International Standard

ISO 16400-2

First edition
2024-02

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

## Part 2: Formal description of a catalogue template

*Systèmes d'automatisation et intégration — Catalogues de comportement des équipements pour les systèmes de production virtuelle —*

*Partie 2: Description formelle d'un modèle de catalogue*



Reference number
ISO 16400-2:2024(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 documents 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 proposes guidelines to generate an executable representing the dynamic behaviour of a nominal or physical instance of an equipment. Such executable 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 of the evolution of smart manufacturing.

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

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

# Part 2:
# Formal description of a catalogue template

## 1 Scope

This document specifies a formal structure and building rules for an equipment behaviour catalogue (EBC) template.

The formal structure of an EBC template represents a schema for descriptions of behaviour and related entities.

Building rules for an EBC template provide required processes and compliance criteria to construct an EBC template.

## 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*

## 3 Terms and definitions

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

## 4 Abbreviated terms

| | |
|---|---|
| ID | Identifier |
| JSON | JavaScript Object Notation |
| mathML | Mathematical Markup Language |
| UML | Unified Modeling Language |
| XML | eXtensible Markup Language |

# 5 Requirements for an EBC template

A virtual production system is constructed as a simulation system on a production behaviour model which is configured by applying a production process model and a production system model (see ISO 16400-1:2020, Figure 6).

A production system model can be constructed as a multi-agent system of equipment agents, i.e. behaviour of a production system can be modelled using software agent technology. However, it is difficult for the user of a manufacturing simulation system, such as a production system designer and an operator, to write a software program of agents. There are requirements that a user can construct a virtual production system by selecting adequate EBC items from an EBC repository. When an EBC item is provided, it becomes possible to automatically generate an equipment instance model as a software agent from the EBC item. A production system model can be constructed as a combination of equipment instance models.

An EBC item is an instance of an EBC template, and an EBC template is a schema representing a model for each equipment including its process model as behaviour description. An EBC template shall include the following elements as essential elements:

— a set of properties;

— description of behaviour.

An EBC template can include the following element:

— specification of external interactions.

In smart manufacturing, a virtual production system is constructed based on the digital twin concept. For this purpose, further entities have to be considered, as they are specified in a production lifecycle information model which is out of scope for this document. An EBC item is a digital description of properties and behaviour of a physical equipment. A virtual equipment can be constructed by referring a corresponding EBC item. An EBC item works as a bridge between physical equipment and virtual equipment in a digital twin.

An EBC shall have interoperability including semantics in order to fulfil the requirements mentioned above.

# 6 Formal structure of an EBC template

## 6.1 General

An EBC template shall include descriptions which specify a property set, behaviour and external interactions. Figure 1 shows a conceptual structure of an EBC template.

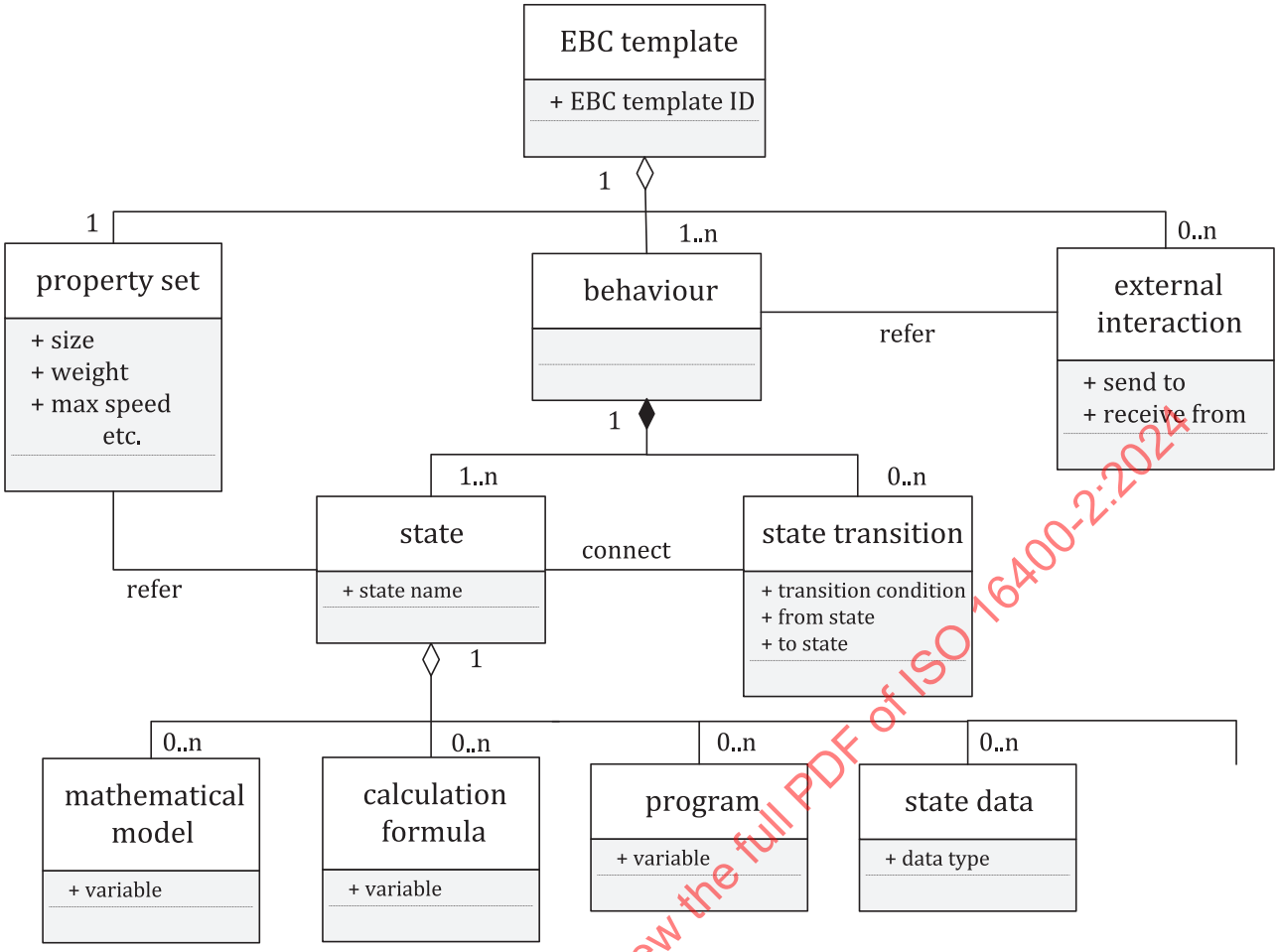


**Figure 1 — Conceptual structure of an EBC template**

An EBC shall be supplied in a common and independently usable form. The EBC template shall be created as a basic form for each equipment type. There is no hierarchical relation like class among EBC templates. Templates for the same equipment type have the same behaviour description structure about states and about relationships among states. There is a possibility that the granularity of descriptions is different.

The structure of an EBC item shall follow the used template. From one template, one or more EBC items may be created corresponding to each individual equipment of the same equipment type. The created EBC items are grouped according to the type of the equipment and stored in the repository. An EBC item is created by filling up the corresponding EBC template by concrete equipment data. Usually, an EBC item is created by setting values on the elements in the property set of the corresponding EBC template by concrete equipment data. For example, an EBC item for a specific equipment series contains equipment data as designed by the equipment maker such as equipment specification data. An EBC item for a specific individual equipment in a specific production line contains physical equipment data such as operational data in addition to equipment specification data.

Annex A shows the relationship between the EBC template and EBC items and variety of EBC items using two examples.

Information about an EBC template itself shall be described in a header part of the template. The corresponding equipment type is included in this information. When creating the EBC template for the specific equipment model/series of a specific equipment maker, the maker and the equipment model/series are described in the property set as ‘equipment maker name’ and ‘equipment type name’.

## 6.2 Property set

Elements in a property set are described in standards including ISO/IEC Guide 77,[1] the ISO 13399 series,[2] ISO 13584-42[3] and the IEC 61360 series (Common Data Dictionary, CDD).[6] Values of some or all elements in a property set are provided when an EBC item is created. A property set may include:

— profile data;

EXAMPLE 1 Equipment maker name, equipment type name, specific equipment name (value is given when creating an EBC item for specific physical equipment).

— specification data (not dynamically affecting behaviour);

EXAMPLE 2 Size, weight.

— operational limits.

EXAMPLE 3 Maximum cutting speed, moving range.

## 6.3 Behaviour

Behaviour of an equipment is defined by a composition of states and state transitions with transition conditions. A state can be represented by mathematical models, calculation formulae, programs, state data and/or additional necessary entities. State transition occurs when transition conditions are met.

The state description is the basis of behaviour description in an EBC. Equipment can be in various states, e.g. idling, under operation and under maintenance. The data values of parameters such as operation time and electric energy consumption at each state are calculated using mathematical models, calculation formulas and/or programs corresponding to the state. A formula or a mathematical model can include parameters/variables. Values for parameters/variables are given from product data, scheduling data and/or operation data as input data through external interaction, from calculation results of behaviour simulation and/or from data description in the property set.

Behaviour shall be represented as dynamic properties according to states and state transitions in an EBC template. Dynamic properties are described by:

— states of an equipment;

EXAMPLE 1 Stand-by, in operation.

NOTE 1 A state can be detailed as an aggregation of substates.

NOTE 2 A name of a state is described in line with the standard, e.g. the IEC 61360 series (CDD)[6] and the ISO 14955 series,[4] because of ensuring semantic interoperability.

— state transition.

EXAMPLE 2 Material input, power on, operation start, operation finish, transition trigger.

The state can be described and represented by:

— mathematical models, calculation formulas and/or programs;

EXAMPLE 3 Calculation formula for energy consumption, calculation formula for operation time.

NOTE 3 External interactions can be included in a state. The state transition can be described and represented by:

— a transition condition formula;

— source and destination states;

— external interaction as a trigger.

EXAMPLE 4 Operation order, operation result.

The representation of state/state transition can include:

— data values (parameter values);

— data variables.

NOTE 4 The name of the data (parameter)/variable is described in line with standards, such as the IEC 61360 series (CDD)[6] and the ISO 22400 series (KPI)[5] because of ensuring semantic interoperability.

## 6.4 External interaction

External interaction is the message data from/to other outside equipment. External interaction can be described by referring to the information models shared with the outside equipment. Some of the information models are supplied by existing standards, such as the IEC 62541 series (OPC UA)[7] and the IEC 62714 series (AutomationML).[8] The external interaction message can include:

— interface and protocol;

— communication partner;

— description about message data;

— content of message data.

## 6.5 Formal description of an EBC template

The EBC template shall be described using a data description language.

EXAMPLE XML, MathML (for a formula), JSON.

An example of a formal description structure is shown in Figure 2. An EBC template shall contain all of the following elements:

a) header;

   — description language name;

   — template name;

   — template identifier;

   — equipment type;

   — referenced dictionaries;

   NOTE 1 The value of "equipment type" is provided using the term in line with the referenced dictionary.

   NOTE 2 "Referenced dictionaries" are standards and publicly available dictionaries which are referred to and used when describing an EBC template.

   NOTE 3 "Item name" and "Item identifier" are added when an EBC item is created using the template. Their values are also provided.

b) property set;

c) behaviour;

d) external interaction;

NOTE 4 Additional attributes can be defined.

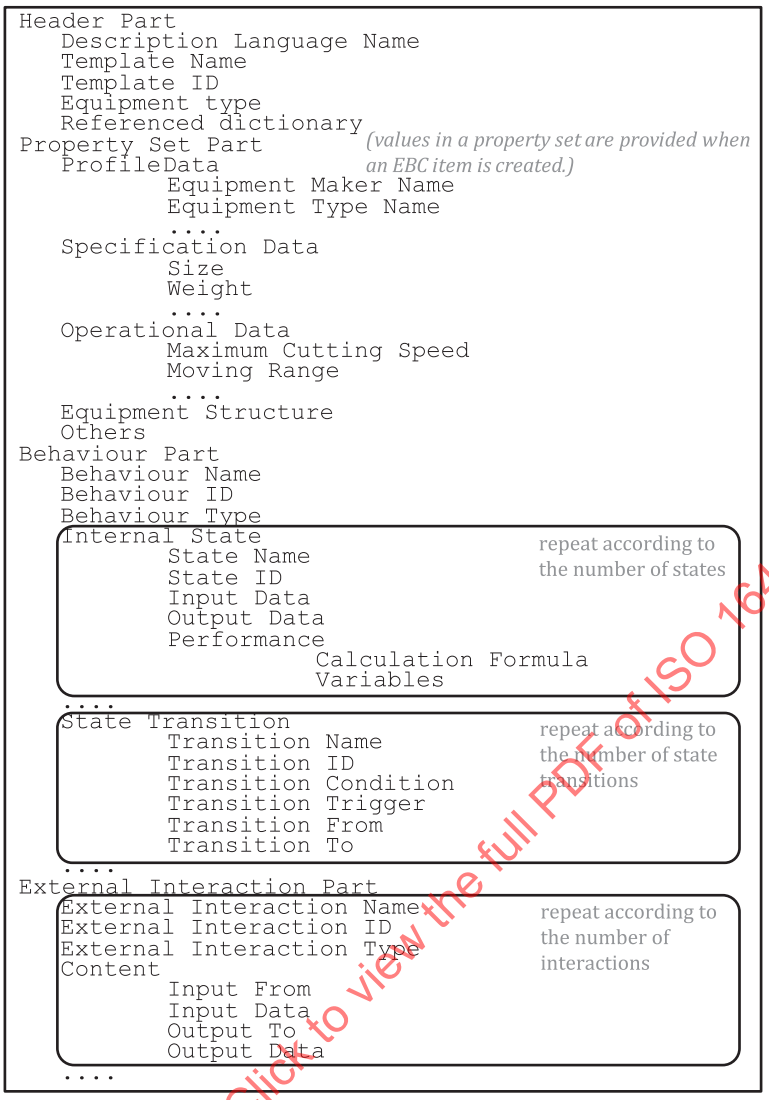


**Figure 2 — Example of a formal description structure for an EBC template**



# 7 Building rules for an EBC template

## 7.1 Building procedure

The procedure for building an EBC template is as follows:

— Step 1: Analyse functions that an equipment provides. The analysis is to list possible states of the equipment and relationships among the states including state transition conditions. Generally, an equipment becomes active when it gets an operation command with data through external interaction. Describe the listed states and state transitions in the equipment. The listed states and relationships including state transition conditions should be described by using formal methods.

EXAMPLE UML state chart, Petri-Net and IDEF3.

— Step 2: List required data, formulae and mathematical models for a dynamic calculation of the parameter's value at each state. Data include variables to which concrete values are assigned when a virtual production system is constructed or when simulation is executed on a virtual production system, i.e. some of these variables are provided by external interactions from outside of the equipment. These data are described as variables in the EBC template and EBC items. Some of the variables get values when an EBC item is created.

— Step 3: Integrate a property description, an external interaction description and a behaviour description, which are listed in the above steps. Create an EBC template by describing the integrated descriptions using the data description language.

Applied examples of the building procedure above for EBC templates are shown in Annex B and Annex C.

## 7.2 Building criteria

The conformance of an EBC template shall be tested according to the criteria below.

The following criteria shall be applied when building an EBC template:

a) An EBC template includes descriptions about a property set, behaviour and external interactions.

b) An EBC template is described in machine readable form.

c) An EBC template has necessary information items to generate an equipment instance model.

NOTE An EBC item is created from an EBC template. An equipment instance model is generated from an EBC item.

In addition, the following criteria should be applied when building an EBC template:

— Terms used in EBC template descriptions are from commonly referenceable standardized dictionary.

— An EBC template has interoperability including semantics.

The evaluation of the quality of the EBC template is out of scope of this document. Whether the integrity and granularity of the EBC template completely describes the equipment differs depending on the usage purpose of users. Users should evaluate the quality of an EBC with their own measurement means.

# Annex A
(informative)

# EBC template and types of EBC items

One EBC template is prepared for one equipment type in one maker. Usually, one equipment type consists of several series. There are several model types in one series. One series can be an EBC item. One model type can be an EBC item. One equipment in one model type can be an EBC item. All of them are described using the same template.

An EBC item has wide granularity. An EBC template is applicable for this wide granularity. An EBC item of specific individual equipment has a description of its product ID, i.e. it is possible that an EBC item for equipment will not have a description of its product ID. If the equipment has some options which a customer orders, its EBC item has a description of the product ID and options. An EBC item can have the data either of shipment inspection or usage history, or both, for the equipment.

Figure A.1 and Figure A.2 show the relationships between the EBC templates and various EBC items for the same example equipment series: Lathe X series. The Lathe X series has a hierarchical structure. Lathe X (X-series by the company Y) is the highest class in the hierarchy. Lathe X-s (simple lathe in X-series) and Lathe X-c (complex lathe in X-series) are subclasses of Lathe X. Lathe X-s-#A (type #A in X-series) is a specific equipment type in subclass Lathe X-s. Lathe X-s-#A-nnnn (serial no. nnnn of type #A) is a produced individual equipment in the type #A.

In Figure A.1, the EBC template is prepared corresponding to the class Lathe X. Lathe X-s, Lathe X-c, Lathe X-s-#A, Lathe X-s-#A-nnnn, etc are catalogued as EBC items using this EBC template. The EBC template for Lathe X has applicability for model descriptions of all lower classes in the hierarchy.

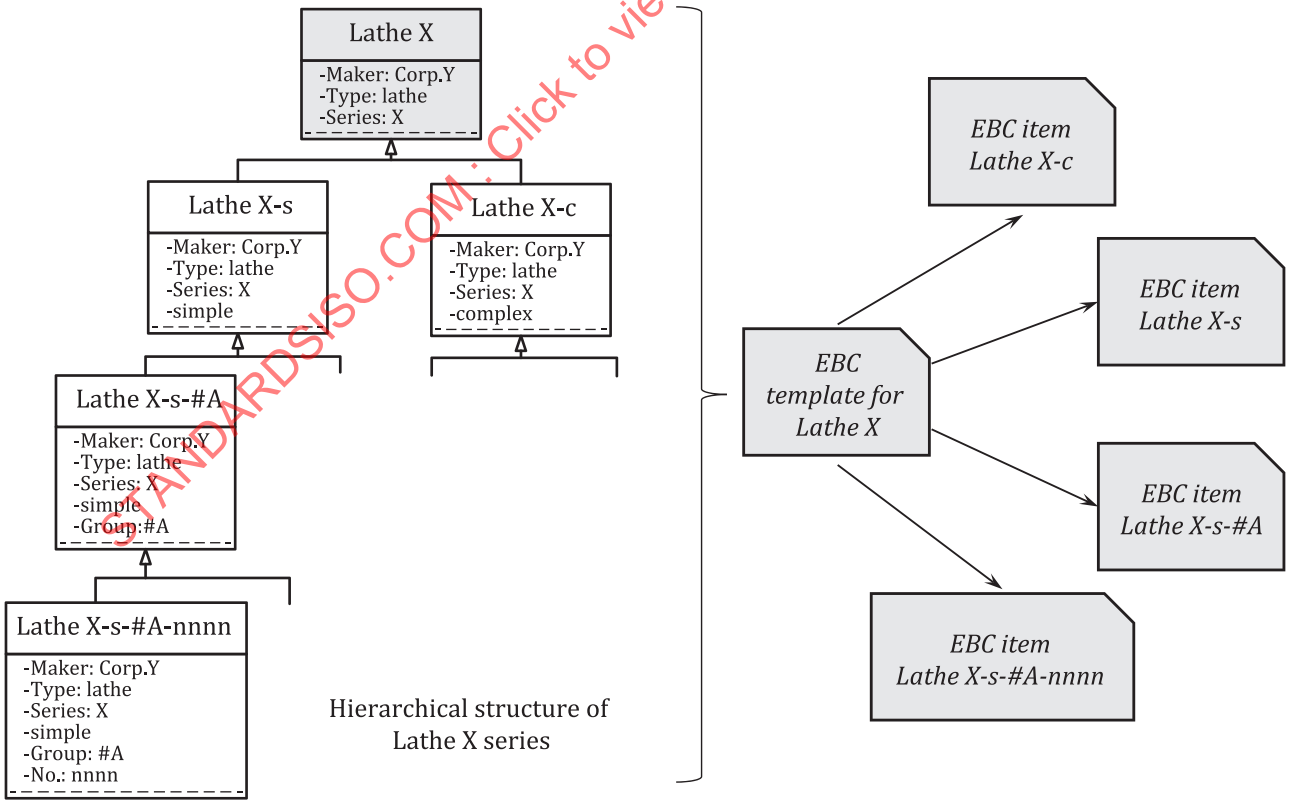


**Figure A.1 — EBC items which are generated using the EBC template for Lathe X**

When higher granularity of behaviour descriptions is required, EBC templates are prepared corresponding to one level lower classes than the class Lathe X. In Figure A.2, EBC templates are prepared corresponding to each of Lathe X-s and Lathe X-c. Here, Lathe X-s and Lathe X-c are treated as different equipment types. Lathe X-c has more complex behaviour than Lathe X-s. Lathe X-s, Lathe X-s-#A, Lathe X-s-#B, Lathe X-s-#A-nnnn, etc are catalogued as EBC items using the EBC template for Lathe X-s. Lathe X-c, Lathe X-c-#A, Lathe X-c-#C, etc are catalogued as EBC items using the EBC template for Lathe X-c.

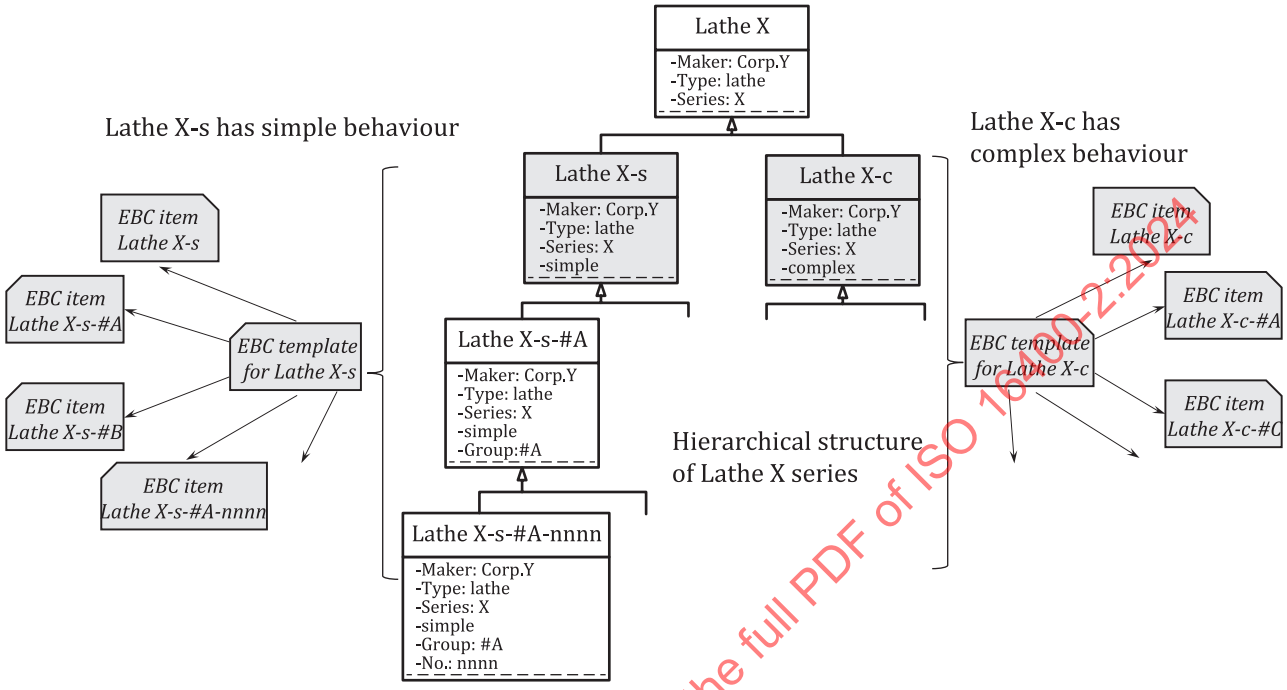


**Figure A.2 — EBC items which are generated using the EBC templates for Lathe X-s and Lathe X-c**

The EBC item for Lathe X-s-#A-nnnn corresponds to a specific individual equipment. An equipment instance model which is constructed from the EBC item for Lathe X-s-#A-nnnn is a physical equipment instance model of Lathe X-s-#A-nnnn. Equipment instance models which are constructed from the other EBC items are nominal equipment instance models. If data about a specific individual equipment are added when construction of an equipment instance model from the other EBC items are nominal equipment instance models, a physical equipment instance model can be constructed.

# Annex B
(informative)

# Example building of EBC templates for a printed circuit assembly (PCA) line

## B.1 Equipment on a printed circuit assembly (PCA) line

The structure of a PCA line is shown in Figure B.1. A PCA line consists of a solder paste printer, several electronic part mounters and a reflow soldering oven. This equipment is connected in sequence from the printer to the reflow soldering oven. The number of electronic part mounters changes depending on the capacity of the line.

A printed circuit board (PCB) is input to the line and a PCA is output from the line. When a blank PCB is input to a solder paste printer, the production process is started.

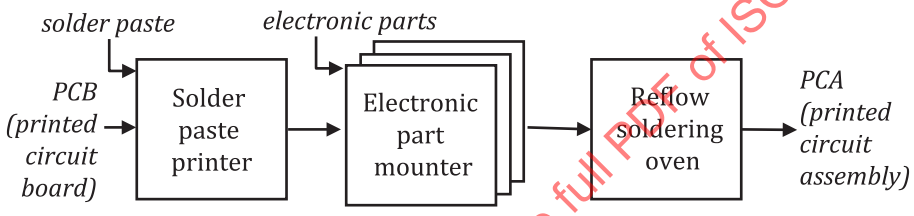


**Figure B.1 — Structure of printed circuit assembly (PCA) line**

According to the building procedure for EBC templates, templates for a solder paste printer, an electronic part mounter and a reflow soldering oven are prepared.

In this example, EBC templates are simplified by omitting sub activity descriptions and by focusing on electric energy consumption. Other focusing points which are related to the equipment's behaviour can be added in the same way.

## B.2 Building an EBC template of a solder paste printer

A solder paste printer is an equipment used for painting solder paste on PCBs by screen printing. A solder paste printer takes the first position on the PCA production line. The behaviour of a solder paste printer is modelled as an activity flow as shown in Figure B.2. After power on, a printer is in "idling" state. In "idling" state the printer consumes idling electric power. A printer moves to "stop" state during a breakdown or when the power is off. When a PCB is input to the printer, the printer starts the operation sequences. As preparation before printing, a metal mask is changed if necessary and solder paste is replenished if needed. The first operation is setting up the PCB. The second is the setting of the mask metal. The third is stage moving. The fourth is printing. The fifth is releasing the metal mask. The sixth and the last is the output of the PCB. Then the printer reverts back to "idling" state.

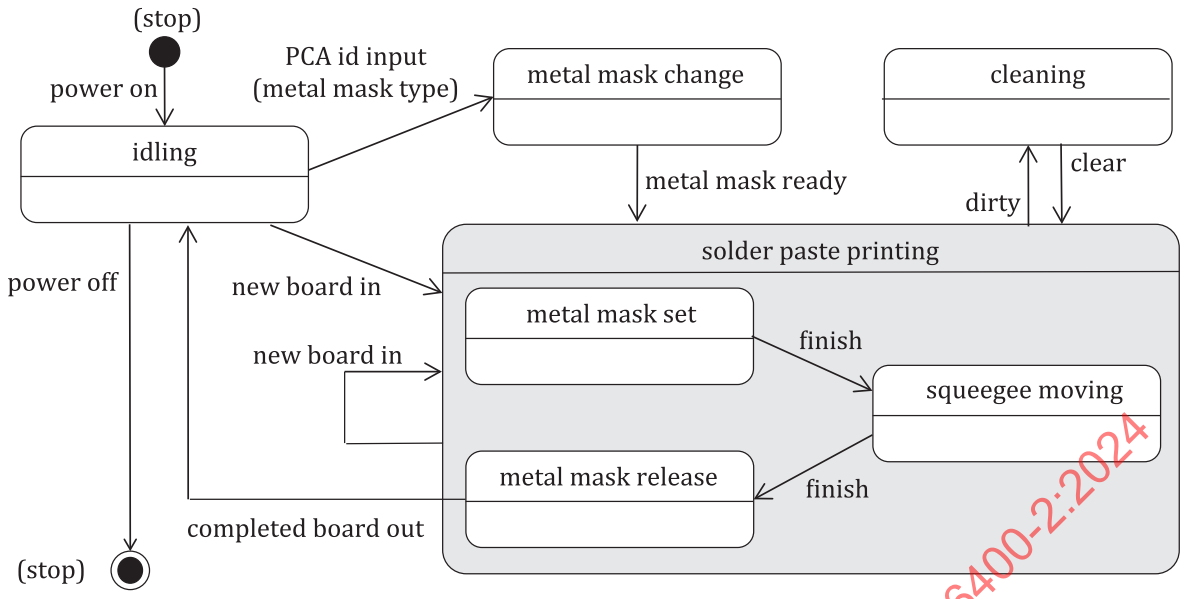


**Figure B.2 — Activity diagram for a solder paste printer**

The building procedure of an EBC template is applied to a solder paste printer as follows:

a) Listing of states

- stop;
- idling;
- metal mask change;
- cleaning;
- solder paste printing;

(sequence of following three substates)

- metal mask set;
- squeegee moving;
- metal mask release;

b) Listing of state transitions

- from stop into idling (when power on);
- from idling into stop (when power off);
- from idling into metal mask change (when PCB input time is reached and when PCA board id is different from the previous production);
- from metal mask change into solder paste printing (when metal mask change time is elapsed);
- from idling into solder paste printing (when PCB input time is reached);
- from solder paste printing into cleaning (when count of printing is reached to the cleaning required point);
- from cleaning into solder paste printing (when cleaning completed);

- from solder paste printing into solder paste printing (when no cleaning is required and production is incomplete);
- from solder paste printing into idling (when number of producing product is achieved);

c) Listing of static parameters (except common items, e.g. equipment ID, equipment type, model name, equipment weight)

- start-up time of equipment [sec];
- shutdown time of equipment [sec];
- cleaning time [sec];
- electric power of idling [kW];
- electric power of metal mask change [kW];
- electric power of cleaning [kW];
- electric power of solder paste printing [kW];

d) Listing of variables (providing by calculation formula)

- elapsed time of idling [sec];
- elapsed time of metal mask change [sec];
- elapsed time of cleaning [sec];
- elapsed time of solder paste printing [sec];
  - elapsed time of metal mask set [sec];
  - elapsed time of squeegee moving [sec];
  - elapsed time of metal mask release [sec];
- total energy consumption of idling [kWh];
- total energy consumption of metal mask change [kWh];
- total energy consumption of cleaning [kWh];
- total energy consumption of solder paste printing [kWh];
- total processing quantity;

e) Listing of variables which are input values when construction of virtual system or when virtual operation

- PCA board id;
- previous processed PCA board id;
- planned production quantity;
- production starting time;
- count of printing before cleaning is required;
- metal mask change time;
- solder paste printing time;
  - metal mask set time [sec]

— squeegee moving time [sec];

— metal mask release time [sec].

Finally, the EBC template for a solder paste printer is described using XML as follows:

```
<?xml version="1.0" standalone="yes"?>
<ebc
    xmlns:opeartion="product-operation.xml"
    xmlns:m="https://www.w3.org/1998/Math/MathML">

  <header>
    <template_name>SOLDER PASTE PRINTER</template_name>
    <template_id>xxxx</template_id>
  </header>

  <property_set>
    <!-- default value setting -->
    <equipment_id>1</equipment_id>
    <equipment_type>XXX</equipment_type>
    <organization>XYZ Corporation</organization>
    <model_name>YYY</model_name>
    <external_size unit="cm">H=100,W=250,D=100</external_size>
    <equipment_weight unit="kg">200</equipment_weight>
    <startup_time unit="sec">20</startup_time>
    <shutdown_time unit="sec">20</shutdown_time>
    <electricpower_idling unit="kW">30</electricpower_idling>
    <electricpower_metal_mask_change unit="kW">40</electricpower_metal_mask_change>
    <electricpower_cleaning unit="kW">40</electricpower_cleaning>
    <electricpower_solder_paste_printing unit="kW">80</electricpower_solder_paste_printing>
    <cleaning_time unit="sec">5</cleaning_time>
  </property_set>

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

  <behaviour>
    <!-- state list -->
    <state_list>
      <state id="s1" name="stop" state="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>&times;</m:mo>
                <m:mi>property_set:electricpower_idling</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
      </state>
      <state id="s3" name="metal_mask_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>&times;</m:mo>
          <m:mi>property_set:electricpower_metal_mask_change</m:mi>
        </m:mrow>
      </m:mrow>
    </m:math>
    <m:math>
      <m:mrow>
        <m:mi>operation:pca_id_old</m:mi>
        <m:mo>=</m:mo>
        <m:mrow>
          <m:mi>operation:pca_id</m:mi>
        </m:mrow>
      </m:mrow>
    </m:math>
  </calculation_formula>
</state>
<state id="s4" name="cleaning">
  <state_data>
    <elapsed_time>0</elapsed_time>
    <total_energyconsumption>0</total_energyconsumption>
    <actual_cleaning_quantity>0</actual_cleaning_quantity>
  </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>&times;</m:mo>
          <m:mi>property_set:electricpower_cleaning</m:mi>
        </m:mrow>
      </m:mrow>
    </m:math>
  </calculation_formula>
  <calculation_formula repeat="no">
    <m:math>
      <m:mrow>
        <m:mi>actual_cleaning_quantity</m:mi>
```

```
            <m:mo>=</m:mo>
            <m:mrow>
              <m:mn>0</m:mn>
            </m:mrow>
          </m:mrow>
        </m:math>
      </calculation_formula>
    </state>
    <state id="s5" name="solder_paste_printing">
      <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>&times;</m:mo>

<m:mi>property_set:electricpower_solder_paste_printing</m:mi>
            </m:mrow>
          </m:mrow>
        </m:math>
      </calculation_formula>
      <!-- substates -->
      <state_list>
        <!-- substate:metal_mask_set -->
        <state id="s5-1" name="metal_mask_set">
          <state_data>
            <elapsed_time>0</elapsed_time>
          </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>
          </calculation_formula>
        </state>
        <!-- substate:squeegee_moving -->
        <state id="s5-2" name="squeegee_moving">
          <state_data>
            <elapsed_time>0</elapsed_time>
          </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>operation:total_processing_quantity</m:mi>
              <m:mo>=</m:mo>
              <m:mrow>
                <m:mi>operation:total_processing_quantity</m:mi>
                <m:mo>+</m:mo>
                <m:mn>1</m:mn>
              </m:mrow>
            </m:mrow>
          </m:math>
          <m:math>
            <m:mrow>
              <m:mi>actual_cleaning_quantity</m:mi>
              <m:mo>=</m:mo>
              <m:mrow>
                <m:mi>actual_cleaning_quantity</m:mi>
                <m:mo>+</m:mo>
                <m:mn>1</m:mn>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
      </state>
      <!-- substate:metal_mask_release -->
      <state id="s5-3" name="metal_mask_release">
        <state_data>
          <elapsed_time>0</elapsed_time>
        </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>
        </calculation_formula>
      </state>
    </state_list>
  </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 operator="==">
      <param name="operation:power" />
      <param>on</param>
    </transition_condition>
    <transition_condition operator="==">
      <param name="property_set:startup_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s1"/>
    <to_state id="s2"/>
  </state_transition>
  <state_transition id="t2" name="power_off">
    <!-- CONDITION: power off -->
    <!-- CONDITION: shutdown time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:power" />
```

```
      <param>off</param>
    </transition_condition>
    <transition_condition operator="==">
      <param name="property_set:shutdown_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s1"/>
  </state_transition>
  <state_transition id="t3" name="pca_id_input">
    <!-- CONDITION: new PCB input time reached -->
    <transition_condition operator="==">
      <param name="operation:product_time" />
      <param name="system_time" />
    </transition_condition>
    <!-- CONDITION: PCA board id is different from the previous production -->
    <transition_condition operator="!=">
      <param name="operation:pca_id" />
      <param name="operation:pca_id_old" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s3"/>
  </state_transition>
  <state_transition id="t4" name="metal_mask_ready">
    <!-- CONDITION: metal mask change time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:metal_mask_change_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s3"/>
    <to_state id="s5"/>
  </state_transition>
  <state_transition id="t5" name="new_board_in">
    <!-- CONDITION: new PCB input time reached -->
    <transition_condition operator="==">
      <param name="operation:product_time" />
      <param name="system_time" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s5"/>
  </state_transition>
  <state_transition id="t6" name="dirty">
    <!-- CONDITION: count of printing is reached to the cleaning required point -->
    <transition_condition operator="==">
      <param name="operation:total_processing_quantity" />
      <param name="operation:cleaning_count" />
    </transition_condition>
    <from_state id="s5"/>
    <to_state id="s4"/>
  </state_transition>
  <state_transition id="t7" name="clear">
    <!-- CONDITION: cleaning time is elapsed -->
    <transition_condition operator="==">
      <param name="property_set:cleaning_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s4"/>
    <to_state id="s5"/>
  </state_transition>
  <state_transition id="t8" name="completed_board_out">
    <!-- CONDITION: number of producing product is achieved -->
    <transition_condition operator="==">
      <param name="operation:total_processing_quantity" />
      <param name="operation:plan_quantity" />
    </transition_condition>
    <transition_condition operator="==">
      <param name="operation:solder_paste_printing_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s5"/>
    <to_state id="s2"/>
  </state_transition>
```

```
      <state_transition id="t9" name="metal_mask_set">
        <!-- CONDITION: when production is uncompleted -->
        <transition_condition operator="&gt;">
          <param name="operation:processing_quantity" />
          <param name="operation:plan_quantity" />
        </transition_condition>
        <from_state id="s5"/>
        <to_state id="s5-1"/>
      </state_transition>
      <state_transition id="t10" name="mask_set_finish">
        <!-- CONDITION: metal mask set time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:metal_mask_set_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s5-1"/>
        <to_state id="s5-2"/>
      </state_transition>
      <state_transition id="t11" name="squeegee_moving_finish">
        <!-- CONDITION: squeegee moving time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:squeege_moving_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s5-2"/>
        <to_state id="s5-3"/>
      </state_transition>
      <state_transition id="t12" name="metal_mask_release_finish">
        <!-- CONDITION: metal mask release time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:metal_mask_relase_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s5-3"/>
        <to_state id="s5"/>
      </state_transition>
    </state_transition_list>
  </behaviour>
</ebc>
```

## B.3 Building an EBC template of an electronic part mounter

An electronic part mounter is an equipment used for picking and placing surface-mount electronic components, such as capacitors, resistors and integrated circuits onto the PCBs. An electronic part mounter takes the second position on the PCA production line. Usually, several part mounters are in the production line. The behaviour of an electronic part mounter is modelled as an activity flow as shown in Figure B.3.

After power on of the equipment, a part mounter is in "idling" state. A part mounter moves to "stop" state during a breakdown or when the power is off. When a PCB is input to the part mounter, the part mounter starts the operation sequences. As preparation before mounting, the required electronic parts are set and replenished. The first operation is placing of the PCB. The second is PCB position recognition and correction. The third is nozzle selection. The fourth is electronic part picking. The fifth is electronic part position recognition and correction. The sixth is electronic part mounting. The seventh and the last is output of the PCB. Then the part mounter reverts to "idling" state.

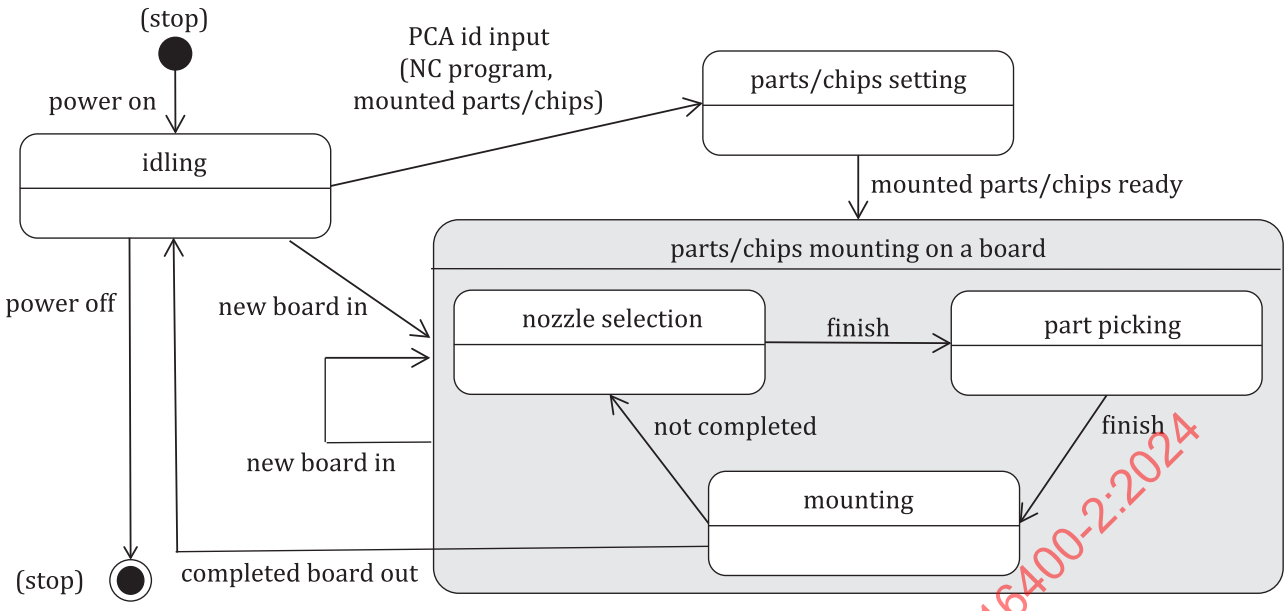


**Figure B.3 — Activity diagram for an electronic part mounter**

The building procedure of an EBC template is applied to an electronic part mounter as follows:

a) Listing of states

- stop;
- idling;
- parts/chips setting;
- parts/chips mounting on a board;

(sequence of the following three substates)

NOTE This sequence is repeated until completion of mounting all parts for a board.

- nozzle selection;
- part picking;
- mounting;

b) Listing of state transitions

- from stop into idling (when power on);
- from idling into stop (when power off);
- from idling into parts/chips setting (when PCB input time is reached and when PCA board id is different from the previous production);
- from parts/chips setting into parts/chips mounting on a board (when parts/chips setting time is elapsed);
- from idling into parts/chips mounting on a board (when PCB input time is reached);

- from parts/chips mounting on a board into parts/chips mounting on a board (when production is uncompleted);
- from parts/chips mounting on a board into idling (when number of producing product is achieved);

c) Listing of static parameters (except common items, e.g. equipment ID, equipment type, model name, equipment weight)

- starting-up time of equipment [sec];
- shutdown time of equipment [sec];
- electric power of idling [kW];
- electric power of parts/chips setting [kW];
- electric power of parts/chips mounting on a board [kW];

d) Listing of variables (providing by calculation formula)

- elapsed time of idling [sec];
- elapsed time of metal mask parts/chips setting [sec];
- elapsed time of parts/chips mounting on a board [sec];
  - elapsed time of nozzle selection [sec];
  - elapsed time of part picking [sec];
  - elapsed time of mounting [sec];
- total energy consumption of idling [kWh];
- total energy consumption of parts/chips setting [kWh];
- total energy consumption of parts/chips mounting on a board [kWh];
- total processing quantity;

e) Listing of variables which are input values when construction of virtual system or when virtual operation

- PCA board id;
- previous processed PCA board id;
- planned production quantity;
- production starting time;
- parts/chips setting time;
- mounting time per chip;
  - nozzle selection time per chip [sec];
  - part picking time per chip [sec];
  - mounting time per chip [sec].

Finally, the EBC template for an electronic part mounter is described using XML as follows:

```
<?xml version="1.0" standalone="yes"?>
<ebc
    xmlns:opeartion="product-operation.xml"
```

```
    xmlns:m="https://www.w3.org/1998/Math/MathML">

  <header>
    <template_name>ELECTRONIC PART MOUNTER</template_name>
    <template_id>xxxx</template_id>
  </header>

  <property_set>
    <!-- default value setting -->
    <equipment_id>1</equipment_id>
    <equipment_type>XXX</equipment_type>
    <organization>XYZ Corporation</organization>
    <model_name>YYY</model_name>
    <external_size unit="cm">H=100,W=250,D=100</external_size>
    <equipment_weight unit="kg">200</equipment_weight>
    <startup_time unit="sec">20</startup_time>
    <shutdown_time unit="sec">20</shutdown_time>
    <electricpower_idling unit="kW">30</electricpower_idling>
    <electricpower_parts_chips_setting unit="kW">40</electricpower_parts_chips_setting>
    <electricpower_parts_chips_mounting_on_a_board unit="kW">60</electricpower_parts_chips_
mounting_on_a_board>
  </property_set>

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

  <behaviour>
    <!-- state list -->
    <state_list>
      <state id="s1" name="stop" state="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>&times;</m:mo>
                <m:mi>property_set:electricpower_idling</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
      </state>
      <state id="s3" name="parts_chips_setting">
        <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>&times;</m:mo>

<m:mi>property_set:electricpower_parts_chips_setting</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
          <m:math>
            <m:mrow>
              <m:mi>operation:pca_id_old</m:mi>
              <m:mo>=</m:mo>
              <m:mrow>
                <m:mi>operation:pca_id</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
      </state>
      <state id="s4" name="parts_chips_mounting_on_a_board">
        <state_data>
          <elapsed_time>0</elapsed_time>
          <total_energyconsumption>0</total_energyconsumption>
          <chips_quantity>0</chips_quantity>
        </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>&times;</m:mo>

<m:mi>property_set:electricpower_parts_chips_mounting_on_a_board</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
        <calculation_formula repeat="no">
          <m:math>
            <m:mrow>
              <m:mi>operation:total_processing_quantity</m:mi>
              <m:mo>=</m:mo>
              <m:mrow>
                <m:mi>operation:total_processing_quantity</m:mi>
                <m:mo>+</m:mo>
                <m:mn>1</m:mn>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
```

```
  <!-- substates -->
  <state_list>
    <!-- substate:nozzle_selection -->
    <state id="s4-1" name="nozzle_selection">
      <state_data>
        <elapsed_time>0</elapsed_time>
      </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>
      </calculation_formula>
    </state>
    <!-- substate:part_picking -->
    <state id="s4-2" name="part_picking">
      <state_data>
        <elapsed_time>0</elapsed_time>
      </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>
      </calculation_formula>
    </state>
    <!-- substate:mounting -->
    <state id="s4-3" name="mounting">
      <state_data>
        <elapsed_time>0</elapsed_time>
      </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>chips_quantity</m:mi>
            <m:mo>=</m:mo>
            <m:mrow>
              <m:mi>chips_quantity</m:mi>
              <m:mo>+</m:mo>
              <m:mn>1</m:mn>
            </m:mrow>
          </m:mrow>
        </m:math>
      </calculation_formula>
    </state>
  </state_list>
</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 operator="==">
      <param name="operation:power" />
      <param>on</param>
    </transition_condition>
    <transition_condition operator="==">
      <param name="property_set:startup_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s1"/>
    <to_state id="s2"/>
  </state_transition>

  <state_transition id="t2" name="power_off">
    <!-- CONDITION: power off -->
    <!-- CONDITION: shutdown time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:power" />
      <param>off</param>
    </transition_condition>
    <transition_condition operator="==">
      <param name="property_set:shutdown_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s1"/>
  </state_transition>

  <state_transition id="t3" name="pca_id_input">
    <!-- CONDITION: product input time reached -->
    <transition_condition operator="==">
      <param name="operation:product_time" />
      <param name="system_time" />
    </transition_condition>
    <!-- CONDITION: PCA ID is different from the previous production -->
    <transition_condition operator="!=">
      <param name="operation:pca_id" />
      <param name="operation:pca_id_old" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s3"/>
  </state_transition>

  <state_transition id="t4" name="mounted_parts/chips_ready">
    <!-- CONDITION: electrical-parts(chips) changing time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:chips_change_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s3"/>
    <to_state id="s4"/>
  </state_transition>

  <state_transition id="t5" name="new_board_in">
    <!-- CONDITION: PCB input time reached -->
    <transition_condition operator="==">
      <param name="opeartion:product_time" />
      <param name="system_time" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s4"/>
  </state_transition>


  <state_transition id="t6" name="completed_board_out">
    <!-- CONDITION: number of producing product is achieved  -->
```

```
        <!-- CONDITION: chip mounting time for one board is elapsed -->
        <transition_condition operator="==">
          <param name="operation:total_processing_quantity" />
          <param name="operation:plan_quantity" />
        </transition_condition>
        <from_state id="s5"/>
        <to_state id="s2"/>
      </state_transition>
      <state_transition id="t7" name="next_chips_mounting">
        <transition_condition operator="&gt;">
          <param name="operation:total_processing_quantity" />
          <param name="operation:plan_quantity" />
        </transition_condition>
        <from_state id="s4"/>
        <to_state id="s4-1"/>
      </state_transition>
      <state_transition id="t8" name="nozzle_selection_finish">
        <!-- CONDITION: nozzle selection time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:nozzle_selection_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s4-1"/>
        <to_state id="s4-2"/>
      </state_transition>
      <state_transition id="t9" name="part_picking_finish">
        <!-- CONDITION: part picking time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:part_picking_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s4-2"/>
        <to_state id="s4-3"/>
      </state_transition>
      <state_transition id="10" name="mouting_finish_new_chips">
        <!-- CONDITION: mouting time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:mouting_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <transition_condition operator="&gt;">
          <param name="chips_quantity" />
          <param name="opeartion:plan_chips_quantity" />
        </transition_condition>
        <from_state id="s4-3"/>
        <to_state id="s4-1"/>
      </state_transition>
      <state_transition id="11" name="mounting_finish">
        <!-- CONDITION: mouting time is elapsed -->
        <transition_condition operator="==">
          <param name="opeartion:mouting_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <transition_condition operator="==">
          <param name="chips_quantity" />
          <param name="opeartion:plan_chips_quantity" />
        </transition_condition>
        <from_state id="s4-3"/>
        <to_state id="s4"/>
      </state_transition>
    </state_transition_list>
  </behaviour>
</ebc>
```

## B.4 Building an EBC template of a reflow soldering oven

A reflow soldering oven is an equipment used for reflow soldering of surface mount electronic components to PCBs. The behaviour of a reflow soldering oven is modelled as an activity flow shown in Figure B.4. After power on and start-up heating, temperature controls and conveyer are started. When a PCB with mounted

electronic parts is input to the reflow oven, a PCB is moved into the reflow oven by conveyer for reflow soldering. Temperature controls and reflow soldering proceed in parallel. A reflow oven is constructed from board pre-heating zones, board soldering zones and board cooling zones. Each zone controls its temperature. Heating and cooling procedures of each zone are controlled followed by the temperature profiles. A reflow oven moves to the shutdown cooling state when the power is off.

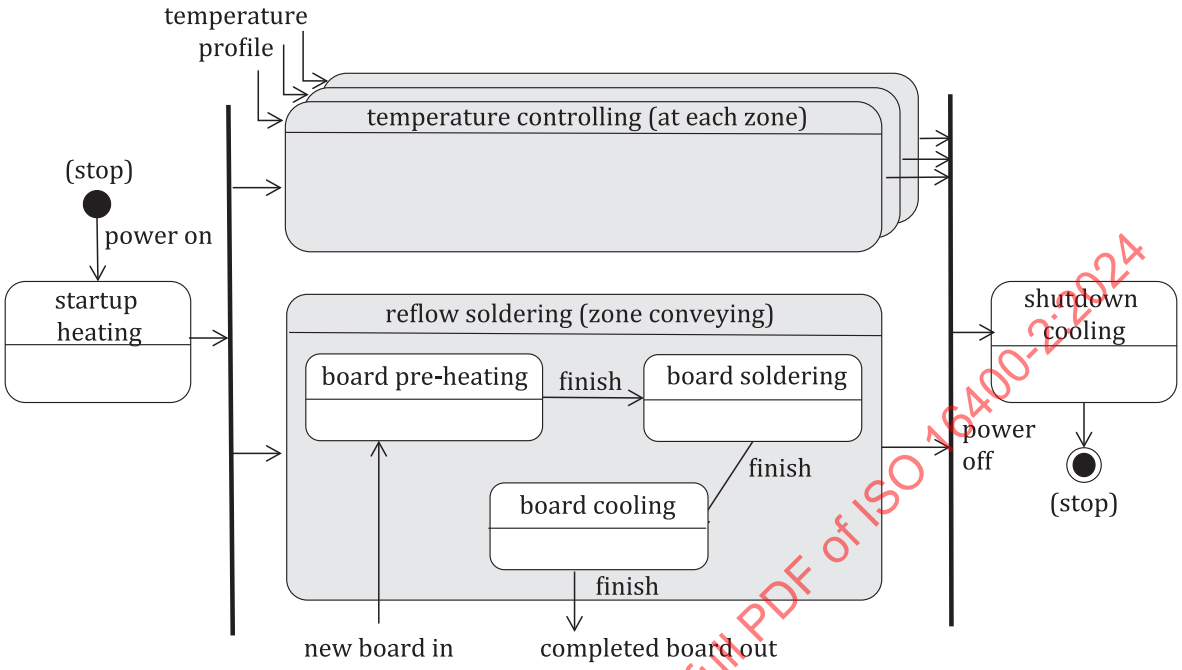


**Figure B.4 — Activity diagram for a reflow soldering oven**

The building procedure of an EBC template is applied to a reflow soldering oven as follows. In this example, the reflow soldering oven has three zones.

a) Listing of states

- stop;
- start-up heating;
- temperature control (for zones1 to 3);
- reflow soldering (zone conveying);

A PCB is carried through the following zones by conveyer:

- board pre-heating;
- board soldering;
- board cooling;

- shutdown cooling;

b) Listing of state transitions

- from stop into start-up heating (when power on);
- from start-up heating into temperature controls and reflow soldering (when start-up heating is completed);
- from temperature controls and reflow soldering into shutdown cooling (when power off);

— from shutdown cooling into stop (when shutdown cooling is completed);

c) Listing of static parameters (except common items, e.g. equipment ID, equipment type, model name, equipment weight)

— electric power of start-up heating [kW];

— electric power of shutdown cooling [kW];

— electric power of temperature control zone 1 [kW];

— electric power of temperature control zone 2 [kW];

— electric power of temperature control zone 3 [kW];

d) Listing of variables (providing by calculation formula)

— elapsed time of start-up heating [sec];

— elapsed time of shutdown cooling [sec];

— elapsed time of temperature control zone 1 [sec];

— elapsed time of temperature control zone 2 [sec];

— elapsed time of temperature control zone 3 [sec];

— total energy consumption of start-up heating [kWh];

— total energy consumption of shutdown cooling [kWh];

— total energy consumption of temperature control zone 1 [kWh];

— total energy consumption of temperature control zone 2 [kWh];

— total energy consumption of temperature control zone 3 [kWh];

e) Listing of variables which are input values when construction of virtual system or when virtual operation

— start-up heating time;

— shutdown cooling time.

Finally, the EBC template for a reflow soldering oven is described using XML as follows:

```
<?xml version="1.0" standalone="yes"?>
<ebc
    xmlns:opeartion="product-operation.xml"
    xmlns:m="https://www.w3.org/1998/Math/MathML">

  <header>
    <template_name>REFLOW SOLDERING OVEN</template_name>
    <template_id>xxxx</template_id>
  </header>

  <property_set>
    <!-- default value setting -->
    <equipment_id>1</equipment_id>
    <equipment_type>XXX</equipment_type>
    <organization>XYZ Corporation</organization>
    <model_name>YYY</model_name>
    <external_size unit="cm">H=100,W=250,D=100</external_size>
    <equipment_weight unit="kg">200</equipment_weight>
    <electricpower_startup_heating unit="kW">100</electricpower_startup_heating>
    <electricpower_shutdown_cooling unit="kW">40</electricpower_shutdown_cooling>
    <electricpower_temperature_controlling_zone1 unit="kW">70</electricpower_temperature_
controlling_zone1>
```

```
    <electricpower_temperature_controlling_zone2 unit="kW">80</electricpower_temperature_
controlling_zone2>
    <electricpower_temperature_controlling_zone3 unit="kW">60</electricpower_temperature_
controlling_zone3>
  </property_set>

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

  <behaviour>
    <!-- state list -->
    <state_list>
      <state id="s1" name="stop" state="on" />
      <state id="s2" name="startup_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>&times;</m:mo>
                <m:mi>property_set:electricpower_startup_heating</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
      </state>

      <state id="s3_1" name="temperature_controlling(zone1)">
        <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>&times;</m:mo>

<m:mi>property_set:electricpower_temperature_controlling_zone1</m:mi>
              </m:mrow>
            </m:mrow>
```

```
            </m:math>
          </calculation_formula>
        </state>

        <state id="s3_2" name="temperature_controlling(zone2)">
          <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>&times;</m:mo>

<m:mi>property_set:electricpower_temperature_controlling_zone2</m:mi>
                </m:mrow>
              </m:mrow>
            </m:math>
          </calculation_formula>
        </state>

        <state id="s3_3" name="temperature_controlling(zone3)">
          <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>&times;</m:mo>

<m:mi>property_set:electricpower_temperature_controlling_zone3</m:mi>
                </m:mrow>
              </m:mrow>
            </m:math>
          </calculation_formula>
        </state>

        <state id="s4" name="reflow_soldering(zone_converying)">
          <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>&times;</m:mo>
            <m:mi>property_set:electricpower_reflow_soldering</m:mi>
          </m:mrow>
        </m:mrow>
      </m:math>
    </calculation_formula>
  </state>

  <state id="s5" name="shutdown_cooling">
    <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>&times;</m:mo>
            <m:mi>property_set:electricpower_shutdown_cooling</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 -->
    <transition_condition operator="==">
      <param name="operation:power" />
      <param>on</param>
    </transition_condition>
    <from_state id="s1"/>
    <to_state id="s2"/>
  </state_transition>
  <state_transition id="t2" name="heating_completed">
    <!-- CONDITION: heating-up completed -->
```

```
        <transition_condition operator="==">
          <param name="operation:startup_heating_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s2"/>
        <to_state id="s3_1,s3_2,s3_3,s4" option="async"/>
      </state_transition>
      <state_transition id="t3" name="power_off">
        <!-- CONDITION: power off -->
        <transition_condition operator="==">
          <param name="operation:power" />
          <param>off</param>
        </transition_condition>
        <from_state id="s3_1,s3_2,s3_3,s4" option="async_wait"/>
        <to_state id="s5"/>
      </state_transition>
      <state_transition id="t4" name="cooling_completed">
        <!-- CONDITION: cooling-down completed -->
        <transition_condition operator="&lt;">
          <param name="operation:shutdown_cooling_time" />
          <param name="elapsed_time" />
        </transition_condition>
        <from_state id="s5"/>
        <to_state id="s1"/>
      </state_transition>
    </state_transition_list>
  </behaviour>
</ebc>
```

# Annex C
(informative)

# Example building of EBC templates for an injection molding line

## C.1 Equipment on an injection molding line

A structure of an injection molding line is shown in Figure C.1. An injection molding line consists of an injection molding machine including mold, a molding extraction robot and a mold temperature controller. A mold is mounted on an injection molding machine. A mold temperature controller and a molding extraction robot are connected with an injection molding machine. An injection molding machine controls starting and stopping the actions of a mold temperature controller and a molding extraction robot. Pellets, which are the raw material of the injection molded part, are put into an injection molding machine.

Pellets are put into an injection molding machine. A mold is set on an injection molding machine. After these preparations, the injection molding process cycle starts. An injection molding machine, a molding extraction robot and a mold temperature controller have complicated interaction in the injection molding process cycle.

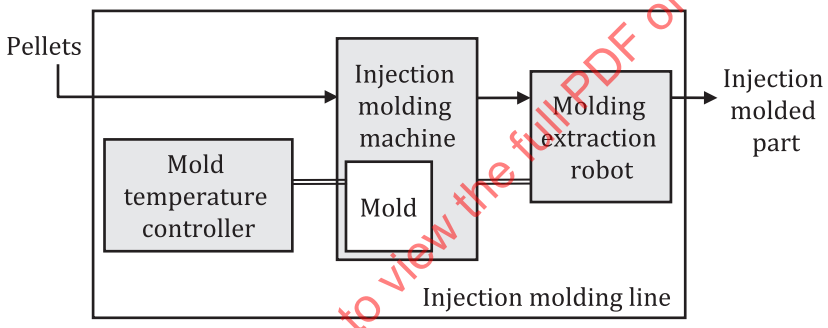


**Figure C.1 — Structure of an injection molding line**

According to the building procedure for an EBC template, templates for an injection molding machine, a molding extraction robot and a mold temperature controller are prepared.

In this example, EBC templates are simplified by omitting sub-activity descriptions and by focusing on electric energy consumption. Other focusing points which are related to the equipment's behaviour can be added in the same way.

## C.2 Building an EBC template of an injection molding machine

The behaviour of an injection molding machine is modelled as an activity flow shown in Figure C.2. When receiving an injection molding order in "idling" state, the "injection molding" state starts. Before this, an adequate mold is prepared and changed. The injection molding process cycle is repeated until completion of the ordered amount. There are four processes in the process cycle: injection molding, opening of a mold, ejection by ejector pins and restoration of the ejector pins. At the beginning of the injection molding process, the starting of the mold temperature control is requested. At the end of the injection molding process, the stopping of the temperature control is requested for cooling of the mold. After cooling the mold, it is opened and a request to the molding extraction robot is sent out. Then the molded part is ejected by the ejector pins. After receiving acknowledgement of the completed extraction from the robot, the ejector pins are restored.

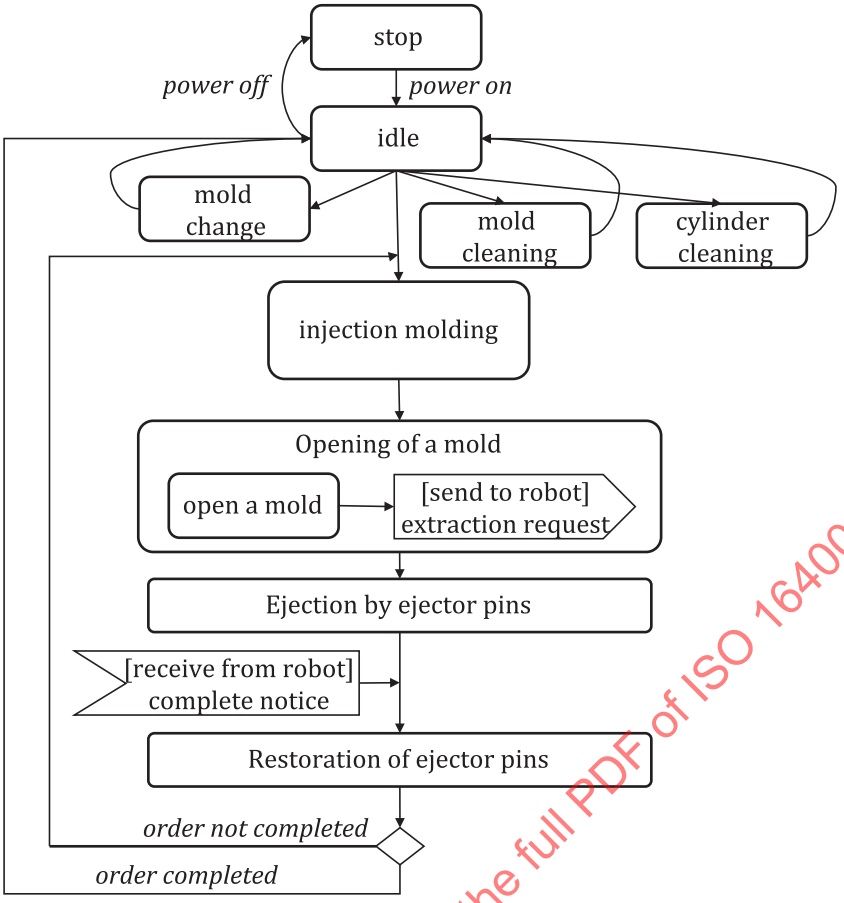


**Figure C.2 — Activity diagram for an injection molding machine**

The building procedure of the EBC template is applied to an injection molding machine as follows:

a) Listing of states

- stop;
- idling;
- mold change;
- mold cleaning;
- cylinder cleaning;
- injection molding;
- opening of a mold;
- ejection by ejector-pins;
- restoration of ejector-pins;

b) Listing of state transitions

- from stop into idling (when power on);
- from idling into stop (when power off);
- from idling into mold change (when mold ID is different from the previous production);
- from mold change into idling (when mold change completed);

- from idling into mold cleaning (when cleaning time is reached);
- from mold cleaning into idling (when mold cleaning completed);
- from idling into cylinder cleaning (when cleaning time is reached);
- from cylinder cleaning into idling (when cylinder cleaning completed);
- from idling into injection molding (when product input time reached);
- from injection molding into mold opening (when injection molding completed);
- from mold opening into ejection (when opening time completed);
- from ejection into restoration (when extraction-complete notice received);
- from restoration into injection molding (production is uncompleted);
- from restoration into idling (when number of producing product is achieved);

c) Listing of static parameters (except common items, e.g. equipment ID, equipment type, model name, equipment weight)

- starting-up time of equipment [sec];
- shutdown time of equipment [sec];
- electric power of idling [kW];
- electric power of mold change [kW];
- electric power of mold cleaning [kW];
- electric power of cylinder cleaning [kW];
- electric power of injection molding [kW];
- electric power of mold opening [kW];
- electric power of ejection [kW];
- electric power of restoration [kW];

d) Listing of variables (providing by calculation formula)

- elapsed time of idling [sec];
- elapsed time of mold change [sec];
- elapsed time of mold cleaning [sec];
- elapsed time of cylinder cleaning [sec];
- elapsed time of injection molding [sec];
- elapsed time of mold opening [sec];
- elapsed time of ejection [sec];
- elapsed time of restoration [sec];
- total energy consumption of idling [kWh];
- total energy consumption of mold change [kWh];
- total energy consumption of mold cleaning [kWh];

- total energy consumption of cylinder cleaning [kWh];
- total energy consumption of injection molding [kWh];
- total energy consumption of mold opening [kWh];
- total energy consumption of ejection [kWh];
- total energy consumption of restoration [kWh];
- total processing quantity;

e) Listing of variables which are input values when construction of virtual system or when virtual operation

- mold ID;
- previous used mold ID;
- planned production quantity;
- production starting time;
- mold cleaning instruction;
- cylinder cleaning instruction;
- mold changing time;
- mold cleaning time;
- cylinder cleaning time;
- injection molding time;
- mold open time;
- ejector-pins ejection time;
- ejector-pins restoration time;

f) Listing of relationships with other equipment

- extraction request (from injection molding machine to molding extraction robot);
- extraction completion notice (from molding extraction robot to injection molding machine).

Finally, the EBC template for an injection molding machine is described using XML as follows:

```
<?xml version="1.0" standalone="yes"?>
<ebc
    xmlns:operation="product-operation.xml"
    xmlns:m="https://www.w3.org/1998/Math/MathML">

  <header>
    <template_name>INJECTION MOLDING MACHINE</template_name>
    <template_id>xxxx</template_id>
  </header>

  <property_set>
    <!-- default value setting -->
    <equipment_id>1</equipment_id>
    <equipment_type>XXX</equipment_type>
    <organization>XYZ Corporation</organization>
    <model_name>YYY</model_name>
    <external_size unit="cm">H=100,W=250,D=100</external_size>
    <equipment_weight unit="kg">200</equipment_weight>
    <startup_time unit="sec">20</startup_time>
```

```
    <shutdown_time unit="sec">20</shutdown_time>
    <electricpower_idling unit="kW">30</electricpower_idling>
    <electricpower_mold_change unit="kW">30</electricpower_mold_change>
    <electricpower_mold_cleanig unit="kW">30</electricpower_mold_cleanig>
    <electricpower_cylinder_cleaning unit="kW">30</electricpower_cylinder_cleaning>
    <electricpower_injection_molding unit="kW">30</electricpower_injection_molding>
    <electricpower_opening_of_a_mold unit="kW">30</electricpower_opening_of_a_mold>
    <electricpower_ejection_by_ejector_pins unit="kW">30</electricpower_ejection_by_ejector_
pins>
    <electricpower_restorationof_ejector_pins unit="kW">30</electricpower_restorationof_
ejector_pins>
  </property_set>

  <external_interaction_list>
    <external_interaction id="e1" name="robot_extraction">
      <send_to equipment_id="" />
      <receive_from equipment_id="" />
    </external_interaction>
    <external_interaction id="e2" name="robot_completed_notice">
      <send_to equipment_id="" />
      <receive_from equipment_id="" />
    </external_interaction>
  </external_interaction_list>

  <behaviour>
    <!-- state list -->
    <state_list>
      <state id="s1" name="stop" state="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>&times;</m:mo>
                <m:mi>property_set: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>&times;</m:mo>
            <m:mi>property_set:electricpower_mold_change</m:mi>
          </m:mrow>
        </m:mrow>
      </m:math>
      <m:math>
        <m:mrow>
          <m:mi>operation:pca_id_old</m:mi>
          <m:mo>=</m:mo>
          <m:mrow>
            <m:mi>operation: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>&times;</m:mo>
            <m:mi>property_set: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>&times;</m:mo>
            <m:mi>property_set: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>&times;</m:mo>
            <m:mi>property_set: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>&times;</m:mo>
            <m:mi>property_set: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>&times;</m:mo>

<m:mi>property_set: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>&times;</m:mo>

<m:mi>property_set:electricpower_restorationof_ejector_pins</m:mi>
              </m:mrow>
            </m:mrow>
          </m:math>
        </calculation_formula>
        <calculation_formula repeat="no">
          <m:math>
            <m:mrow>
              <m:mi>operation:total_processing_quantity</m:mi>
              <m:mo>=</m:mo>
              <m:mrow>
                <m:mi>operation: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="energy_on">
    <!-- CONDITION: energy on -->
    <!-- CONDITION: start-up time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:energy" />
      <param>on</param>
    </transition_condition>
    <transition_condition operator="==">
      <param name="property_set:startup_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s1"/>
    <to_state id="s2"/>
  </state_transition>
  <state_transition id="t2" name="power_off">
    <!-- CONDITION: power off -->
    <!-- CONDITION: shutdown time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:power" />
      <param>off</param>
    </transition_condition>
    <transition_condition operator="==">
      <param name="property_set:shutdown_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s1"/>
  </state_transition>
  <state_transition id="t3" name="mold_change">
    <!-- CONDITION: product input time reached -->
    <transition_condition operator="==">
      <param name="opeartion:product_time" />
      <param name="system_time" />
    </transition_condition>
    <!-- CONDITION: the mold ID is different from the previous production -->
    <transition_condition operator="!=">
      <param name="operation:mold_id" />
      <param name="operation:mold_id_old" />
    </transition_condition>
    <from_state id="s2"/>
    <to_state id="s3"/>
  </state_transition>
  <state_transition id="t4" name="mold_change_completed">
    <!-- CONDITION: mold change time is elapsed -->
    <transition_condition operator="==">
      <param name="operation:mold_change_time" />
      <param name="elapsed_time" />
    </transition_condition>
    <from_state id="s3"/>
    <to_state id="s2"/>
  </state_transition>
  <state_transition id="t5" name="mold_cleaning">
    <!-- CONDITION: product input time reached -->
    <transition_condition operator="==">
      <param name="operation:product_time" />
      <param name="system_time" />
    </transition_condition>
    <!-- CONDITION: mold cleaning instruction -->
    <transition_condition operation="==">
      <param name="operation:mold_cleaning_instruction" />
      <param>true</param>
```