in 7.2. The construction procedures shall be implemented in the construction processor by following the construction requirements. The construction procedures are executed by the construction processor which accepts an EBC item to construct an equipment instance model. Figure 3 shows a construction processor based on the construction requirements.



Key

process flow

reference

Figure 3 — Construction processor

7.2 Construction rules

An EBC item is an instance of an EBC template which includes descriptions for a header, a property set, behaviours and external interactions. The structure of an EBC item uses the correspondent EBC template. An equipment instance model is constructed from an EBC item and related data of an equipment design (nominal data) or related data of an equipment individual (physical data).

Rules to construct an equipment instance model shall consist of the following:

— Rule 1: describes general information in an equipment instance model using a header of an EBC item;

— Rule 2: describes properties to be processable in an equipment instance model using a property set of an EBC item;

— Rule 3: describes a state of an equipment instance model using each state of the behaviour of an EBC item;

— Rule 4: describes a state transition of an equipment instance model using each state transition of the behaviour of an EBC item;

— Rule 5: describes a procedure in either a state or a state transition, or both, of an equipment instance model using each external interaction of an EBC item.

The required functionalities for an equipment instance model described in Clause 5 shall be implemented according to the construction rules as follows:

— the description of general information for an equipment instance model is made by Rule 1;

— the processable description of behaviour flow is made by Rule 2, Rule 3 and Rule 4;

— the processable description of formula(s) or mathematical model(s) is made by Rule 2 and Rule 3;

— the processable description of external interactions is made by Rule 5.

Annex A shows three examples of EBC items. Annex B shows an implementation example of a construction processor for EBC items in Annex A.

Annex A
(informative)

Example of EBC items for an injection molding line

A.1 Equipment on an injection molding line

The structure of an injection molding line is shown in Figure A.1. This injection molding line consists of an injection molding machine including mold, a molding extraction robot, and a mold temperature controller. The examples of building the EBC templates for the injection molding line are presented in ISO 16400-2: 2024, Annex C. The example of the EBC items using these EBC templates is described in A.2 to A.4. These EBC items are used to construct equipment instance models by the construction processor in Annex B.

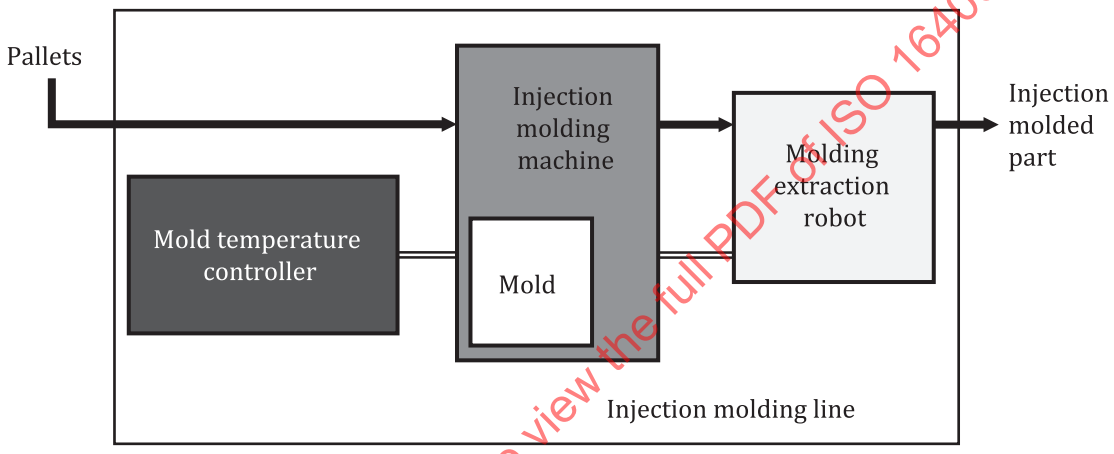


Figure A.1 — Structure of an injection molding line

A.2 EBC item for an injection molding machine

In ISO 16400-2:2024, C.2, the example of the EBC template of an injection molding machine is represented. The example of the EBC item using this EBC template can be described using XML as follows:

```
<ebc xmlns:op="product-operation.xml" xmlns:m="https://www.w3.org/1998/Math/MathML">
   <header>
    <template name="INJECTION_MOLDING_MACHINE" id="template_id_imm1"/>
   </header>

   <property_set>
    <!-- default value setting -->
    <!-- default value setting -->
    <var name="equipment_id" type="String" value="imm1"/>
    <var name="equipment_type" type="String" value="XXX"/>
    <var name="organization" type="String" value="XYZ"/>
    <var name="model_name" type="String" value="im_machine"/>
    <var name="external_size_hight" type="Double" value="100" unit="cm"/>
    <var name="external_size_width" type="Double" value="250" unit="cm"/>
    <var name="external_size_depth" type="Double" value="100" unit="cm"/>
    <var name="equipment_weight" type="Double" value="200" unit="kg"/>
    <var name="startup_time" type="Double" value="20" unit="sec"/>
    <var name="shutdown_time" type="Double" value="20" unit="sec"/>
    <var name="electricpower_idling" type="Double" value="30" unit="kW"/>
    <var name="electricpower_mold_change" type="Double" value="30" unit="kW"/>
    <var name="electricpower_mold_cleanig" type="Double" value="30" unit="kW"/>
    <var name="electricpower_cylinder_cleaning" type="Double" value="30" unit="kW"/>
```

```
 <var name="electricpower_injection_molding" type="Double" value="30" unit="kW"/>
 <var name="electricpower_opening_of_a_mold" type="Double" value="30" unit="kW"/>
 <var name="electricpower_ejection_by_ejector_pins" type="Double" value="30" unit="kW"/>
 <var name="electricpower_restorationof_ejector_pins" type="Double" value="30" unit="kW"/>
 <!-- state data value setting -->
 <var name="elapsed_time" type="Double" value="0"/>
 <var name="total_energyconsumption" type="Double" value="0"/>
 <!-- production operation value setting -->
 <var name="op_power" type="Boolean" value="false"/>
 <var name="op_cylinder_cleaning_instruction" type="Boolean" value="false"/>
 <var name="op_cylinder_cleaning_time" type="Double" value="10"/>
 <var name="op_injection_molding_instruction" type="Boolean" value="false"/>
 <var name="op_injection_molding_time" type="Double" value="10"/>
 <var name="op_mold_change_time" type="Double" value="10"/>
 <var name="op_mold_cleaning_instruction" type="Boolean" value="false"/>
 <var name="op_mold_cleaning_time" type="Double" value="10"/>
 <var name="op_mold_id_old" type="Double" value="10"/>
 <var name="op_mold_id" type="Double" value="100"/>
 <var name="op_pca_id_old" type="Double" value="20"/>
 <var name="op_pca_id" type="Double" value="200"/>
 <var name="op_opening_mold_time" type="Double" value="10"/>
 <var name="op_plan_quantity" type="Double" value="10"/>
 <var name="op_product_time" type="Double" value="10"/>
 <var name="op_restoreation_ejector_pins_time" type="Double" value="10"/>
 <var name="op_shutdown_time" type="Double" value="10"/>
 <var name="op_startup_time" type="Double" value="10"/>
 <var name="op_total_processing_quantity" type="Double" value="10"/>
</property_set>
<external_interaction_list>
 <external_interaction id="e1" name="robot_extraction">
  <send_to equipment_id="mer" message="robot_extraction_request"/>
 </external_interaction>
 <external_interaction id="e2" name="robot_completed_notice">
  <receive_from equipment_id="mer" message="robot_completed_notice"/>
 </external_interaction>
</external_interaction_list>
<behaviour>
 <!-- state list -->
 <state_list>
  <state id="s1" name="initial" status="on"/>
  <state id="s2" name="idling">
   <state_data>
    <elapsed_time>0</elapsed_time>
    <total_energyconsumption>0</total_energyconsumption>
   </state_data>
   <calculation_formula>
    <m:math>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> = </m:mo>
      <m:mrow>
       <m:mi>elapsed_time</m:mi>
       <m:mo> + </m:mo>
       <m:mn>1</m:mn>
      </m:mrow>
     </m:mrow>
    </m:math>
    <m:math>
     <m:mrow>
      <m:mi>total_energyconsumption</m:mi>
      <m:mo> = </m:mo>
      <m:mrow>
       <m:mi>elapsed_time</m:mi>
       <m:mo> * </m:mo>
       <m:mi>electricpower_idling</m:mi>
      </m:mrow>
     </m:mrow>
    </m:math>
   </calculation_formula>
  </state>
  <state id="s3" name="mold_change">
   <state_data>
```

```
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> + </m:mo>
      <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> * </m:mo>
      <m:mi>electricpower_mold_change</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>op_pca_id_old</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>op_pca_id</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
 </calculation_formula>
</state>
<state id="s4" name="mold_cleanig">
 <state_data>
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> + </m:mo>
     <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo>=</m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> * </m:mo>
     <m:mi>electricpower_mold_cleanig</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
 </calculation_formula>
</state>
<state id="s5" name="cylinder_cleaning">
 <state_data>
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
```

```
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> + </m:mo>
     <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> * </m:mo>
     <m:mi>electricpower_cylinder_cleaning</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
 </calculation_formula>
</state>
<state id="s6" name="injection_molding">
 <state_data>
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> + </m:mo>
     <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> * </m:mo>
     <m:mi>electricpower_injection_molding</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
 </calculation_formula>
</state>
<state id="s7" name="opening_of_a_mold">
 <state_data>
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> + </m:mo>
     <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
```

```
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> * </m:mo>
     <m:mi>electricpower_opening_of_a_mold</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
 </calculation_formula>
 <external_interaction id="e1" seq="post">
  <param>true</param>
 </external_interaction>
</state>
<state id="s8" name="ejection_by_ejector_pins">
 <state_data>
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> + </m:mo>
     <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> * </m:mo>
     <m:mi>electricpower_ejection_by_ejector_pins</m:mi>
    </m:mrow>
   </m:mrow>
  </m:math>
 </calculation_formula>
</state>
<state id="s9" name="restorationof_ejector_pins">
 <state_data>
  <elapsed_time>0</elapsed_time>
  <total_energyconsumption>0</total_energyconsumption>
 </state_data>
 <calculation_formula>
  <m:math>
   <m:mrow>
    <m:mi>elapsed_time</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> + </m:mo>
     <m:mn>1</m:mn>
    </m:mrow>
   </m:mrow>
  </m:math>
  <m:math>
   <m:mrow>
    <m:mi>total_energyconsumption</m:mi>
    <m:mo> = </m:mo>
    <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> * </m:mo>
      <m:mi>electricpower_restorationof_ejector_pins</m:mi>
    </m:mrow>
```

```
    </m:mrow>
   </m:math>
  </calculation_formula>
  <calculation_formula repeat="no">
   <m:math>
    <m:mrow>
     <m:mi>op_total_processing_quantity</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
       <m:mi>op_total_processing_quantity</m:mi>
       <m:mo> + </m:mo>
       <m:mn>1</m:mn>
     </m:mrow>
    </m:mrow>
   </m:math>
  </calculation_formula>
 </state>
</state_list>
<!-- state transition list -->
<state_transition_list>
 <state_transition id="t1" name="power_on">
  <!-- CONDITION: power on -->
  <!-- CONDITION: start-up time is elapsed -->
  <transition_condition>op_power? = true</transition_condition>
  <transition_timeout>op_startup_time</transition_timeout>
  <from_state id="s1" name="initial"/>
  <to_state id="s2" name="idling"/>
 </state_transition>
 <state_transition id="t2" name="power_off">
  <!-- CONDITION: power off -->
  <!-- CONDITION: shutdown time is elapsed -->
  <transition_condition>op_power? = false</transition_condition>
  <transition_timeout>op_shutdown_time</transition_timeout>
  <from_state id="s2" name="idling"/>
  <to_state id="s1" name="initial"/>
 </state_transition>
 <state_transition id="t3" name="mold_change">
  <!-- CONDITION: product input time reached -->
  <!-- CONDITION: the mold ID is different from the previous production -->
  <transition_condition>op_mold_id != op_mold_id_old</transition_condition>
  <transition_timeout>op_product_time</transition_timeout>
  <from_state id="s2" name="idling"/>
  <to_state id="s3" name="mold_change"/>
 </state_transition>
 <state_transition id="t4" name="mold_change_completed">
  <!-- CONDITION: mold change time is elapsed -->
  <transition_timeout>op_mold_change_time</transition_timeout>
  <from_state id="s3" name="mold_change"/>
  <to_state id="s2" name="idling"/>
 </state_transition>
 <state_transition id="t5" name="mold_cleaning">
  <!-- CONDITION: product input time reached -->
  <!-- CONDITION: mold cleaning instruction -->
  <transition_condition>op_mold_cleaning_instruction? = true</transition_condition>
  <transition_timeout>op_product_time</transition_timeout>
  <from_state id="s2" name="idling"/>
  <to_state id="s4" name="mold_cleanig"/>
 </state_transition>
 <state_transition id="t6" name="mold_cleaning_completed">
  <!-- CONDITION: mold cleaning time is elapsed -->
  <transition_timeout>op_mold_cleaning_time</transition_timeout>
  <from_state id="s4" name="mold_cleanig"/>
  <to_state id="s2" name="idling"/>
 </state_transition>
 <state_transition id="t7" name="cylinder_cleaning">
  <!-- CONDITION: product input time reached -->
  <!-- CONDITION: cylinder cleaning instruction -->
  <transition_condition>op_cylinder_cleaning_instruction? = true</transition_condition>
  <transition_timeout>op_product_time</transition_timeout>
  <from_state id="s2" name="idling"/>
  <to_state id="s5" name="cylinder_cleaning"/>
 </state_transition>
```

```
      <state_transition id="t8" name="cylinder_cleaning_completed">
       <!-- CONDITION: cylinder cleaning time is elapsed -->
       <transition_timeout>op_cylinder_cleaning_time</transition_timeout>
       <from_state id="s5" name="cylinder_cleaning"/>
       <to_state id="s2" name="idling"/>
      </state_transition>
      <state_transition id="t9" name="product_start">
       <!-- CONDITION: product input time reached -->
       <transition_condition>op_injection_molding_instruction? = true</transition_condition>
       <transition_timeout>op_product_time</transition_timeout>
       <from_state id="s2" name="idling"/>
       <to_state id="s6" name="injection_molding"/>
      </state_transition>
      <state_transition id="t10" name="injection_molding_completed">
       <!-- CONDITION: injection-molding time is elapsed -->
       <transition_timeout>op_injection_molding_time</transition_timeout>
       <from_state id="s6" name="injection_molding"/>
       <to_state id="s7" name="opening_of_a_mold"/>
      </state_transition>
      <state_transition id="t11" name="opening_mold_completed">
       <!-- CONDITION: opening-mold time is elapsed -->
       <transition_timeout>op_opening_mold_time</transition_timeout>
       <from_state id="s7" name="opening_of_a_mold"/>
       <to_state id="s8" name="ejection_by_ejector_pins"/>
      </state_transition>
      <state_transition id="t12" name="robot_completed_notice">
       <!-- CONDITION: reception of extraction-complete notice from robot -->
       <transition_external>e2</transition_external>
       <from_state id="s8" name="ejection_by_ejector_pins"/>
       <to_state id="s9" name="restorationof_ejector_pins"/>
      </state_transition>
      <state_transition id="t13" name="ejection_pins_completed">
       <!-- CONDITION: pin ejection time is elapsed -->
       <!-- CONDITION: number of producing product is unachieved -->
       <transition_condition>op_total_processing_quantity &gt; op_plan_quantity</transition_
condition>
       <transition_timeout>op_restoreation_ejector_pins_time</transition_timeout>

       <from_state id="s9" name="restorationof_ejector_pins"/>
       <to_state id="s6" name="injection_molding"/>
      </state_transition>
      <state_transition id="t14" name="order_completed">
       <!-- CONDITION: all process time is elapsed -->
       <!-- CONDITION: number of producing product is achieved -->
       <transition_condition>op_total_processing_quantity = op_plan_quantity</transition_
condition>
       <transition_timeout>op_restoreation_ejector_pins_time</transition_timeout>
       <from_state id="s9" name="restorationof_ejector_pins"/>
       <to_state id="s2" name="idling"/>
      </state_transition>
     </state_transition_list>
    </behaviour>
</ebc>
```

A.3 EBC item for a molding extraction robot

In ISO 16400-2: 2024, C.3, the example of the EBC template of a molding extraction robot is represented. The example of the EBC item using this EBC template can be described using XML as follows:

```
<ebc xmlns:op="product-operation.xml" xmlns:m="https://www.w3.org/1998/Math/MathML">

    <header>
     <template name="MOLDING_EXTRACTION_ROBOT" id="template_id_mer1"/>
    </header>

    <property_set>
     <!-- default value setting -->
     <var name="equipment_id" type="String" value="mer1"/>
     <var name="equipment_type" type="String" value="XXX"/>
     <var name="organization" type="String" value="XYZ"/>
```

```
 <var name="model_name" type="String" value="me_robot"/>
 <var name="external_size_hight" type="Double" value="100" unit="cm"/>
 <var name="external_size_width" type="Double" value="250" unit="cm"/>
 <var name="external_size_depth" type="Double" value="100" unit="cm"/>
 <var name="equipment_weight" type="Double" value="100" unit="kg"/>
 <var name="startup_time" type="Double" value="20" unit="sec"/>
 <var name="shutdown_time" type="Double" value="20" unit="sec"/>
 <var name="electricpower_idling" type="Double" value="20" unit="kW"/>
 <var name="electricpower_extraction_of_molded_part" type="Double" value="30" unit="kW"/>
 <var name="electricpower_releasing_of_moded_part" type="Double" value="30" unit="kW"/>
 <!-- state data value setting -->
 <var name="elapsed_time" type="Double" value="0"/>
 <var name="total_energyconsumption" type="Double" value="0"/>
 <!-- production operation value setting -->
 <var name="op_power" type="Boolean" value="false"/>
 <var name="op_extraction_time" type="Double" value="10"/>
 <var name="op_releasing_time" type="Double" value="10"/>
</property_set>
<external_interaction_list>
 <external_interaction id="e1" name="robot_extraction">
  <receive_from equipment_id="imm1" message="robot_extraction_request"/>
 </external_interaction>
 <external_interaction id="e2" name="robot_completed_notice">
  <send_to equipment_id="imm1" message="robot_completed_notice"/>
 </external_interaction>
</external_interaction_list>
<behaviour>
 <!-- state list -->
 <state_list>
  <state id="s1" name="initial" status="on"/>
  <state id="s2" name="idling">
   <state_data>
    <elapsed_time>0</elapsed_time>
    <total_energyconsumption>0</total_energyconsumption>
   </state_data>
   <calculation_formula>
    <m:math>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> = </m:mo>
      <m:mrow>
       <m:mi>elapsed_time</m:mi>
       <m:mo> + </m:mo>
       <m:mn>1</m:mn>
      </m:mrow>
     </m:mrow>
    </m:math>
    <m:math>
     <m:mrow>
      <m:mi>total_energyconsumption</m:mi>
      <m:mo> = </m:mo>
      <m:mrow>
       <m:mi>elapsed_time</m:mi>
       <m:mo> * </m:mo><!-- <m:mo>&times;</m:mo> -->
       <m:mi>electricpower_idling</m:mi>
      </m:mrow>
     </m:mrow>
    </m:math>
   </calculation_formula>
  </state>
  <state id="s3" name="extraction_of_molded_part">
   <state_data>
    <elapsed_time>0</elapsed_time>
    <total_energyconsumption>0</total_energyconsumption>
   </state_data>
   <calculation_formula>
    <m:math>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> = </m:mo>
      <m:mrow>
       <m:mi>elapsed_time</m:mi>
```

```
      <m:mo> + </m:mo>
      <m:mn>1</m:mn>
     </m:mrow>
    </m:mrow>
   </m:math>
   <m:math>
    <m:mrow>
     <m:mi>total_energyconsumption</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> * </m:mo><!-- <m:mo>&times;</m:mo> -->
      <m:mi>electricpower_extraction_of_molded_part</m:mi>
     </m:mrow>
    </m:mrow>
   </m:math>
  </calculation_formula>
  <external_interaction id="e2" seq="post">
   <param>true</param>
  </external_interaction>
 </state>
 <state id="s4" name="releasing_of_moded_part">
  <state_data>
   <elapsed_time>0</elapsed_time>
   <total_energyconsumption>0</total_energyconsumption>
  </state_data>
  <calculation_formula>
   <m:math>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> + </m:mo>
      <m:mn>1</m:mn>
     </m:mrow>
    </m:mrow>
   </m:math>
   <m:math>
    <m:mrow>
     <m:mi>total_energyconsumption</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> * </m:mo><!-- <m:mo>&times;</m:mo> -->
      <m:mi>electricpower_releasing_of_moded_part</m:mi>
     </m:mrow>
    </m:mrow>
   </m:math>
  </calculation_formula>
 </state>
</state_list>
<!-- state transition list -->
<state_transition_list>
 <state_transition id="t1" name="power_on">
  <!-- CONDITION: power on -->
  <!-- CONDITION: start-up time is elapsed -->
  <transition_condition>op_power? = true</transition_condition>
  <transition_timeout>startup_time</transition_timeout>
  <from_state id="s1" name="initial"/>
  <to_state  id="s2" name="idling"/>
 </state_transition>
 <state_transition id="t2" name="power_off">
  <!-- CONDITION: power off -->
  <!-- CONDITION: shutdown time is elapsed -->
  <transition_condition>op_power? = false</transition_condition>
  <transition_timeout>shutdown_time</transition_timeout>
  <from_state  id="s2" name="idling"/>
  <to_state id="s1" name="initial"/>
 </state_transition>
 <state_transition id="t3" name="extraction_start">
  <!-- CONDITION: reception of extraction-request from molding machine -->
```

```
      <transition_external>e1</transition_external>
      <from_state  id="s2" name="idling"/>
      <to_state  id="s3" name="extraction_of_molded_part"/>
     </state_transition>
     <state_transition id="t4" name="extraction_of_molded_parts_completed">
      <!-- CONDITION: extraction time is elapsed -->
      <transition_timeout>op_extraction_time</transition_timeout>
      <from_state id="s3" name="extraction_of_molded_part"/>
      <to_state id="s4" name="releasing_of_moded_part"/>
     </state_transition>
     <state_transition id="t5" name="releasing_of_molded_part_completed">
      <!-- CONDITION: releasing time is elapsed -->
      <transition_timeout>op_releasing_time</transition_timeout>
      <from_state id="s4" name="releasing_of_moded_part"/>
      <to_state  id="s2" name="idling"/>
     </state_transition>
    </state_transition_list>
   </behaviour>
</ebc>
```

A.4 EBC item for a mold temperature controller

In ISO 16400-2: 2024, C.4, the example of the EBC template of a mold temperature controller is represented. The example of the EBC item using this EBC template can be described using XML as follows:

```
<ebc xmlns:op="product-operation.xml" xmlns:m="https://www.w3.org/1998/Math/MathML">

   <header>
    <template name="MOLD_TEMPERATURE_CONTROLLER" id="template_id_mtc1"/>
   </header>

   <property_set>
    <var name="equipment_id" type="String" value="mtc1"/>
    <var name="equipment_type" type="String" value="XXX"/>
    <var name="organization" type="String" value="XYZ"/>
    <var name="model_name" type="String" value="mt_controller"/>
    <var name="external_size_hight" type="Double" value="100" unit="cm"/>
    <var name="external_size_width" type="Double" value="250" unit="cm"/>
    <var name="external_size_depth" type="Double" value="100" unit="cm"/>
    <var name="equipment_weight" type="Double" value="200" unit="kg"/>
    <var name="startup_time" type="Double" value="20" unit="sec"/>
    <var name="shutdown_time" type="Double" value="20" unit="sec"/>
    <var name="heating_time" type="Double" value="60" unit="sec"/>
    <var name="electricpower_heating" type="Double" value="40" unit="kW"/>
    <var name="electricpower_mold_temperture_control" type="Double" value="30" unit="kW"/>
    <!-- state data value setting -->
    <var name="elapsed_time" type="Double" value="0"/>
    <var name="total_energyconsumption" type="Double" value="0"/>
    <!-- production operation value setting -->
    <var name="op_power" type="Boolean" value="false"/>
   </property_set>

   <!-- external_interaction no difinition -->
   <external_interaction_list/>

   <behaviour>
    <!-- state list -->
    <state_list>
     <state id="s1" name="initial" status="on"/>
     <state id="s2" name="heating">
      <state_data>
       <elapsed_time>0</elapsed_time>
       <total_energyconsumption>0</total_energyconsumption>
      </state_data>
      <calculation_formula>
       <m:math>
        <m:mrow>
         <m:mi>elapsed_time</m:mi>
         <m:mo> = </m:mo>
         <m:mrow>
```

```
      <m:mi>elapsed_time</m:mi>
      <m:mo> + </m:mo>
      <m:mn>1</m:mn>
     </m:mrow>
    </m:mrow>
   </m:math>
   <m:math>
    <m:mrow>
     <m:mi>total_energyconsumption</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> * </m:mo>
      <m:mi>electricpower_heating</m:mi>
     </m:mrow>
    </m:mrow>
   </m:math>
  </calculation_formula>
 </state>
 <state id="s3" name="mold_temperture_control">
  <state_data>
   <elapsed_time>0</elapsed_time>
   <total_energyconsumption>0</total_energyconsumption>
  </state_data>
  <calculation_formula>
   <m:math>
    <m:mrow>
     <m:mi>elapsed_time</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
      <m:mi>elapsed_time</m:mi>
      <m:mo> + </m:mo>
      <m:mn>1</m:mn>
     </m:mrow>
    </m:mrow>
   </m:math>
   <m:math>
    <m:mrow>
     <m:mi>total_energyconsumption</m:mi>
     <m:mo> = </m:mo>
     <m:mrow>
       <m:mi>elapsed_time</m:mi>
       <m:mo> * </m:mo>
       <m:mi>electricpower_mold_temperture_control</m:mi>
     </m:mrow>
    </m:mrow>
   </m:math>
  </calculation_formula>
 </state>
</state_list>
<!-- state transition list -->
<state_transition_list>
 <state_transition id="t1" name="power_on">
  <!-- CONDITION: power on -->
  <!-- CONDITION: start-up time is elapsed -->
  <transition_condition>op_power? = true</transition_condition>
  <transition_timeout>startup_time</transition_timeout>
  <from_state id="s1" name="initial"/>
  <to_state id="s2" name="heating"/>
 </state_transition>
 <state_transition id="t2" name="mold_heating_completed">
  <!-- CONDITION: heating time is elapsed -->
  <transition_timeout>heating_time</transition_timeout>
  <from_state id="s2" name="heating"/>
  <to_state id="s3" name="mold_temperture_control"/>
 </state_transition>
 <state_transition id="t3" name="power_off">
  <!-- CONDITION: power off -->
  <!-- CONDITION: shutdown time is elapsed -->
  <transition_condition>op_power? = false</transition_condition>
  <transition_timeout>shutdown_time</transition_timeout>
  <from_state id="s3" name="mold_temperture_control"/>
```

```
      <to_state id="s1" name="initial"/>
     </state_transition>
    </state_transition_list>
   </behaviour>
</ebc>
```

Annex B
(informative)

Implementation example of a construction processor

Figure B.1 shows an example of constructing an equipment instance model in the target multi-agent environment NetLogo[1)].

The construction processor corresponds to an XSLT processor. The construction processor is a kind of software that takes an EBC item (*.xml) as an input and performs the construction procedure (*.xsl), and constructs an equipment instance model (*.nlogo). The construction processor will encounter specific patterns that are described in the construction rules. Based on the pattern that is found, the construction processor will perform the construction that is associated with the matched pattern. The resulting constructed output is an equipment instance model.

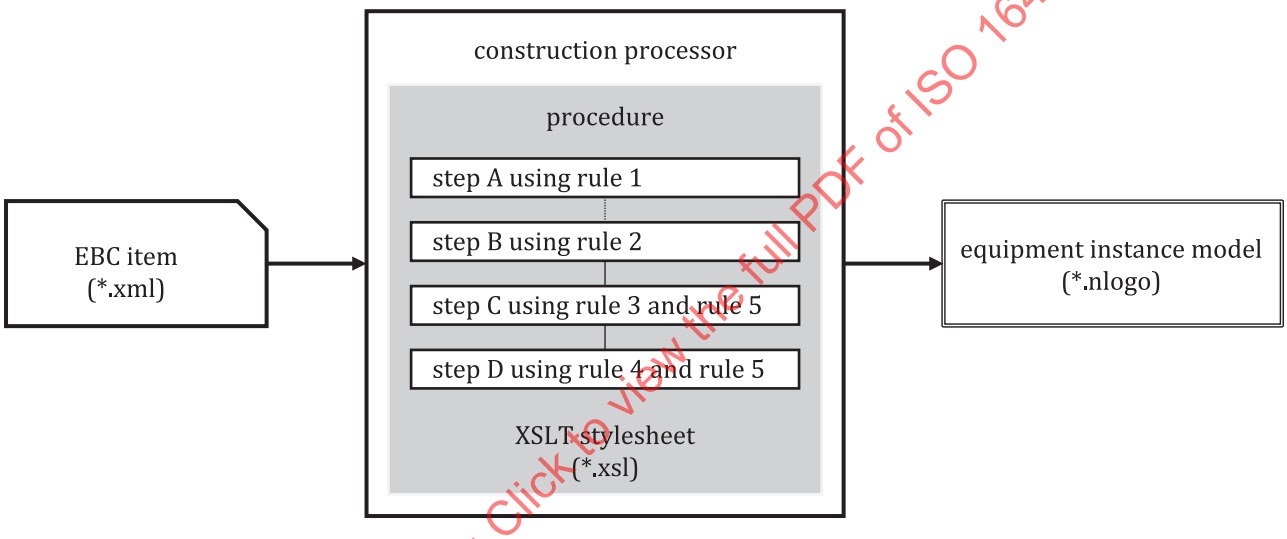


Figure B.1 — Construction of an equipment instance model in NetLogo

The following is the construction procedure using construction rules in NetLogo.

Step A: An equipment instance model is constructed to have a general information specified in the EBC item based on the construction rule 1. Step A procedure is the same sequence as the flow chart for applying the rule 1 indicated by a) and b) in Figure B.2 and is marked as Step A in the following XLST code. When the Step A procedure is performed by the construction processor, the general information such as an EBC template name and an EBC template identifier is constructed. Rule 1 consists of the following XSL elements.

— XSL element: <xsl:value-of select>, <xsl:text>

1) NetLogo is an example of a suitable product available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO of this product.

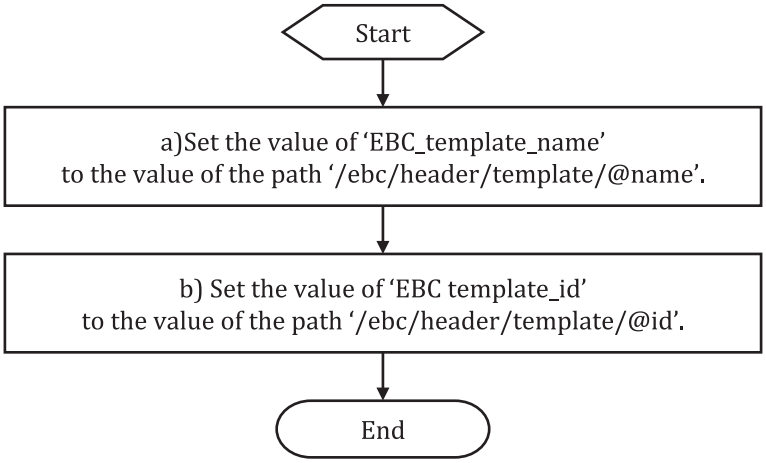


Figure B.2 — Step A using Rule 1

The following XSLT codes are an example of Step A using Rule 1.

```
;; (Step A)
;; /ebc/header/template
<xsl:text>;; EBC_template_name : </xsl:text>
<xsl:value-of select="/ebc/header/template/@name"/>
<xsl:text>&#xa;</xsl:text>
<xsl:text>;; EBC_template_id : </xsl:text>
<xsl:value-of select="/ebc/header/template/@id"/>
```

Step B: An equipment instance model is constructed to have a processable description of each element of a property set specified in the EBC item based on the construction rule 2. Step B procedure is the same sequence as the flow chart for applying the rule 2 indicated by a), b), c), and d) in Figure B.3 and is marked as Step B in the following XLST code. When this Step B procedure is performed by the construction processor, all variables are defined and set the defined variables as initial values or default values. Rule 2 consists of the following XSL elements and NetLogo keywords.

— NetLogo keyword: to … end, set

— XSL element: <xsl:for-each select>, <xsl:if test>, <xsl:value-of select>, <xsl:text>

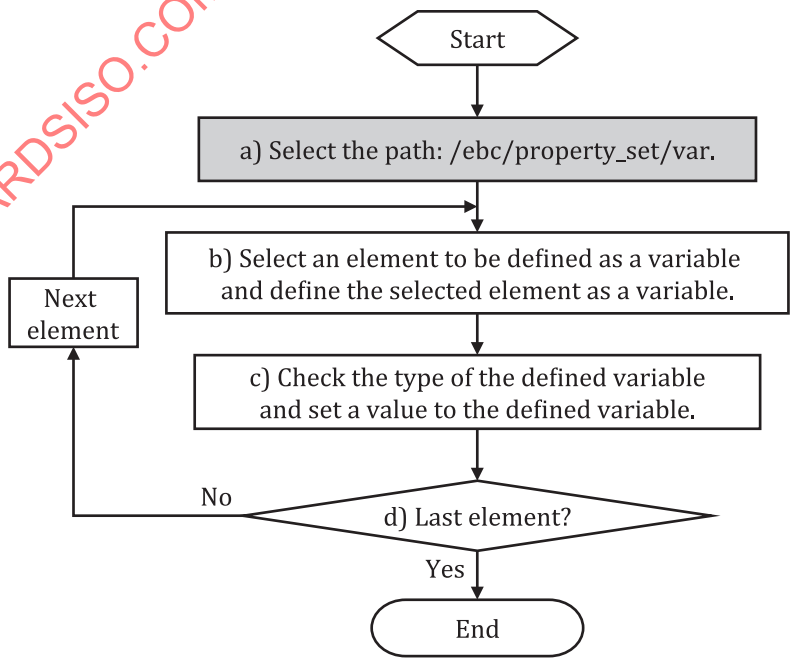


Figure B.3 — Step B using Rule 2

The following XSLT codes are an example of the procedure of Step B using Rule 2.

```
;; (Step B)
to setup-'plural of equipment name'
   ;; /ebc/property_set/var
   <xsl:for-each select="/ebc/property_set/var">
    set   <xsl:value-of select="@name"/>
    <xsl:if test="@type='Double'">
     <xsl:text>&#32;</xsl:text>
     <xsl:value-of select="@value"/>
    </xsl:if>
    <xsl:if test="@type='Boolean'">
     <xsl:text>?</xsl:text>
     <xsl:text>&#32;</xsl:text>
     <xsl:value-of select="@value"/>
    </xsl:if>
    <xsl:if test="@type='String'">
     <xsl:text>&#32;</xsl:text>
     <xsl:text>"</xsl:text>
     <xsl:value-of select="@value"/>
     <xsl:text>"</xsl:text>
    </xsl:if>
    <xsl:text>&#32;</xsl:text>
   </xsl:for-each>
End
```

Step C: An equipment instance model is constructed to have a processable description of each state of either the behaviour or each external interaction, or both, specified in the EBC item based on the construction Rule 3 and Rule 5. Step C procedure is the same sequence as the flow chart for applying the Rule 3 indicated by a), b), c) and i) and the Rule 5 indicated by d), e), f), g) and h) in Figure B.4 and is marked as Step C in the following XLST code. When this Step C procedure is performed by the construction processor, all state procedure of either behaviour(s) or an external interaction(s), or both, are implemented. Rule 3 and Rule 5 consist of the following XSL elements and NetLogo keywords.

— NetLogo keyword: to … end, wait, if, ifelse, set

— XSL element: <xsl:for-each select>, <xsl:variable name>, <xsl:if test>, <xsl:value-of select>, <xsl:text>, <xsl:apply-templates select>