Reference number of working document: **ISO/TC 184/SC 1 N 000**

Version description: **V19 - February 2002**

Date: 2002-02-06

Reference number of document: **ISO/FDIS 14649-11**

Committee identification: ISO/TC 184/SC 1/WG 7

Secretariat: DIN

Industrial automation systems and integration

Physical device control

ISO 14649 Data model for Computerized Numerical Controllers

Part 11: PROCESS DATA FOR MILLING

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: Draft International standard
Document subtype:
Document stage: (20) Preparation
Document language: E

Information on technical or structural contents are available at the following addresses:

Convener: Friedrich Glantschnig
AMT Consulting
Address: Höhenweg 33a
CH-5417 Untersiggenthal
Telephone: +41 (056) 288 2042
FAX; +41 (056) 288 3942
E-mail: fglantschnig@swissonline.ch

Owners of this part of documents,

	Yong Tak Hyun/Jochen Wolf	Dr. Chiaki Sakomoto
	WZL, RWTH-Aachen	Komatsu Engineering Co. LZD
Adress:	Werkzeugmaschinenlabor	3-20-1 Nakase
	RWTH	Kawasaki-Ward
	D-52074 Aachen	J-210-0818 Kawasaki City
Telephone:	+49 (0241) 80-27458	+89 (44) 288 8782
FAX:	+49 (0241) 80-22293	+89 (44) 288 8777

E-mail: hyt@wzl.rwth-aachen.de
chiaki_sakamoto@KEG.komatsu.co.jp

Contents

Foreword iv
Introduction v
1 Scope 1
2 Normative references 1
3 Terms and definitions 1

3.1 Finishing 1

3.2 Roughing 2
4 Process data for milling 2

4.1 Header and references 2

4.2 Technology-specific machining operations 2
4.2.1 NC functions for milling 2
4.2.2 Tool direction for milling 3
4.2.3 Milling machining operation 4
4.2.4 Milling technology 5
4.2.5 Milling machine functions 6
4.2.6 Milling type operation 7
4.2.7 Freeform operation 13
4.2.8 Two5D milling operation 17
4.2.9 Plane milling 22
4.2.10 Side milling 23
4.2.11 Bottom and side milling 23
4.2.12 Drilling type operation 24
4.2.13 Drilling operation 26
4.2.14 Boring operation 27
4.2.15 Back boring 28
4.2.16 Tapping 29
4.2.17 Thread drilling 29
5 Conformance requirements 29

5.1 Conformance class 1 entities 30

5.2 Conformance class 2 entities 31
Annex A **(normative)** EXPRESS listing 34
Annex B **(normative)** Short names of entities 47
Annex C **(normative)** Implementation method specific requirements 51
Annex D **(informative)** EXPRESS-G diagram 52
Annex E **(informative)** Computer-interpretable listings 59
Annex F **(informative)** Sample NC programmes 60
Index 77

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 14649 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 14649-11 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 1, *Physical device control*.

ISO 14649 consists of the following parts, under the general title

Data model for Computerized Numerical Controllers

— Part 1 : *Overview and fundamental principles, published as actual DIS Phase 1*

— Part 2: *Language bindings, Fundamentals, will be published as Phase 3*

— Part 3: *Language binding in Java, will be published as Phase 3*

— Part 9: *Glossary, will be published as Phase 3*

— Part 10: *General Process Data, published as actual DIS Phase 1*

— Part 11: *Process Data for Milling, published as actual DIS Phase 1*

— Part 12: *Process Data for Turning, will be published as Phase 3*

— Part 13: *Process Data for EDM, will be published as Phase 3*

— Part 50: *AIM of General Process Data, will be published as Phase 2*

— Part 51: *AIM of Process Data for Milling, will be published as Phase 2*

— Part 52: *AIM of Process Data for Turning, will be published as Phase 3*

— Part 53: *AIM of Process Data for EDM, will be published as Phase 3*

— Part 111: *Tools for Milling, published as actual DIS Phase 1*

Introduction

Modern manufacturing enterprises are built from facilities spread around the globe, which contain equipment from hundreds of different manufacturers. Immense volumes of product information must be transferred between the various facilities and machines. Today's digital communications standards have solved the problem of reliably transferring information across global networks. For mechanical parts, the description of product data has been standardized by ISO 10303. This leads to the possibility of using standard data throughout the entire process chain in the manufacturing enterprise. Impediments to realizing this principle are the data formats used at the machine level. Most computer numerical control (CNC) machines are programmed in the ISO 6983 "G and M code" language. Programs are typically generated by computer-aided manufacturing (CAM) systems that use computer-aided design (CAD) information. However, ISO 6983 limits program portability for three reasons. First, the language focuses on programming the tool center path with respect to machine axes, rather than the machining process with respect to the part. Second, the standard defines the syntax of program statements, but in most cases leaves the semantics ambiguous. Third, vendors usually supplement the language with extensions that are not covered in the limited scope of ISO 6983.
ISO 14649 is a new model of data transfer between CAD/CAM systems and CNC machines, which replaces ISO 6983. It remedies the shortcomings of ISO 6983 by specifying machining processes rather than machine tool motion, using the object-oriented concept of Workingsteps. Workingsteps correspond to high-level machining features and associated process parameters. CNCs are responsible for translating Workingsteps to axis motion and tool operation. A major benefit of ISO 14649 is its use of existing data models from ISO 10303. As ISO 14649 provides a comprehensive model of the manufacturing process, it can also be used as the basis for a bi- and multi-directional data exchange between all other information technology systems.
ISO 14649 represents an object oriented, information and context preserving approach for NC-programming, that supersedes data reduction to simple switching instructions or linear and circular movements. As it is object- and feature oriented and describes the machining operations executed on the workpiece, and not machine dependent axis motions, it will be running on different machine tools or controllers. This compatibility will spare all data adaptations by postprocessors, if the new data model is correctly implemented on the NC-controllers. If old NC programs in ISO 6983 are to be used on such controllers, the corresponding interpreters shall be able to process the different NC program types in parallel.
ISO TC184/SC1/WG7 envisions a gradual evolution from ISO 6983 programming to portable feature-based programming. Early adopters of ISO 14649 will certainly support data input of legacy "G and M codes" manually or through programs, just as modern controllers support both command-line interfaces and graphical user interfaces. This will likely be made easier as open-architecture controllers become more prevalent. Therefore, ISO 14649 does not include legacy program statements, which would otherwise dilute the effectiveness of the standard.

Industrial automation systems and integration — Physical device control — Data model for Computerized Numerical Controllers — Part 11: Process data for milling

Scope

This part of ISO 14649 specifies the technology-specific data elements needed as process data for milling. Together with the general process data described in ISO 14649-10, it describe the interface between a computerized numerical controller and the programming system (i.e. CAM system or shopfloor programming system) for milling . It can be used for milling operations on all types of machines, be it milling machines, machining centers, or lathes with motorized tools capable of milling. The scope of this part does not include any other technologies, like turning, grinding, or EDM. These technologies will be described in further parts of ISO 14649.
Subject of the milling_schema, which is described in this part of ISO 14649, is the definition of technology-specific data types representing the machining process for milling and drilling. This includes both milling of freeform surfaces as well as milling of prismatic workpieces (also known as 2½D-milling). Not included in this schema are geometric items, representations, manufacturing features, executable objects, and base classes which are common for all technologies. They are referenced from ISO 10303's generic resources and ISO 14649-10. The description of process data is done using the EXPRESS language as defined in ISO 10303-11. The encoding of the data is done using ISO 10303-21.

Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 14649. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 14649 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.
ISO 10303 Part11, Industrial automation systems and integration - Product data and exchange - Description methods: the EXPRESS Language Reference Manual.

ISO 10303 Part21, Industrial automation systems and integration - Product data and exchange - Implementation methods: Clear text encoding of exchange structure.

Terms and definitions

For the purposes of this part of ISO 14649, the terms and definitions given in ISO 14649-10 and the followings apply.

Finishing

A milling operation used to cut a part. The finishing operation usually follows a roughing operation. The goal of finishing is to reach the surface quality required, cf. roughing.

Roughing

A milling operation used to cut a part. While the aim of roughing is to remove large quantities of material in a short time, the surface quality is usually not important. The roughing operation is usually followed by a finishing operation, cf. finishing.

Process data for milling

5.1 Header and references

The following listing gives the header and the list of entities which are referenced within this schema.

```
SCHEMA milling_schema;
(* Version 19 date: 2002-02-06
 * Author: ISO TC184/SC1/WG7
 *)

(* ********************************************************** *)
(* Types from machining_schema                  ISO 14649-10   *)
(* ********************************************************** *)

REFERENCE FROM machining_schema(
        bounded_curve,
        cartesian_point,
        direction,
        identifier,
        label,
        length_measure,
        nc_function,
        machine_functions,
        machining_operation,
        machining_tool,
        material,
        plane_angle_measure,
        positive_ratio_measure,
        pressure_measure,
        property_parameter,
        rot_direction,
        rot_speed_measure,
        speed_measure,
        technology,
        time_measure,
        toolpath_list,
        tool_direction);
```

5.2 Technology-specific machining operations

NC functions for milling

The NC functions specific to milling technologies are described in the following subs clauses. These are subtypes of entity nc_function defined in ISO 14649-10.

Exchange pallet

This function is used to execute a pallet exchange.

```
ENTITY exchange_pallet
  SUBTYPE OF (nc_function);
END_ENTITY;
```

Index pallet

This function is used to place the pallet to the indicated position by the parameter index.

```
ENTITY index_pallet
  SUBTYPE OF (nc_function);
  its_index:           INTEGER;
END_ENTITY;
```

its_index: The parameter index value by which the destined position of the pallet is indicated.

Index table

This function is used to place the rotation table to the indicated position by the parameter index.

```
ENTITY index_table
  SUBTYPE OF (nc_function);
  its_index:           INTEGER;
END_ENTITY;
```

its_index: The parameter index value by which the destined position of the rotation table is indicated.

Load tool

This function is used to load a tool that can be seleted independent from the geometrical information.

```
ENTITY load_tool
  SUBTYPE OF (nc_function);
  its_tool:            machining_tool;
END_ENTITY;
```

its_tool: The tool which has to be loaded.

Unload tool

This function is used to unload a tool.

```
ENTITY unload_tool
  SUBTYPE OF (nc_function);
  its_tool:            OPTIONAL machining_tool;
END_ENTITY;
```

its_tool: The tool which has to be exchanged. In case of an operation where more than one tool is in use at the same time this attribute has to be set.

Tool direction for milling

This is the base class of all tool orientations used for freeform machining. It is subtypes of entity tool_direction defined in ISO 14649-10.

```
ENTITY tool_direction_for_milling
  ABSTRACT SUPERTYPE OF (ONEOF(three_axes_tilted_tool, five_axes_var_tilt_yaw,
  five_axes_const_tilt_yaw))
  SUBTYPE OF (tool_direction);
END_ENTITY;
```

Three axes tilted tool

In this mode of operation, the tool is tilted, so the tool direction is not parallel to any of the three machine axes. However, the tool is clamped to fix the tool angle and motion is still only in the three linear axes. Unlike five_axes_var_tilt_yaw the tilt and/or yaw angles are not variable.

```
ENTITY three_axes_tilted_tool
  SUBTYPE OF (tool_direction_for_milling);
  its_tool_direction: direction;
END_ENTITY;
```

its_tool_direction: The direction of the tool in absolute machine co-ordinates.

Five axes with variable tilt and yaw angles

Simultaneous tool movements in five axes are used for machining. During motion, the tool direction is adjusted so as to follow the curve given in the toolpath instances.

```
ENTITY five_axes_var_tilt_yaw
  SUBTYPE OF (tool_direction_for_milling);
END_ENTITY;
```

Five axes with constant tilt and yaw angles

This is a special case of five_axes_var_tilt_yaw. The tool is moved so that the tilt and yaw angles are constant in each point of the toolpath, relative to the co-ordinate system given by the surface normal in the cutter contact point and the tangent in feed direction. Tilt and yaw are given as attributes of this entity. Note that these values may be overridden if an explicit tool direction curve is specified for a toolpath.

```
ENTITY five_axes_const_tilt_yaw
  SUBTYPE OF (tool_direction_for_milling);
  tilt_angle : plane_angle_measure;
  yaw_angle  : plane_angle_measure;
END_ENTITY;
```

tilt_angle: The inclination of the tool in feed direction, measured against the surface normal in the cutter contact point.

yaw_angle: The rotation of the inclined tool around the surface normal, measured against the surface tangent in feed direction in the cutter contact point.

Milling machining operation

This is the base class of all operations described in this part of ISO 14649. It is a subtype of entity machining_operation defined in ISO 14649-10. In case that feedrate_per_tooth of its_technology is chosen, number_of_teeth of its_tool_body of its_tool should be given.

```
ENTITY milling_machining_operation
   ABSTRACT SUPERTYPE OF (ONEOF(milling_type_operation, drilling_type_operation))
   SUBTYPE OF (machining_operation);
   overcut_length: OPTIONAL length_measure;
WHERE
WR1: (EXISTS(SELF.its_technology.feedrate_per_tooth) AND
     EXISTS(SELF.its_tool.its_tool_body.number_of_teeth))
     OR(NOT(EXISTS(SELF.its_technology.feedrate_per_tooth)));
END_ENTITY;
```

overcut_length: The overcut on the open side(s) of the feature. It is not allowed for manufacturing of features which are bounded by material on all sides, i. e. pockets. In case of round_hole, this attribute is allowed only for through-bottom holes. If the cutting_depth of drilling_type_operation specifies a conflicting value, overcut_ length is ignored.

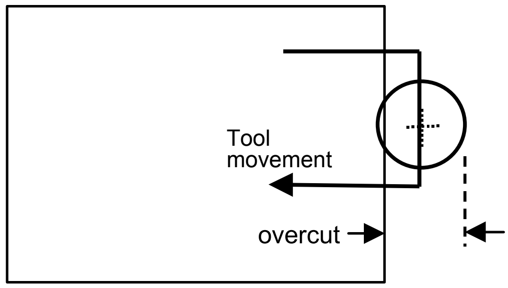


Fig. 1: Overcut

Milling technology

This entity defines the technological parameters of the milling operation. It is a subtype of entity technology defined in ISO 14649-10. Of the four alternatives for specifying speeds, exactly two must be given as indicated by the WHERE rules. If the attribute adaptive_control s invoked, some or all of these values may be ignored.

```
ENTITY milling_technology
  SUBTYPE OF (technology);
  cutspeed:                       OPTIONAL speed_measure;
  spindle:                        OPTIONAL rot_speed_measure;
  feedrate_per_tooth:             OPTIONAL length_measure;
  synchronize_spindle_with_feed: BOOLEAN;
  inhibit_feedrate_override:      BOOLEAN;
  inhibit_spindle_override:       BOOLEAN;
  its_adaptive_control:           OPTIONAL adaptive_control;
  WHERE
  WR1: (EXISTS(cutspeed) AND NOT EXISTS(spindle))
    OR (EXISTS(spindle) AND NOT EXISTS(cutspeed))
    OR (EXISTS(its_adaptive_control));
  WR2: (EXISTS(SELF.feedrate) AND NOT EXISTS(feedrate_per_tooth))
    OR (EXISTS(feedrate_per_tooth) AND NOT EXISTS(SELF.feedrate))
    OR (EXISTS(its_adaptive_control));
END_ENTITY;
```

cutspeed: Cutting speed of the tool, the speed of spindle converted into a linear speed.

spindle: Rotational speed of the tool. As defined for rot_speed_measure, positive values indicate tool rotation in mathematical positive direction of the c axis, i. e. counter-clockwise motion if looking from the tool holder to the workpiece. Note that usual cutting tools require clockwise motion so the value of this attribute will typically be negative.

feedrate_per_tooth: Feed of the tool expressed as a distance.

synchronize_spindle_with_feed:

If true, cutting speed and feed of the tool is synchronised. Therefore, the pitch of tap can be kept constant at the bottom of a hole when cutting speed is being decelerated and accelerated.

inhibit_feedrate_override: If true, the feedrate override through the operating panel or by adaptive control systems is not allowed.

inhibit_spindle_override: If true, the spindle speed override through the operating panel or by adaptive control systems is not allowed.

its_adaptive_control: Any kind of vendor specific adaptive control strategy.

Adaptive control

This entity defines the vendor-specific adaptive control strategy. At a later time, the specific nature of the adaptive control algorithm and further parameters can be specified in appropriate subtypes.

```
ENTITY adaptive_control;
END_ENTITY;
```

Milling machine functions

The entity describes the state of various functions of the machine, like coolant, chip removal, etc. to be applied during the time span of an operation. It is a subtype of entity machine_functions defined in ISO 14649-10.

```
ENTITY milling_machine_functions
  SUBTYPE OF (machine_functions);
  coolant              :      BOOLEAN;
  coolant_pressure     :      OPTIONAL pressure_measure;
  mist                 :      OPTIONAL BOOLEAN;
  through_spindle_coolant:    BOOLEAN;
  through_pressure:           OPTIONAL pressure_measure;
  axis_clamping        :      LIST [0:?] OF identifier;
  chip_removal         :      BOOLEAN;
  oriented_spindle_stop:      OPTIONAL direction;
  its_process_model:          OPTIONAL process_model_list;
  other_functions      :      SET [0:?] OF property_parameter;
END_ENTITY;
```

coolant: If true, the coolant is activated.

coolant_pressure: Optional specification of the pressure of the coolant system. Only valid if coolant is true.

mist: If true, activate mist coolant. Default is false. Only valid if coolant is true.

through_spindle_coolant: If true, activate coolant through the spindle. Default is false.

through_pressure: Pressure of coolant through the spindle. Only valid if through_spindle_coolant is true.

axis_clamping: Describes which axes are to be clamped, e.g. X,Y,A. Note that this information is machine dependent and should be avoided.

chip_removal: If true, activate chip removal.

oriented_spindle_stop: If specified, the spindle will stop in the given direction relative to the machine zero position of C-axis in case a spindle stop occurs during or at the end of the workingstep.

its_process_model: Optional information for process control.

other_functions: Optional list of other functions of generic type.

Process model list

For each workingstep, one or more process models may be started. These are modules for process control like chatter avoidance, thermal compensation, etc.

```
ENTITY process_model_list;
  its_list: LIST [1:?] OF process_model;
END_ENTITY;
```

its_list: List of process models for the current workingstep

Process model

Special machine-specific functions to make the machining process more secure and accurate. (e.g. chatter avoidance, thermal compensation, ...)

```
ENTITY process_model;
  ini_data_file:  label;
  its_type:       label;
END_ENTITY;
```

ini_data_file: A filename including path of the file containing the initialisation data of the process model.

its_type: The type of process model (e.g. chatter avoidance, thermal compensation, ...)

Milling type operation

This is the base class of all operations for milling. It includes all necessary attributes to describe technology and strategy. It is a subtype of entity milling_machining_operation.
In general, there are two types of machining operations: roughing and finishing. The roughing is to remove all material from the original raw piece surface down to the bottom or side of the feature minus the finishing allowance in multiple passes. The finishing will then remove the finish allowance to yield the final surface of the feature. In case of pre-cast features, e.g. pre-cast holes and pockets, roughing operation need to be one pass. This special condition is considered in the 2½D milling strategy with the attribute allow_multiple_passes.

```
ENTITY milling_type_operation
  ABSTRACT SUPERTYPE OF (ONEOF(freeform_operation, two5D_milling_operation))
  SUBTYPE OF (milling_machining_operation);
  approach:         OPTIONAL approach_retract_strategy;
  retract:          OPTIONAL approach_retract_strategy;
END_ENTITY;
```

approach: Optional information about approach (plunge) strategy to reach the first cut. If multiple layers are cut, as specified by allow_multiple_passes, this strategy will also be used to move from one layer to the start point of the next layer.

By default, the NC controller decides about the approach strategy. It may decide not to use any approach movement at all if the start point of cutting coincides with the end point of cutting for the preceding operation. If its_toolpath is given, this attribute will be ignored.

retract: Optional information about retract strategy after finishing the last cut. By default, the NC controller decides about the retract strategy. It may decide not to use any retract movement at all if the end point of cutting coincides with the start point of cutting for the next operation. If its_toolpath is given, this attribute will be ignored.

Approach retract strategy

Base class for the approach (plunge) and retract strategy. All approach and retract strategies are defined relative to the start or end point of the cutting operation, whether this is explicitly given in the operation of determined by the NC controller. The resulting start point of the approach or end point of the retract movement are defined to be the start and end point of the current operation. The feed rate on the approach or retract path is the feed rate specified for the related start or end point, respectively, of cutting.

```
ENTITY approach_retract_strategy
  ABSTRACT SUPERTYPE OF (ONEOF (plunge_strategy, air_strategy, along_path));
  tool_orientation: OPTIONAL direction;
END_ENTITY;
```

tool_orientation:	Only for machines with five-axis positioning capabilities. This specified the tool orientation at the beginning or end, respectively, of the approach or retract movement.

Plunge strategy

This is the base class for all approach movements which include cutting of material. This is typically the case for pocketing operations where the approach to the depth of the first cutting layer or between cutting layers requires the removal of material in order to create the approach path.

All plunge movements are guaranteed to occur within the boundaries of the underlying feature. All plunge movements will start at the retract plane valid for the current operation. They will end in the start point of the cutting operation, with the tangent of its approach path coinciding with the tangent of the ensuing cutting motion.

```
ENTITY plunge_strategy
  ABSTRACT SUPERTYPE OF (ONEOF (plunge_toolaxis, plunge_ramp, plunge_helix,
    plunge_zigzag))
  SUBTYPE OF (approach_retract_strategy);
END_ENTITY;
```

Plunge tool axis

Plunge in the direction of the tool axis.

Note: If the milling tool itself is unable to cut it’s way into the layer, a plunge drilling operation with a separate tool is required. As each operation can have only one tool, this will require the definition of a preceding drilling_type_operation. In this case, no plunge strategy should be given for the milling_type_operation, and the cut_start_point of both the milling_type_operation and the drilling_type_operation must coincide.

```
ENTITY plunge_toolaxis
  SUBTYPE OF (plunge_strategy);
END_ENTITY;
```

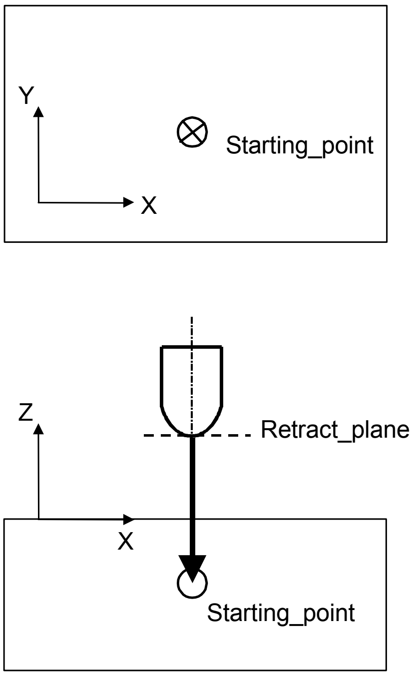


Fig. 2: Plunge tool axis

Plunge ramp

Plunge on a linear path which forms an angle with the feature surface.

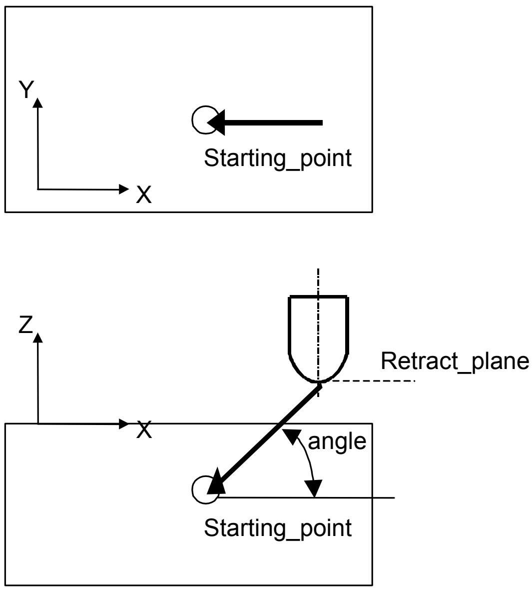


Fig. 3: Plunge ramp

```
ENTITY plunge_ramp
  SUBTYPE OF (plunge_strategy);
  angle: plane_angle_measure;
END_ENTITY;
```


angle: The angle of the ramp movement versus the surface in the end point of the approach. Note: start and end point can be calculated from the restrictions in Section 0.

Plunge helix

Plunge movement forming a helix. The path is defined by specifying the radius and grade of the helix. A circular movement can be specified by setting grade to zero.

```
ENTITY plunge_helix
  SUBTYPE OF (plunge_strategy);
  radius  :   length_measure;
  angle   :   plane_angle_measure;
END_ENTITY;
```

radius: Radius of the helical movement.

angle: The angle of the helical movement versus the surface in the end point of the approach. Note: start and end point can be calculated from the restrictions in Section 0.

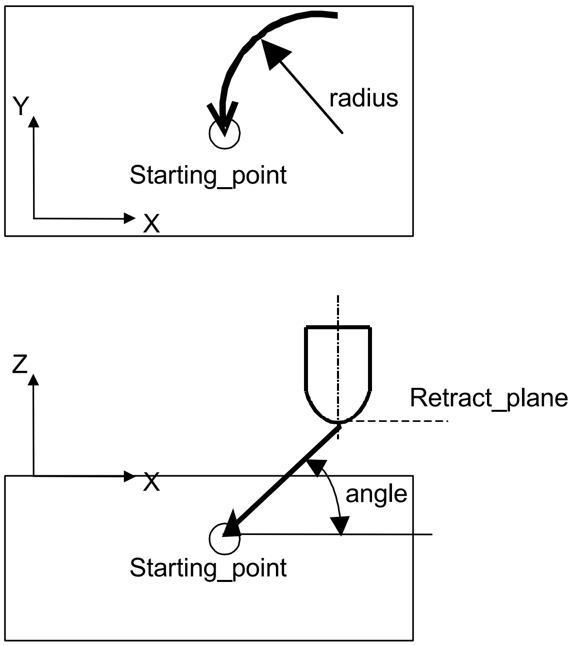


Fig. 4: Plunge helix

Plunge zigzag

Plunge movement using a zigzag motion. This is similar to the ramp-type movement, except the cutter changes direction if it touches a feature boundary or if the path length would exceed the specified width of the zigzag pattern.

```
ENTITY plunge_zigzag
  SUBTYPE OF (plunge_strategy);
  angle: plane_angle_measure;
  width: length_measure;
END_ENTITY;
```

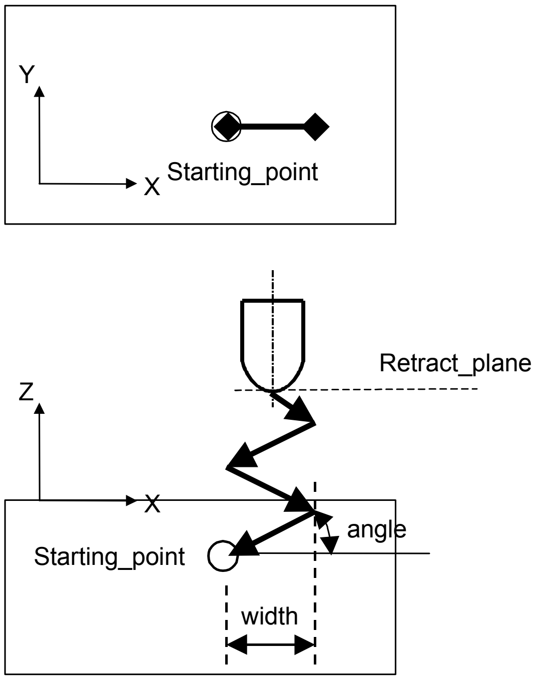


Fig. 5: Plunge zigzag

angle:	The angle of the movement versus the surface in the end point of the approach. Note: start and end point can be calculated from the restrictions in Section 0.
width:	The with of the zigzag path perpendicular to the direction of the descent.

Air strategy

This is the base class for all approach or retract movements through the air.
Unlike the plunge_strategy types these movements are not limited to the inside of the feature. All of these movements shall take place in a plane which is defined by the normal of the machined feature and the tangent of the cutting path in the start or end point, respectively, of the related cutting movement. If the start or end point lies at the intersection of two planes, as may be the case for bottom_and_side_milling operations, the surface normal is deemed to be the intermediate direction between the two normals.
Note that for side milling operations, e. g. for the milling of a contour, the resulting movements will be in the xy plane of the machine co-ordinate system.

```
ENTITY air_strategy
  ABSTRACT SUPERTYPE OF (ONEOF (ap_retract_angle, ap_retract_tangent))
  SUBTYPE OF (approach_retract_strategy);
END_ENTITY;
```

Approach retract angle

The movement is heading towards the start or from the end point in an angle to the surface. For plane milling, this may typically be an angle of 0 degrees in order to move straight from outside the workpiece into the material.

```
ENTITY ap_retract_angle
  SUBTYPE OF (air_strategy);
  angle:         plane_angle_measure;
  travel_length: length_measure;
END_ENTITY;
```

angle: Approach or lift angle versus the surface in the end point of the approach or the start point of the lift, respectively.

travel_length: The length of the angular approach. After travel_length has been reached, the tool will proceed to the retract plane using the shortest connection and vice versa.

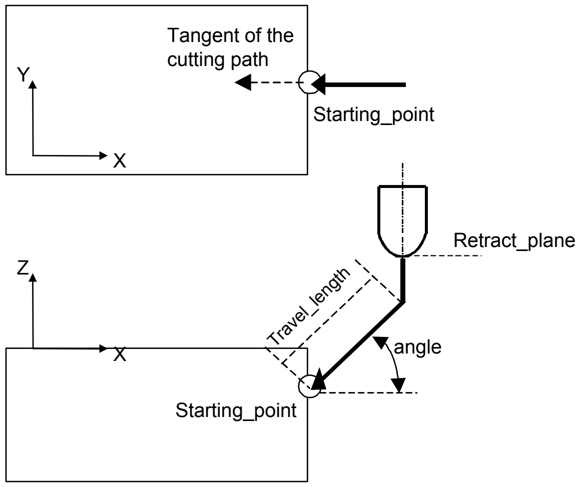


Fig. 6: Approach retract angle

Approach retract tangent

The movement is heading towards the start or from the end point in a curve. The motion start or ends in the retract plane valid for the current operation. If the specified radius for this motion is smaller than the distance to the retract plane as specified in the attribute retract_plane of the current operation, the remaining path will be executed in linear motion perpendicular to the retract plane.

```
ENTITY ap_retract_tangent
  SUBTYPE OF (air_strategy);
  radius: length_measure;
END_ENTITY;
```

radius: The radius of the approach or retract movement.

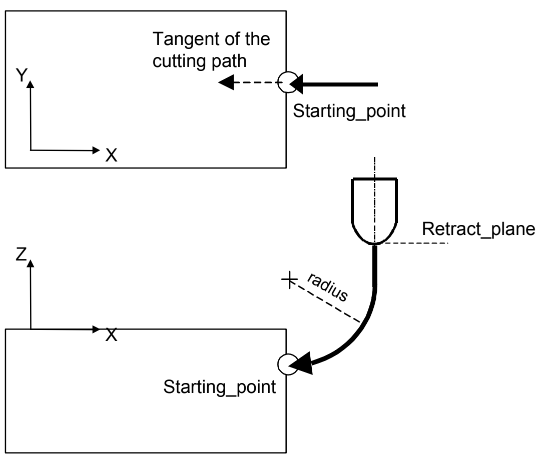


Fig. 7: Approach retract tangent

Along path

Approach or lift movement on a general path. This should be used if full control of the tool orientation during approach is required or for other special purposes.

```
ENTITY along_path
  SUBTYPE OF (approach_retract_strategy);
  path: toolpath_list;
END_ENTITY;
```

path: Specification of a general path for approach or lift movement. Note that the path is specified in a special co-ordinate system. The origin is the start or end point of the cutting operation, the axes are oriented like the local co-ordinate system of the feature.

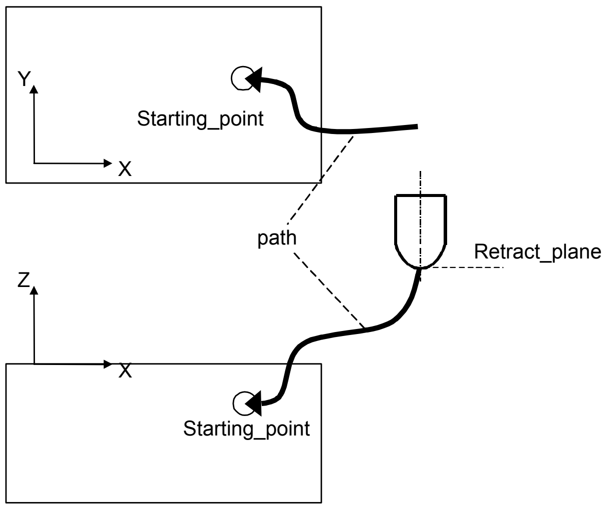


Fig. 8: Along path

Freeform operation

Derived from the milling type operation, this is the class of operations for freeform milling. Note that only some Hi-Tech NC controllers today will not be able to machine a freeform surface without specifying explicit toolpaths.

```
ENTITY freeform_operation
  SUBTYPE OF (milling_type_operation);
  its_machining_strategy : OPTIONAL freeform_strategy;
END_ENTITY;
```

its_machining_strategy: Description of the strategy to be used when executing the operation. In case the attribute its_toolpath of the supertype operation is specified, the strategy is for information only.

Freeform strategy

The following entities define the strategy used for milling a freeform surface. If this entity is used, the toolpath is defined only by means of the milling strategy and the tolerances. The CNC itself has to calculate the resulting toolpaths out of these values.
If the toolpath and the freeform strategy are defined, the attribute "freeform_strategy" is for information only.

```
ENTITY freeform_strategy
```

```
    ABSTRACT SUPERTYPE OF (ONEOF(uv_strategy, plane_cc_strategy,
    plane_cl_strategy, leading_line_strategy));
    pathmode:                pathmode_type;
    cutmode:                 cutmode_type;
    its_milling_tolerances: tolerances;
    stepover:                OPTIONAL length_measure;
  END_ENTITY;
```

pathmode: The feed direction.

cutmode: The stepover direction.

its_milling_tolerances: The tolerance values to be used during creation of the toolpaths.

stepover: The distance between two neighboring toolpaths. If given, the stepover calculated by use of its_milling_tolerances will be ignored.

Pathmode type

The pathmode used in milling. This can be forward (or unidirectional) milling or zigzag (or bidirectional) milling.

```
  TYPE pathmode_type = ENUMERATION OF (
    forward,
    zigzag
  );
  END_TYPE;
```

Cutmode type

The cutting mode used in milling. This can be climb or conventional. In unidirectional mode, climb means that the stepover motion is directed to the left of the feed direction if tool rotation is counter-clockwise. In bidirectional mode, the cutmode type refers to the first cut only.

```
  TYPE cutmode_type = ENUMERATION OF (
    climb,
    conventional
  );
  END_TYPE;
```

Tolerances

The tolerances which are associated with the free form operation. This does not refer to the general manufacturing tolerances but specifies two parameter which are needed if the NC controllers generates toolpaths for free-form surfaces. Through these values the stepover distance between the toolpaths can be derived.

```
  ENTITY tolerances;
    chordal_tolerance : length_measure;
    scallop_height    : length_measure;
  END_ENTITY;
```

chordal_tolerance: Geometric error resulting from a linear approximation of a curve.

scallop_height: Height of the grooves caused by the tool radius (C in the figure).

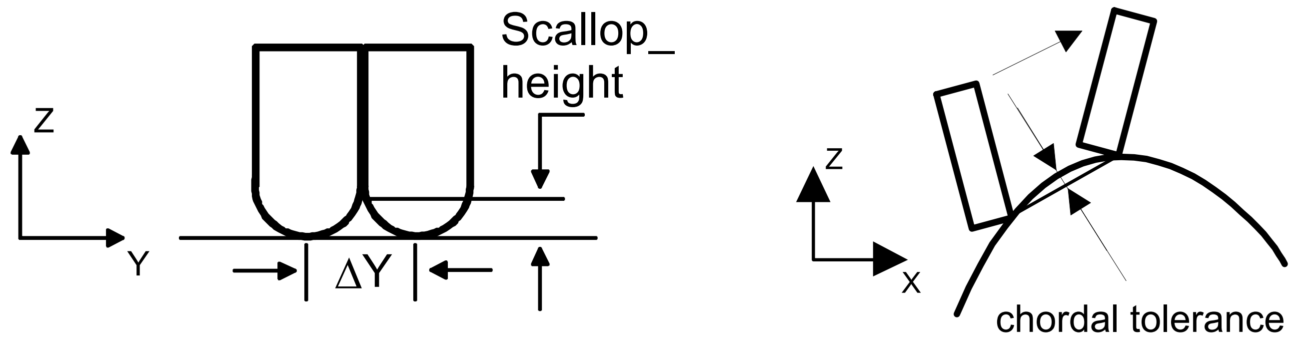


Fig. 9: Scallop height and chordal tolerance

UV strategy

Milling follows the parameter lines in the local (u,v) coordinate system.

```
ENTITY uv_strategy
  SUBTYPE OF (freeform_strategy);
  forward_direction:      direction;
  sideward_direction:     direction;
END_ENTITY;
```

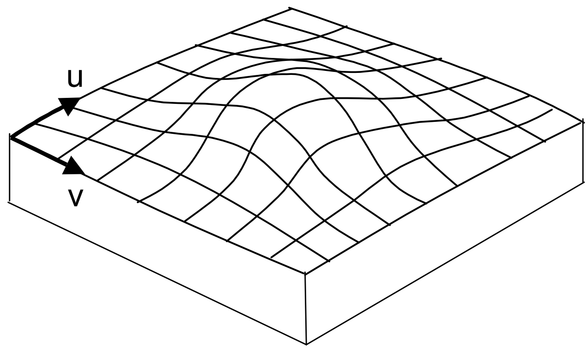


Fig. 10: UV strategy

forward_direction: The direction used in the first cut.

sideward_direction: The direction in which the second cut is offset from the first.

Plane cutter contact strategy

The paths are generated by intersecting the target surface with parallel planes. The result of these intersections form the cutter contact paths.

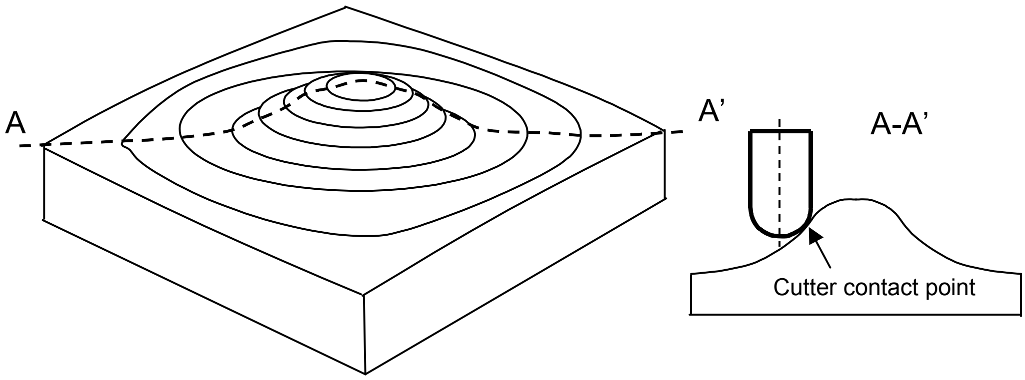


Fig. 11: Plane cutter contact strategy

```
ENTITY plane_cc_strategy
  SUBTYPE OF (freeform_strategy);
  its_plane_normal: direction;
END_ENTITY;
```

its_plane_normal: The normal of the planes used for intersection with the target surface.

Plane cutter location strategy

The paths are generated by intersecting the target surface, offset by the cutter radius, with planes. The result form the cutter location paths. This strategy makes sense with ball end and bullnose cutters.

```
ENTITY plane_cl_strategy
  SUBTYPE OF (freeform_strategy);
  its_plane_normal: direction;
END_ENTITY;
```

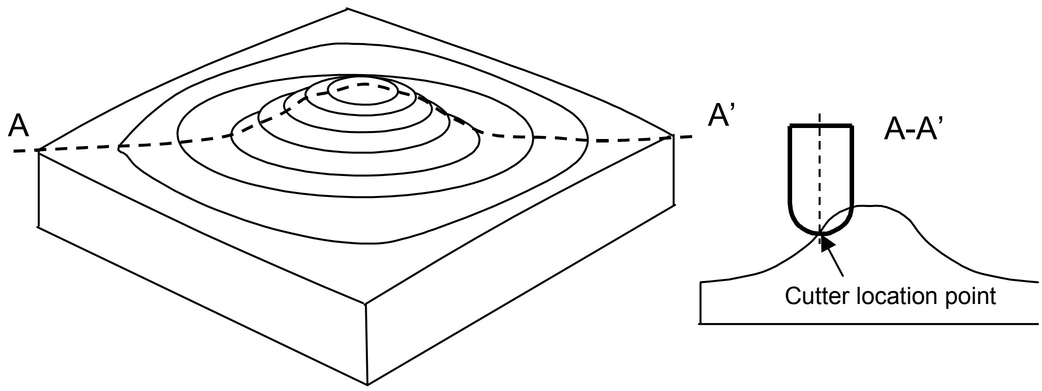


Fig. 12: Plane cutter location strategy

its_plane_normal: The normal of the planes used for intersection with the target surface.

Leading line strategy

The toolpaths are calculated by projecting a curve on the workpiece surface along the Z-axis of local coordinate system. The curve is given as an attribute.

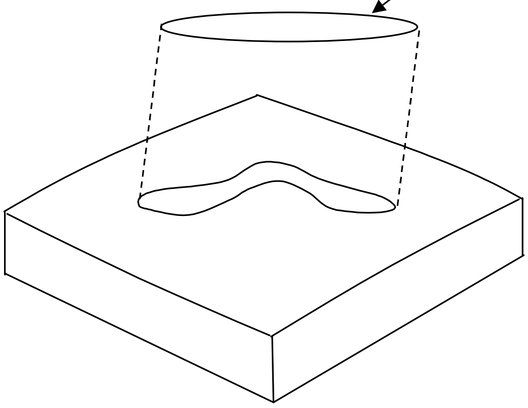


Fig. 13: Leading line strategy

```
ENTITY leading_line_strategy
  SUBTYPE OF (freeform_strategy);
  its_line : bounded_curve;
END_ENTITY;
```

its_line: The curve used to calculate the toolpaths.

Two5D milling operation

This is the base class of all operations for 2½D milling derived from milling_type_operation.

```
ENTITY two5D_milling_operation
  ABSTRACT SUPERTYPE OF (ONEOF(plane_milling, side_milling,
    bottom_and_side_milling))
  SUBTYPE OF (milling_type_operation);
  its_machining_strategy : OPTIONAL two5D_milling_strategy;

END_ENTITY;
```

its_machining_strategy: Description of the strategy to be used when executing the operation. In case the attribute its_toolpath of the supertype operation is specified, the strategy is for information only.

Two5D milling strategy

This is the base class of all strategies used for creating 2½D milling toolpaths

```
ENTITY two5D_milling_strategy
ABSTRACT SUPERTYPE OF (ONEOF (unidirectional, bidirectional, contour_parallel,
  bidirectional_contour, contour_bidirectional, contour_spiral, center_milling,
  explicit_strategy));
  overlap:               OPTIONAL positive_ratio_measure;
  allow_multiple_passes:  OPTIONAL BOOLEAN;
END_ENTITY;
```

overlap: The overlap in the path between two neighbouring cutting movements as percentage of the tool diameter.

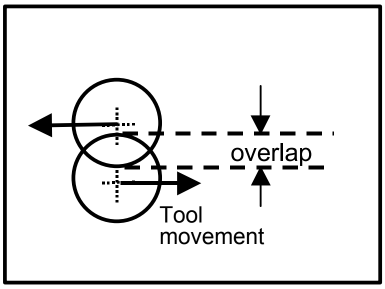


Fig. 14: Overlap

allow_multiple_passes: Optional flag only for roughing workingsteps. If true, this is the standard roughing operation with multiple passes, i. e. several layers of material are removed sequentially, taking into account the maximum cutting depth. If false, this is the special roughing operation for pre-cast features with one pass. Default is true.

Unidirectional milling

Milling in a linear fashion, i.e. going from one side to the other, then lifting the tool and going back to the starting point. In this way, the cutting mode (conventional or climb cutting) is not changed like it is in bidirectional milling. The step over direction is automatically derived from feed_direction and cutmode.

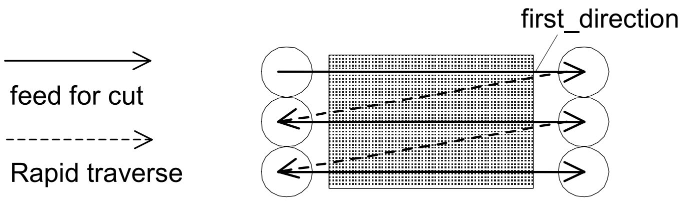


Fig. 15: Unidirectional milling

```
ENTITY unidirectional
  SUBTYPE OF (two5D_milling_strategy);
  feed_direction:        OPTIONAL direction;
  cutmode:                OPTIONAL cutmode_type;
END_ENTITY;
```

feed_direction: Feed direction of the milling operation. The attribute cutmode, if given, takes precedence over this attribute.

cutmode: Specifies whether conventional or climb cutting should be used. Default is conventional.

Bidirectional milling

Milling in a zigzag fashion, i.e. going from one side to the other and back. For further describing the strategy of milling, it may be specified, which are the first and second directions for zigzagging. The cutting mode (conventional or climb cutting) is alternated.

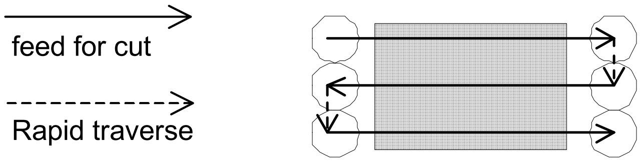


Fig. 16: Bidirectional milling

```
ENTITY bidirectional
  SUBTYPE OF (two5D_milling_strategy);
  feed_direction:        OPTIONAL direction;
  stepover_direction:       OPTIONAL left_or_right;
  its_stroke_connection_strategy:
                           OPTIONAL stroke_connection_strategy;
END_ENTITY;
```

feed_direction: Feed direction of the first toolpath of the milling operation.

stepover_direction: Stepover direction of the zigzag operation.

its_stroke_connection_strategy: Specification of the behaviour of the tool between strokes.

Left or right

Specification of the step over direction relative to the feed direction.

```
TYPE left_or_right = ENUMERATION OF (left, right);
END_TYPE;
```

Stroke connection strategy

Enumerator describing the behaviour of the tool between strokes in bidirectional milling.

```
TYPE stroke_connection_strategy = ENUMERATION OF
  (straghtline, lift_shift_plunge, degouge, loop_back);
END_TYPE;
```

Contour parallel milling

Milling in several paths following the contour of the feature. A typical strategy for pocket milling. The step over direction (outside_in or inside_out) is automatically derived from rotation_direction and cutmode.

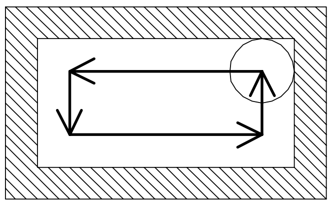


Fig. 17: Contour parallel milling

```
ENTITY contour_parallel
  SUBTYPE OF (two5D_milling_strategy);
  rotation_direction:     OPTIONAL rot_direction;
  cutmode:                OPTIONAL cutmode_type;
END_ENTITY;
```


rotation_direction: Direction of the spiral (clockwise or counterclockwise) as seen from the top of the feature. The default is counterclockwise. The attribute cutmode, if given, takes precedence over this attribute.

cutmode: Specifies whether conventional or climb cutting should be used. Default is conventional. The cutmode refers to the functional walls of the contour which are produced by side milling, i. e. the outer contour of of pocket and possible bosses.

Bidirectional and contour milling

Milling of the contour in bidirectional fashion first, then one final contour-parallel path on the very outside of the feature.

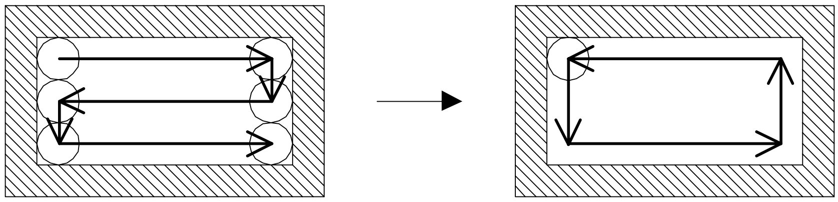


Fig. 18: Bidirectional and contour milling

```
ENTITY bidirectional_contour
  SUBTYPE OF (two5D_milling_strategy);
  feed_direction:         OPTIONAL direction;
  stepover_direction:        OPTIONAL left_or_right;
  rotation_direction:      OPTIONAL rot_direction;
  spiral_cutmode:          OPTIONAL cutmode_type;
END_ENTITY;
```

feed_direction: Feed direction of the first toolpath of the milling operation. The attribute first_cutmode, if given, takes precedence over this attribute.

stepover_direction: Stepover direction of the zigzag operation.

rotation_direction: Direction of the spiral (clockwise or counterclockwise) for the final cut as seen from the top of the feature. The default is counterclockwise. The attribute spiral_cutmode, if given, takes precedence over this attribute.

spiral_cutmode: Specifies whether conventional or climb cutting should be used on the final cut. Default is conventional. The cutmode refers to the functional walls of the contour which are produced by side milling, i. e. the outer contour of pocket and possible bosses.

Contour and bidirectional milling

Milling of a contour parallel path on the very outside of the contour first, then bidirectional milling of the remaining center.

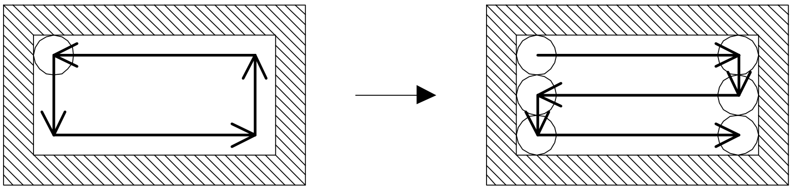


Fig. 19: Contour and bidirectional milling

```
ENTITY contour_bidirectional
  SUBTYPE OF (two5D_milling_strategy);
  feed_direction:        OPTIONAL direction;
  stepover_direction:       OPTIONAL left_or_right;
  rotation_direction:     OPTIONAL rot_direction;
  spiral_cutmode:         OPTIONAL cutmode_type;
END_ENTITY;
```

feed_direction: Feed direction of the first toolpath of the zigzag operation. The attribute first_cutmode, if given, takes precedence over this attribute.

stepover_direction: Stepover direction of the zigzag operation.

rotation_direction: Direction of the spiral (clockwise or counterclockwise) for the final cut as seen from the top of the feature. The default is counterclockwise. The attribute spiral_cutmode, if given, takes precedence over this attribute.

spiral_cutmode: Specifies whether conventional or climb cutting should be used. Default is conventional. The cutmode refers to the functional walls of the contour which are produced by side milling, i. e. the outer contour of pocket and possible bosses.

Contour spiral milling

Contour spiral milling is similar to contour parallel milling, with the exception, that in this case the milling path is a truly spiral path rather than concentric paths which are connected by a orthogonal movement. The step over direction (outside_in or inside_out) is automatically derived from rotation_direction and cutmode.

```
ENTITY contour_spiral
  SUBTYPE OF (two5D_milling_strategy);
  rotation_direction:     OPTIONAL rot_direction;
  cutmode:                OPTIONAL cutmode_type;
END_ENTITY;
```

rotation_direction: The direction of the spiral path (clockwise or counterclockwise) as seen from the top of the feature. The default is counterclockwise. The attribute cutmode, if given, takes precedence over this attribute.

cutmode: Specifies whether conventional or climb cutting should be used. Default is conventional. The cutmode refers to the functional walls of the contour which are produced by side milling, i. e. the outer contour of of pocket and possible bosses.

Center milling

This entity describes a milling strategy along the center of the feature. This is used e.g. for milling along the center of a slot.

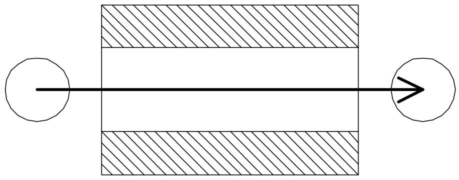


Fig. 20: Center milling

```
ENTITY center_milling
  SUBTYPE OF (two5D_milling_strategy);
END_ENTITY;
```

Explicit_strategy

Any two5D strategy which can not be described using any of the above given types can be specified using explicit. In this case, an exact definition of all movements needs to be given in the attribute its_toolpath of the entity workingstep.

```
ENTITY explicit_strategy
  SUBTYPE OF (two5D_milling_strategy);
END_ENTITY;
```

Plane milling

Entity to describe the milling of a plane. This is the supertype for roughing and finishing operations.

```
ENTITY plane_milling
  ABSTRACT SUPERTYPE OF (ONEOF(plane_rough_milling, plane_finish_milling))
  SUBTYPE OF (two5D_milling_operation);
  axial_cutting_depth:        OPTIONAL length_measure;
  allowance_bottom:           OPTIONAL length_measure;
END_ENTITY;
```

axial_cutting_depth:	The cutting depth in the direction of the tool axis. This can be given to specify a maximal cutting depth smaller than the material removal required by the feature's depth. As a result, several layers will be manufactured. If omitted, the selected cutting depth will be implementation dependent.
allowance_bottom:	The allowance is a layer of material which will be left on top of the plane surface defined by the associated manufacturing feature.

Plane rough milling

Roughing operation for milling. All material inside the manufacturing feature will be removed using the given tool, except for the allowance_bottom.

```
ENTITY plane_rough_milling
  SUBTYPE OF (plane_milling);
  WHERE
  WR1: EXISTS(SELF.allowance_bottom) AND (SELF.allowance_bottom >= 0.0);
END_ENTITY;
```

Plane finish milling

Finishing operation for milling. All material inside the manufacturing feature will be removed, applying an appropriate strategy to maintain the given tolerances. If allowance_bottom is given, other special operation like grinding shall be applied for removing the material left.

```
ENTITY plane_finish_milling
  SUBTYPE OF (plane_milling);
END_ENTITY;
```

Side milling

Entity to describe a side milling process during which material is removed along the flank of the tool.

```
ENTITY side_milling
  ABSTRACT SUPERTYPE OF (ONEOF(side_rough_milling, side_finish_milling))
  SUBTYPE OF (two5D_milling_operation);
  axial_cutting_depth:    OPTIONAL length_measure;
  radial_cutting_depth:   OPTIONAL length_measure;
  allowance_side:         OPTIONAL length_measure;
END_ENTITY;
```

axial_cutting_depth:	The cutting depth in the direction of the tool axis. See plane milling.
radial_cutting_depth:	The cutting depth perpendicular to the tool axis. This can be used to limit the chip thickness. If radial_cutting_depth is smaller than the radial material removal required by the feature, this will cause the execution of the operation in several layers.
allowance_side:	The allowance is a layer of material which will be left on side of the surface defined by the associated manufacturing feature.

Side rough milling

Roughing operation for side milling. All material inside the manufacturing feature will be removed using the given tool, except for the allowance_side.

```
ENTITY side_rough_milling
  SUBTYPE OF (side_milling);
  WHERE
  WR1: EXISTS(SELF.allowance_side) AND (SELF.allowance_side >= 0.0);
END_ENTITY;
```

Side finish milling

Finishing operation for side milling. All material inside the manufacturing feature will be removed, applying an appropriate strategy to maintain the given tolerances. If allowance_side is given, other special operation like grinding shall be applied for removing the material left.

```
ENTITY side_finish_milling
  SUBTYPE OF (side_milling);
END_ENTITY;
```

Bottom and side milling

Entity to describe a combined bottom and side milling process.

```
ENTITY bottom_and_side_milling
  ABSTRACT SUPERTYPE OF (ONEOF(bottom_and_side_rough_milling,
    bottom_and_side_finish_milling))
  SUBTYPE OF (two5D_milling_operation);
  axial_cutting_depth:    OPTIONAL length_measure;
```

```
    radial_cutting_depth:   OPTIONAL length_measure;
    allowance_side:         OPTIONAL length_measure;
    allowance_bottom:         OPTIONAL length_measure;
  END_ENTITY;
```

axial_cutting_depth: The cutting depth in the direction of the tool. See plane milling.

radial_cutting_depth: The cutting depth perpendicular to the tool, used in side milling. See side milling.

allowance_side: The allowance is a layer of material which will be left on side of the surface defined by the associated manufacturing feature.

allowance_bottom: The allowance is a layer of material which will be left on top of the plane surface defined by the associated manufacturing feature.

Bottom and side rough milling

Roughing operation for a combined bottom and side milling workingstep. All material inside the manufacturing feature will be removed using the given tool, except for the allowance_side and allowance_bottom.

```
  ENTITY bottom_and_side_rough_milling
    SUBTYPE OF (bottom_and_side_milling);

    WHERE
     WR1: EXISTS(SELF.allowance_side) AND (SELF.allowance_side>=0.0);
     WR2: EXISTS(SELF.allowance_bottom) AND (SELF.allowance_bottom>=0.0);
  END_ENTITY;
```

Bottom and side finish milling

Finishing operation for a combined bottom and side milling workingstep. All material inside the manufacturing feature will be removed, applying an appropriate strategy to maintain the given tolerances. If allowance_side and allowance_bottom are given, other special operation like grinding shall be applied for removing the material left.

```
  ENTITY bottom_and_side_finish_milling
    SUBTYPE OF (bottom_and_side_milling);
  END_ENTITY;
```

Drilling type operation

This is the base class for all operations concerned with drilling a hole, reaming, sinking, etc. It is a subtype of entity milling_machining_operation. Cutting of a thread is included here also. The base class provides all necessary attributes to describe technology and strategy for drilling type operations. In case of pre-cast holes, the predrilling can be operated before the finish drilling by means of specifying a drilling depth (and an appropriate tool) which is smaller than that of the feature. Subsequent drilling operations can specify the attribute previous_diameter appropriately to allow for the already removed material.

The start point is given by the inherited attributes retract_plane and cut_start_point. If cut_start_point is omitted, the centre of the underlying feature will be used instead. From there the tool will advance with drilling feed along the local z axis. Using the prescribed strategy, the tool will drill to the depth of the associated feature, for through holes applying an inherited attribute overcut_length. For pre-drilling operations, the attribute cutting_depth can be used to reduce the depth of a cut to an amount smaller than the hole's depth.

On retract, the tool will return to the retract plane using the drilling feed, or – if specified – the feed_on_retract given by drilling_type_strategy.

Note that all geometric information in these operations is given in the local co-ordinate system of the underlying feature.

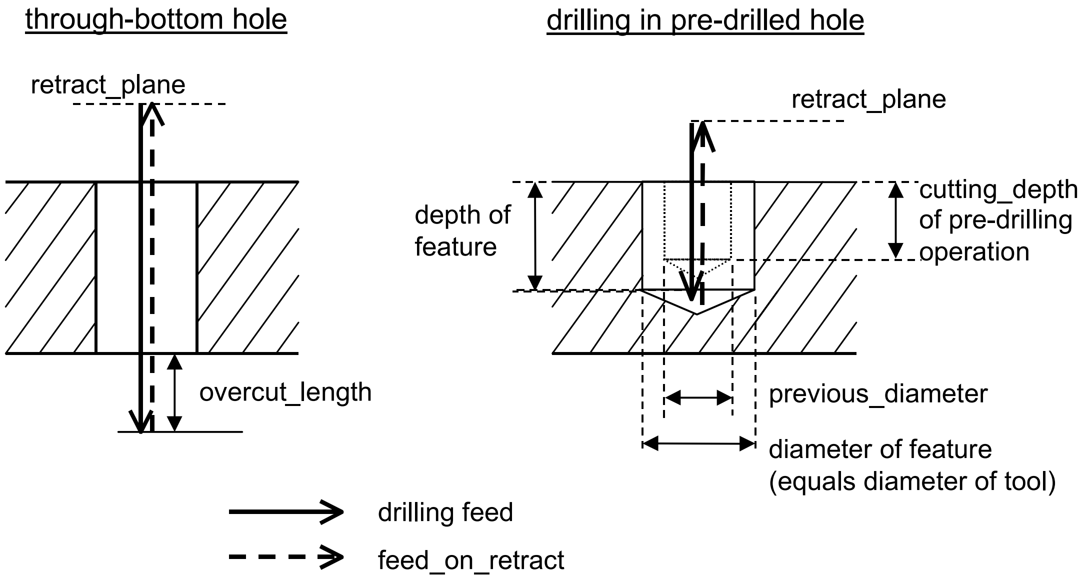


Fig. 21: Drilling type operation

```
ENTITY drilling_type_operation
  ABSTRACT SUPERTYPE OF (ONEOF(drilling_operation, boring_operation, back_boring,
    tapping, thread_drilling))
  SUBTYPE OF (milling_machining_operation);
  cutting_depth:              OPTIONAL length_measure;
  previous_diameter:          OPTIONAL length_measure;
  dwell_time_bottom:          OPTIONAL time_measure;
  feed_on_retract:            OPTIONAL positive_ratio_measure;
  its_machining_strategy:     OPTIONAL drilling_type_strategy;
END_ENTITY;
```

cutting_depth: The depth of the cut of this operation, which may differ from the depth of the hole as such. The NC controller will not check if cutting_depth violates the boundaries of the associated hole feature. If omitted, the total depth of the feature will be drilled. In case of center drilling operation, the cutting_depth is measured from the lowest point of the cutting tip to the highest point of the hole. In other cases, it is measured from the starting point of cylindrical part of the tool.(Or, tapered cylindrical part in case of tapered drill.)

previous_diameter: If the operation is performed on a pre-drilled or pre-cast hole, this value, if given, specifies the diameter of the existing hole. It thus describes the amount of material which the tool as to remove and is for information only.

dwell_time_bottom: Possible dwell time at the bottom of the hole.

feed_on_retract: Feed used for retract to the retract_plane as ratio of the drilling feed. If not specified, the drilling feed is used.

its_machining_strategy: Description of the strategy to be used when executing the operation. In case the attribute its_toolpath of the supertype operation is specified, the strategy is for information only.

Note: A drilling operation cannot only be used with holes but with all sorts of features. For example, a plunge drilling operation would be associated with a pocket. In that case, however, depending on the type of tool, the bottom of the feature may be violated if the entire depth of the feature is drilled (compare Fig. 21). To avoid this, a smaller cutting_depth can be specified explicitly. Also, it may be advisable for non-rotary features to explicitly specify the inherited attribute cut_start_point.

Drilling type strategy

This is the specification of a dedicated strategy for drilling. For drilling, this mainly refers to a variation of cutting speed and feed along the movement of the tool.

```
ENTITY drilling_type_strategy;
  reduced_cut_at_start:       OPTIONAL positive_ratio_measure;
  reduced_feed_at_start:      OPTIONAL positive_ratio_measure;
  depth_of_start:             OPTIONAL length_measure;
  reduced_cut_at_end:         OPTIONAL positive_ratio_measure;
  reduced_feed_at_end:        OPTIONAL positive_ratio_measure;
  depth_of_end:               OPTIONAL length_measure;
WHERE
  WR1: EXISTS(depth_of_start) OR NOT (EXISTS(reduced_cut_at_start) OR
    EXISTS(reduced_feed_at_start));
  WR2: EXISTS(depth_of_end) OR NOT (EXISTS(reduced_cut_at_end) OR
    EXISTS(reduced_feed_at_end));
END_ENTITY;
```

reduced_cut_at_start.	Reduced cutting speed at the beginning of the cut as a percentage of the programmed value.
reduced_feed_at_start:	Reduced feed at the beginning of the cut as a percentage of the programmed value.
depth_of_start:	Depth to which the reduced values at the start are valid.
reduced_cut_at_end.	Reduced cutting speed at the end of the cut as a percentage of the programmed value.
reduced_feed_at_end:	Reduced feed at the end of the cut as a percentage of the programmed value.
depth_of_end:	Depth from which the reduced values at the end are valid.

Drilling operation

Base class for drilling operation concerned with drilling, center drilling, counter sinking, and multistep drilling.

```
ENTITY drilling_operation
  ABSTRACT SUPERTYPE OF (ONEOF(drilling, center_drilling, counter_sinking,
    multistep_drilling))
  SUBTYPE OF (drilling_type_operation);
END_ENTITY;
```

Drilling

Workingstep for drilling a regular hole.

```
ENTITY drilling
  SUBTYPE OF (drilling_operation);
END_ENTITY;
```

Center drilling

Workingstep for centering a hole.

```
ENTITY center_drilling
  SUBTYPE OF (drilling_operation);
END_ENTITY;
```

Counter sinking

Workingstep for counter sinking a hole.

```
ENTITY counter_sinking
  SUBTYPE OF (drilling_operation);
END_ENTITY;
```

Multistep drilling

Workingstep for drilling of deep holes in multiple steps.

```
ENTITY multistep_drilling
  SUBTYPE OF (drilling_operation);
  retract_distance:       length_measure;
  first_depth:            length_measure;
  depth_of_step:          length_measure;
  dwell_time_step:        OPTIONAL time_measure;
END_ENTITY;
```

retract_distance: If retract_distance is a positive value, the tool is retracted this value between steps to enable chip breaking. If it is zero or negative, the tool is retracted to the retract plane between steps to clear the hole of chips.

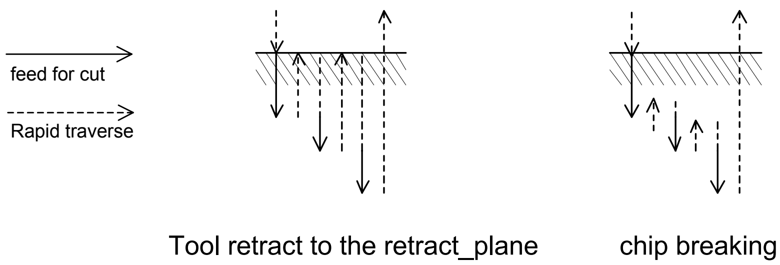


Fig. 22: Multi-step drilling

first_depth: Depth of the first step.

depth_of_step: Depth of each additional step (repeated until the depth of the hole is reached).

dwell_time_step: Dwell time between steps.

Note: If more complex drilling operations are needed, e. g. for special tools, this can be specified by an explicit toolpath definition in the workingstep's its_toolpath attribute.

Boring operation

Base class for boring operation concerned with boring and reaming. The spindle orientation at the bottom should be given.

```
ENTITY boring_operation
  ABSTRACT SUPERTYPE OF (ONEOF(boring, reaming))
  SUBTYPE OF (drilling_type_operation);
  spindle_stop_at_bottom:         BOOLEAN;
```

```
    depth_of_testcut:               OPTIONAL length_measure;
    waiting_position:               OPTIONAL cartesian_point;
  END_ENTITY;
```

spindle_stop_at_bottom: Possible spindle stop at the bottom of the hole. If the attribute oriented_spindle_stop in the workingstep's technology is set, this will cause an oriented spindle stop.

depth_of_testcut: Depth of a testcut after which the hole is measured.

waiting_position: A waiting position for the tool i.e. to allow measuring. The tool moves out of the hole along the tool axis until it reaches the plane of the waiting_position. It then moves to the waiting position itself.

Boring

Workingstep for boring a hole.

```
  ENTITY boring
    SUBTYPE OF (boring_operation);
  END_ENTITY;
```

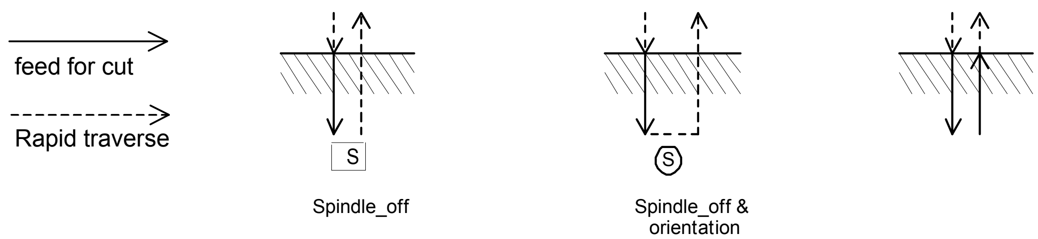


Fig. 23: Hole boring

Reaming

Workingstep for reaming a hole.

```
  ENTITY reaming
    SUBTYPE OF (boring_operation);
  END_ENTITY;
```

Back boring

Workingstep for back boring a hole. Backboring is the cutting of the back plane of a through bottom hole. After the tool is positioned at the start point, the spindle stops at the direction_of_spindle_orientation specified as an attribute of milling_cutting_tool. The tool is shifted in the opposite of direction_of_spindle_orientation in order to path through the hole with rapid traverse. At the bottom of the hole, it returns to the cutting position for the shifted value and the spindle starts to rotate. In special tools whose cutting edge can be hidden inside the tool body, the spindle will stop after passing through the hole and will reverse its direction. The attribute oriented_spindle_stop of its_maching_functions is required and will cause an oriented spindle stop which is needed for collapsible backboring tools.

```
  ENTITY back_boring
    SUBTYPE OF (drilling_type_operation);
  WHERE
     WR1:  EXISTS(SELF.its_machine_functions.oriented_spindle_stop);
  END_ENTITY;
```

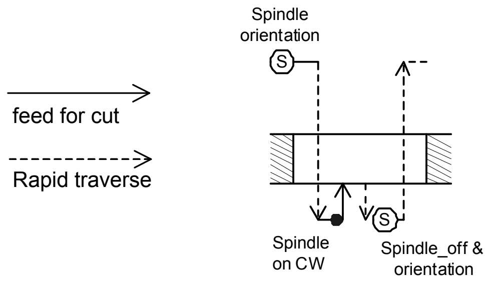


Fig. 24: Hole backboring

Note: If more complex boring operations are needed, e.g. for special tools, this can be specified by an explicit toolpath definition in the workingstep's its_toolpath attribute.

Tapping

Workingstep for tapping (or threading) a hole. This is performed with a special cutter which is rotated and moved along the tool axis.

```
ENTITY tapping
  SUBTYPE OF (drilling_type_operation);
    compensation_chuck:   BOOLEAN;
END_ENTITY;
```

compensation_chuck: If true, a compensation chuck shall be used for tapping the hole.

Thread drilling

Workingstep for thread drilling a hole. This involves a helical movement of the tool. It is sometimes also called "thrilling", short for "thread drilling". A helical movement may be needed for the forward movement or not. The feed per revolution is calculated from the thread of the hole. The pitch of the helical movement corresponds to the pitch of the thread. The feedrate is the relative speed between tool tip and material along the helical path.

```
ENTITY thread_drilling
  SUBTYPE OF (drilling_type_operation);
  helical_movement_on_forward:   BOOLEAN;
END_ENTITY;
```

helical_movement_on_forward:
Specifies if the helical movement is needed for the forward operation as well.

Conformance requirements

Conformance to this part of ISO 14649 includes satisfying the requirements stated in this part, the requirements of the implementation methods supported, and the relevant requirements of the normative references.
For requirements with respect to implementation methods see Annex C.
This part of ISO 14649 provides a number of options that may be supported by an implementation. These options shall all be supported by six classes of conformance.
These conformance classes are characterized as follows:

- conformance class 1 – `c0m0`: Minimum set of curve geometry and minimum set of manufacturing data;

- conformance class 2 – `c0m0m1`: Class 1 plus full set of manufacturing data, especially manufacturing features;
- conformance class 3 – `s0c0m0m1`: Class 2 plus minimum set of surface geometry;
- conformance class 4 – `s0s1c0c1m0m1`: Class 3 plus full curve and surface geometry;
- conformance class 5 – `t0t1s0c0m0m1`: Class 3 plus topological information;
- conformance class 6 – `t0t1s0s1c0c1m0m1`: Class 4 plus topological information.

The identifiers for the respective data sets (`m0/m1`, `c0/c1`, `s0/s1`, `t0/t1`) can be found in the EXPRESS listing in Appendix A for each member of the data set in order to facilitate implementation. As all entities defined in this part of ISO14649 belong to one of m0 or m1, the list of conformance class 2 described in the following section is the same with the list for class 3, class 4, class 5, and class 6.

5.1 Conformance class 1 entities

An implementation of conformance class 1 of this part of ISO 14649 shall support the following entities and related constructs:

```
back_boring
backside_counterbore
backside_countersink
ball_endmill
boring
boring_operation
boring_tool
bottom_and_side_finish_milling
bottom_and_side_milling
bottom_and_side_rough_milling
bullnose_endmill
center_drill
center_drilling
combined_drill_and_reamer
combined_drill_and_tap
counter_sinking
counterbore
countersink
dovetail_mill
drill
drilling
drilling_operation
drilling_type_operation
drilling_type_strategy
endmill
exchange_pallet
facemill
five_axes_const_tilt_yaw
freeform_operation
index_pallet
index_table
load_tool
milling_cutter
milling_cutting_tool
milling_machine_functions
milling_machining_operation
milling_technology
milling_tool_body
milling_type_operation
multistep_drilling
plane_finish_milling
plane_milling
```

```
plane_rough_milling
reamer
reaming
side_finish_milling
side_mill
side_milling
side_rough_milling
spade_drill
t_slot_mill
tap
tapered_drill
tapered_endmill
tapered_reamer
tapered_tap
tapping
thread_drilling
thread_mill
threading_tool
three_axes_tilted_tool
tool_dimension
tool_direction_for_milling
twist_drill
two5D_milling_operation
unload_tool
user_defined_tool
woodruff_keyseat_mill
```

Conformance class 2 entities

An implementation of conformance class 2 of this part of ISO 14649 shall support the following entities and related constructs:

```
adaptive_control
air_strategy
along_path
ap_retract_angle
ap_retract_tangent
approach_retract_strategy
back_boring
backside_counterbore
backside_countersink
ball_endmill
bidirectional
bidirectional_contour
boring
boring_operation
boring_tool
bottom_and_side_finish_milling
bottom_and_side_milling
bottom_and_side_rough_milling
bullnose_endmill
center_drill
center_drilling
center_milling
combined_drill_and_reamer
combined_drill_and_tap
contour_bidirectional
contour_parallel
contour_spiral
counter_sinking
counterbore
```

```
countersink
dovetail_mill
drill
drilling
drilling_operation
drilling_type_operation
drilling_type_strategy
endmill
exchange_pallet
explicit_strategy
facemill
five_axes_const_tilt_yaw
five_axes_var_tilt_yaw
freeform_operation
freeform_strategy
index_pallet
index_table
leading_line_strategy
load_tool
milling_cutter
milling_cutting_tool
milling_machine_functions
milling_machining_operation
milling_technology
milling_tool_body
milling_type_operation
multistep_drilling
plane_cc_strategy
plane_cl_strategy
plane_finish_milling
plane_milling
plane_rough_milling
plunge_helix
plunge_ramp
plunge_strategy
plunge_toolaxis
plunge_zigzag
process_model
process_model_list
reamer
reaming
side_finish_milling
side_mill
side_milling
side_rough_milling
spade_drill
t_slot_mill
tap
tapered_drill
tapered_endmill
tapered_reamer
tapered_tap
tapping
thread_drilling
thread_mill
threading_tool
three_axes_tilted_tool
tolerances
tool_dimension
tool_direction_for_milling
twist_drill
```

```
two5D_milling_operation
two5D_milling_strategy
unidirectional
unload_tool
user_defined_tool
uv_strategy
woodruff_keyseat_mill
```

Annex A
(normative)

EXPRESS listing

The following EXPRESS is the whole schema given in clause 6. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used. The two-character labels used for each entity indicate to which conformance class an entity belongs; please refer to Chapter 5.

```
SCHEMA milling_schema;
(* Version 19 date: 2002-02-06
 * Author: ISO TC184/SC1/WG7
*)

(* ********************************************************** *)
(* Types from machining_schema                 ISO 14649-10   *)
(* ********************************************************** *)

REFERENCE FROM machining_schema(
        bounded_curve,
        cartesian_point,
        direction,
        identifier,
        label,
        length_measure,
        nc_function,
        machine_functions,
        machining_operation,
        machining_tool,
        material,
        plane_angle_measure,
        positive_ratio_measure,
        pressure_measure,
        property_parameter,
        rot_direction,
        rot_speed_measure,
        speed_measure,
        technology,
        time_measure,
        toolpath_list,
        tool_direction);


(* ********************************************************** *)
(* ********************************************************** *)
(*                                                            *)
(* Types from tools for milling                ISO 14649-111  *)
(*                                                            *)
(* ********************************************************** *)
(* ********************************************************** *)

ENTITY milling_cutting_tool                                                   (* m0 *)
SUBTYPE OF (cutting_tool);
direction_for_spindle_orientation:
                             OPTIONAL direction;
tool_holder_diameter_for_spindle_orientation:
```

```
                             OPTIONAL length_measure;
END_ENTITY;

(* ********************************************************** *)
(* Milling tool body                                          *)
(* ********************************************************** *)

ENTITY milling_tool_body                                                    (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(center_drill, countersink, drill, milling_cutter,
  tap, threading_tool, counterbore, reamer, boring_tool, user_defined_tool))
SUBTYPE OF (tool_body);
dimension:                   tool_dimension;
number_of_teeth:             OPTIONAL INTEGER;
hand_of_cut:                 OPTIONAL hand;
coolant_through_tool:        OPTIONAL BOOLEAN;
pilot_length:                OPTIONAL length_measure;END_ENTITY;

(* ********************************************************** *)
(* Tool dimension                                             *)
(* ********************************************************** *)

ENTITY tool_dimension;                                                      (* m0 *)
diameter:                    length_measure;
tool_top_angle:              OPTIONAL plane_angle_measure;
tool_circumference_angle:    OPTIONAL plane_angle_measure;
cutting_edge_length:         OPTIONAL length_measure;
edge_radius:                 OPTIONAL length_measure;
edge_center_vertical:        OPTIONAL length_measure;
edge_center_horizontal:      OPTIONAL length_measure;
END_ENTITY;


TYPE hand = ENUMERATION OF(left, right, neutral);
END_TYPE;

ENTITY center_drill                                                         (* m0 *)
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY countersink                                                          (* m0 *)
SUPERTYPE OF (backside_countersink)
SUBTYPE OF (milling_tool_body);
countersink_radius:    OPTIONAL length_measure;
END_ENTITY;

ENTITY backside_countersink                                                 (* m0 *)
SUBTYPE OF (countersink);
END_ENTITY;

ENTITY drill                                                                (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(twist_drill, spade_drill))
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY twist_drill                                                          (* m0 *)
SUPERTYPE OF (tapered_drill)
SUBTYPE OF (drill);
END_ENTITY;

ENTITY tapered_drill                                                        (* m0 *)
SUBTYPE OF (twist_drill);
```

```
taper_angle:                OPTIONAL plane_angle_measure;
END_ENTITY;

ENTITY spade_drill                                                    (* m0 *)
SUBTYPE OF (drill);
END_ENTITY;

ENTITY milling_cutter                                                 (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(facemill, endmill, t_slot_mill, dovetail_mill,
  woodruff_keyseat_mill, side_mill, thread_mill))
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY facemill                                                       (* m0 *)
SUBTYPE OF (milling_cutter);
END_ENTITY;

ENTITY endmill                                                        (* m0 *)
SUPERTYPE OF (ONEOF(tapered_endmill, ball_endmill, bullnose_endmill))
SUBTYPE OF (milling_cutter);
END_ENTITY;

ENTITY tapered_endmill                                                (* m0 *)
SUBTYPE OF (endmill);
taper_angle:          OPTIONAL plane_angle_measure;
END_ENTITY;

ENTITY ball_endmill                                                   (* m0 *)
SUBTYPE OF (endmill);
WHERE
WR1: (NOT EXISTS(SELF.dimension.edge_center_horizontal))
     OR ((EXISTS(SELF.dimension.edge_center_horizontal)) AND
     (SELF.dimension.edge_center_horizontal = 0));
WR2: (NOT EXISTS(SELF.dimension.edge_center_vertical))
     OR ((EXISTS(SELF.dimension.edge_center_vertical)) AND
     (SELF.dimension. edge_center_vertical = SELF.dimension.diameter/2));
WR3: (NOT EXISTS(SELF.dimension.edge_radius))
     OR ((EXISTS(SELF.dimension. edge_radius)) AND
     (SELF.dimension.edge_radius = SELF.dimension.diameter/2));
WR4: (NOT EXISTS(SELF.dimension.tool_top_angle))
     OR ((EXISTS(SELF.dimension.tool_top_angle)) AND
     (SELF.dimension.tool_top_angle = 0));
WR5: (NOT EXISTS(SELF.dimension.tool_circumference_angle))
     OR ((EXISTS(SELF.dimension.tool_circumference_angle)) AND
     (SELF.dimension.tool_circumference_angle = 0));
END_ENTITY;

ENTITY bullnose_endmill                                               (* m0 *)
SUBTYPE OF (endmill);
WHERE
WR1: (NOT EXISTS(SELF.dimension.tool_top_angle))
     OR ((EXISTS(SELF.dimension.tool_top_angle)) AND
     (SELF.dimension.tool_top_angle = 0));
WR2: (NOT EXISTS(SELF.dimension.tool_circumference_angle))
     OR ((EXISTS(SELF.dimension.tool_circumference_angle)) AND
     (SELF.dimension.tool_circumference_angle = 0));
END_ENTITY;

ENTITY t_slot_mill                                                    (* m0 *)
SUBTYPE OF (milling_cutter);
cutting_thickness:    OPTIONAL length_measure;
```



```
END_ENTITY;

ENTITY dovetail_mill                                           (* m0 *)
SUBTYPE OF (milling_cutter);
included_angle:        OPTIONAL plane_angle_measure;
END_ENTITY;

ENTITY woodruff_keyseat_mill                                   (* m0 *)
SUBTYPE OF (milling_cutter);
cutter_width:          OPTIONAL length_measure;
END_ENTITY;

ENTITY side_mill                                               (* m0 *)
SUBTYPE OF (milling_cutter);
cutter_width:          OPTIONAL length_measure;
END_ENTITY;

ENTITY thread_mill                                             (* m0 *)
SUBTYPE OF (milling_cutter);
END_ENTITY;

ENTITY tap                                                     (* m0 *)
SUPERTYPE OF (ONEOF(tapered_tap, combined_drill_and_tap))
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY tapered_tap                                             (* m0 *)
SUBTYPE OF (tap);
taper_angle:           OPTIONAL plane_angle_measure;
END_ENTITY;

ENTITY combined_drill_and_tap                                  (* m0 *)
SUBTYPE OF (tap);
drill_length:          OPTIONAL length_measure;
END_ENTITY;

ENTITY threading_tool                                          (* m0 *)
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY counterbore                                             (* m0 *)
SUPERTYPE OF (backside_counterbore)
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY backside_counterbore                                    (* m0 *)
SUBTYPE OF (counterbore);
END_ENTITY;

ENTITY reamer                                                  (* m0 *)
SUPERTYPE OF (ONEOF(tapered_reamer, combined_drill_and_reamer))
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY tapered_reamer                                          (* m0 *)
SUBTYPE OF (reamer);
taper_angle:           OPTIONAL plane_angle_measure;
END_ENTITY;

ENTITY combined_drill_and_reamer                               (* m0 *)
SUBTYPE OF (reamer);
```

```
drill_length:          OPTIONAL length_measure;
END_ENTITY;

ENTITY boring_tool                                                    (* m0 *)
SUBTYPE OF (milling_tool_body);
END_ENTITY;

ENTITY user_defined_tool                                              (* m0 *)
SUBTYPE OF (milling_tool_body);
identifier:   label;
END_ENTITY;


(* ********************************************************** *)
(* ********************************************************** *)
(*                                                            *)
(* Types defined in process data for milling     ISO 14649-11   *)
(*                                                            *)
(* ********************************************************** *)
(* ********************************************************** *)

(* ********************************************************** *)
(* NC functions for milling                                   *)
(* ********************************************************** *)

ENTITY exchange_pallet                                                (* m0 *)
  SUBTYPE OF (nc_function);
END_ENTITY;

ENTITY index_pallet                                                   (* m0 *)
  SUBTYPE OF (nc_function);
  its_index:            INTEGER;
END_ENTITY;

ENTITY index_table                                                    (* m0 *)
  SUBTYPE OF (nc_function);
  its_index:            INTEGER;
END_ENTITY;

ENTITY load_tool                                                      (* m0 *)
  SUBTYPE OF (nc_function);
  its_tool:             machining_tool;
END_ENTITY;

ENTITY unload_tool                                                    (* m0 *)
  SUBTYPE OF (nc_function);
  its_tool:             OPTIONAL machining_tool;
END_ENTITY;

(* ********************************************************** *)
(* Technology-specific Tool direction select                  *)
(* ********************************************************** *)

ENTITY tool_direction_for_milling                                     (* m0 *)
  ABSTRACT SUPERTYPE OF (ONEOF(three_axes_tilted_tool, five_axes_var_tilt_yaw,
  five_axes_const_tilt_yaw))
  SUBTYPE OF (tool_direction);
END_ENTITY;

ENTITY three_axes_tilted_tool                                         (* m0 *)
SUBTYPE OF (tool_direction_for_milling);
```

```
its_tool_direction: direction;
END_ENTITY;

ENTITY five_axes_var_tilt_yaw                                          (* m1 *)
SUBTYPE OF (tool_direction_for_milling);
END_ENTITY;

ENTITY five_axes_const_tilt_yaw                                        (* m0 *)
SUBTYPE OF (tool_direction_for_milling);
tilt_angle : plane_angle_measure;
yaw_angle  : plane_angle_measure;
END_ENTITY;

(* ****************************************************************** *)
(* Base class for technology specific operation and strategy    *)
(* ****************************************************************** *)

ENTITY milling_machining_operation                                     (* m0 *)
   ABSTRACT SUPERTYPE OF (ONEOF(milling_type_operation, drilling_type_operation))
   SUBTYPE OF (machining_operation);
overcut_length: OPTIONAL length_measure;
WHERE
    WR1: (EXISTS(SELF.its_technology.feedrate_per_tooth) AND
          EXISTS(SELF.its_tool.its_tool_body.number_of_teeth))
          OR(NOT(EXISTS(SELF.its_technology.feedrate_per_tooth)));
END_ENTITY;


(* ****************************************************************** *)
(* Milling technology                                                  *)
(* ****************************************************************** *)

ENTITY milling_technology                                              (* m0 *)
SUBTYPE OF (technology);
cutspeed:                       OPTIONAL speed_measure;
spindle:                        OPTIONAL rot_speed_measure;
feedrate_per_tooth:             OPTIONAL length_measure;
synchronize_spindle_with_feed: BOOLEAN;
inhibit_feedrate_override:      BOOLEAN;
inhibit_spindle_override:       BOOLEAN;
its_adaptive_control:           OPTIONAL adaptive_control;
WHERE
WR1: (EXISTS(cutspeed) AND NOT EXISTS(spindle))
OR (EXISTS(spindle) AND NOT EXISTS(cutspeed))
OR (EXISTS(its_adaptive_control));
WR2: (EXISTS(SELF.feedrate) AND NOT EXISTS(feedrate_per_tooth))
OR (EXISTS(feedrate_per_tooth) AND NOT EXISTS(SELF.feedrate))
OR (EXISTS(its_adaptive_control));
END_ENTITY;

ENTITY adaptive_control;                                               (* m1 *)
END_ENTITY;

(* ****************************************************************** *)
(* Milling machine functions                                           *)
(* ****************************************************************** *)

ENTITY milling_machine_functions                                       (* m0 *)
SUBTYPE OF (machine_functions);
coolant             :       BOOLEAN;
coolant_pressure    :       OPTIONAL pressure_measure;
```

```
mist                 :       OPTIONAL BOOLEAN;
through_spindle_coolant:    BOOLEAN;
through_pressure:            OPTIONAL pressure_measure;
axis_clamping        :       LIST [0:?] OF identifier;
chip_removal         :       BOOLEAN;
oriented_spindle_stop:       OPTIONAL direction;
its_process_model:           OPTIONAL process_model_list;
other_functions      :       SET [0:?] OF property_parameter;
END_ENTITY;

ENTITY process_model_list;                                              (* m1 *)
its_list: LIST [1:?] OF process_model;
END_ENTITY;

ENTITY process_model;                                                   (* m1 *)
ini_data_file:  label;
its_type:       label;
END_ENTITY;

(* ********************************************************* *)
(* Milling type operation                                     *)
(* ********************************************************* *)

ENTITY milling_type_operation                                           (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(freeform_operation, two5D_milling_operation))
SUBTYPE OF (milling_machining_operation);
approach:         OPTIONAL approach_retract_strategy;
retract:          OPTIONAL approach_retract_strategy;
END_ENTITY;

(* ********************************************************* *)
(* approach retract strategy                                  *)
(* ********************************************************* *)

ENTITY approach_retract_strategy                                        (* m1 *)
ABSTRACT SUPERTYPE OF (ONEOF (plunge_strategy, air_strategy, along_path));
tool_orientation: OPTIONAL direction;
END_ENTITY;

ENTITY plunge_strategy                                                  (* m1 *)
ABSTRACT SUPERTYPE OF (ONEOF (plunge_toolaxis, plunge_ramp, plunge_helix,
  plunge_zigzag))
SUBTYPE OF (approach_retract_strategy);
END_ENTITY;

ENTITY plunge_toolaxis                                                  (* m1 *)
SUBTYPE OF (plunge_strategy);
END_ENTITY;

ENTITY plunge_ramp                                                      (* m1 *)
SUBTYPE OF (plunge_strategy);
angle: plane_angle_measure;
END_ENTITY;

ENTITY plunge_helix                                                     (* m1 *)
SUBTYPE OF (plunge_strategy);
radius  :   length_measure;
angle   :   plane_angle_measure;
END_ENTITY;

ENTITY plunge_zigzag                                                    (* m1 *)
```

```
SUBTYPE OF (plunge_strategy);
angle: plane_angle_measure;
width: length_measure;
END_ENTITY;

ENTITY air_strategy                                                       (* m1 *)
ABSTRACT SUPERTYPE OF (ONEOF (ap_retract_angle, ap_retract_tangent))
SUBTYPE OF (approach_retract_strategy);
END_ENTITY;

ENTITY ap_retract_angle                                                   (* m1 *)
SUBTYPE OF (air_strategy);
angle:        plane_angle_measure;
travel_length: length_measure;
END_ENTITY;

ENTITY ap_retract_tangent                                                 (* m1 *)
SUBTYPE OF (air_strategy);
radius: length_measure;
END_ENTITY;

ENTITY along_path                                                         (* m1 *)
SUBTYPE OF (approach_retract_strategy);
path: toolpath_list;
END_ENTITY;

(* ********************************************************** *)
(* Freeform operation                                         *)
(* ********************************************************** *)

ENTITY freeform_operation                                                 (* m0 *)
SUBTYPE OF (milling_type_operation);
its_machining_strategy: OPTIONAL freeform_strategy;
END_ENTITY;

(* ********************************************************** *)
(* Freeform strategy                                          *)
(* ********************************************************** *)

ENTITY freeform_strategy                                                  (* m1 *)
ABSTRACT SUPERTYPE OF (ONEOF(uv_strategy, plane_cc_strategy, plane_cl_strategy,
  leading_line_strategy));
pathmode:              pathmode_type;
cutmode:               cutmode_type;
its_milling_tolerances: tolerances;
stepover:              OPTIONAL length_measure;
END_ENTITY;

TYPE pathmode_type = ENUMERATION OF (
forward,
zigzag
);
END_TYPE;

TYPE cutmode_type = ENUMERATION OF (
climb,
conventional
);
END_TYPE;

ENTITY tolerances;                                                        (* m1 *)
```



```
chordal_tolerance : length_measure;
scallop_height    : length_measure;
END_ENTITY;

ENTITY uv_strategy                                                      (* m1 *)
SUBTYPE OF (freeform_strategy);
forward_direction:      direction;
sideward_direction:     direction;
END_ENTITY;

ENTITY plane_cc_strategy                                                (* m1 *)
SUBTYPE OF (freeform_strategy);
its_plane_normal: direction;
END_ENTITY;

ENTITY plane_cl_strategy                                                (* m1 *)
SUBTYPE OF (freeform_strategy);
its_plane_normal: direction;
END_ENTITY;

ENTITY leading_line_strategy                                            (* m1 *)
SUBTYPE OF (freeform_strategy);
its_line : bounded_curve;
END_ENTITY;

 (* ************************************************************ *)
 (* two5D milling operation                                      *)
 (* ************************************************************ *)

ENTITY two5D_milling_operation                                          (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(plane_milling, side_milling,
  bottom_and_side_milling))
SUBTYPE OF (milling_type_operation);
its_machining_strategy: OPTIONAL two5D_milling_strategy;
END_ENTITY;

 (* ************************************************************ *)
 (* two5D_milling_strategy                                       *)
 (* ************************************************************ *)

ENTITY two5D_milling_strategy                                           (* m1 *)
ABSTRACT SUPERTYPE OF (ONEOF (unidirectional, bidirectional, contour_parallel,
  bidirectional_contour, contour_bidirectional, contour_spiral, center_milling,
  explicit_strategy));
overlap:                OPTIONAL positive_ratio_measure;
allow_multiple_passes:  OPTIONAL BOOLEAN;
END_ENTITY;

ENTITY unidirectional                                                   (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
feed_direction:        OPTIONAL direction;
cutmode:                OPTIONAL cutmode_type;
END_ENTITY;

ENTITY bidirectional                                                    (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
feed_direction:        OPTIONAL direction;
stepover_direction:       OPTIONAL left_or_right;
its_stroke_connection_strategy: OPTIONAL stroke_connection_strategy;
END_ENTITY;
```

```
TYPE left_or_right = ENUMERATION OF (left, right);
END_TYPE;

TYPE stroke_connection_strategy = ENUMERATION OF
  (straghtline, lift_shift_plunge, degouge, loop_back);
END_TYPE;

ENTITY contour_parallel                                                 (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
rotation_direction:     OPTIONAL rot_direction;
cutmode:                OPTIONAL cutmode_type;
END_ENTITY;

ENTITY bidirectional_contour                                            (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
feed_direction:        OPTIONAL direction;
stepover_direction:       OPTIONAL left_or_right;
rotation_direction:     OPTIONAL rot_direction;
spiral_cutmode:         OPTIONAL cutmode_type;
END_ENTITY;

ENTITY contour_bidirectional                                            (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
feed_direction:        OPTIONAL direction;
stepover_direction:       OPTIONAL left_or_right;
rotation_direction:     OPTIONAL rot_direction;
spiral_cutmode:         OPTIONAL cutmode_type;
END_ENTITY;

ENTITY contour_spiral                                                   (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
rotation_direction:     OPTIONAL rot_direction;
cutmode:                OPTIONAL cutmode_type;
END_ENTITY;

ENTITY center_milling                                                   (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
END_ENTITY;

ENTITY explicit_strategy                                                (* m1 *)
SUBTYPE OF (two5D_milling_strategy);
END_ENTITY;

(* ********************************************************** *)
(* plane_milling, side_milling, bottom_and_side_milling        *)
(* ********************************************************** *)

ENTITY plane_milling                                                    (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(plane_rough_milling, plane_finish_milling))
SUBTYPE OF (two5D_milling_operation);
axial_cutting_depth:        OPTIONAL length_measure;
allowance_bottom:           OPTIONAL length_measure;
END_ENTITY;

ENTITY plane_rough_milling                                              (* m0 *)
SUBTYPE OF (plane_milling);
WHERE
WR1: EXISTS(SELF.allowance_bottom) AND (SELF.allowance_bottom>=0.0);
END_ENTITY;

ENTITY plane_finish_milling                                             (* m0 *)
```

```
SUBTYPE OF (plane_milling);
END_ENTITY;

ENTITY side_milling                                                      (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(side_rough_milling, side_finish_milling))
SUBTYPE OF (two5D_milling_operation);
axial_cutting_depth:    OPTIONAL length_measure;
radial_cutting_depth:   OPTIONAL length_measure;
allowance_side:         OPTIONAL length_measure;
END_ENTITY;

ENTITY side_rough_milling                                                (* m0 *)
SUBTYPE OF (side_milling);
WHERE
WR1: EXISTS(SELF.allowance_side) AND (SELF.allowance_side>=0.0);
END_ENTITY;

ENTITY side_finish_milling                                               (* m0 *)
SUBTYPE OF (side_milling);
END_ENTITY;

ENTITY bottom_and_side_milling                                           (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(bottom_and_side_rough_milling,
  bottom_and_side_finish_milling))
SUBTYPE OF (two5D_milling_operation);
axial_cutting_depth:    OPTIONAL length_measure;
radial_cutting_depth:   OPTIONAL length_measure;
allowance_side:         OPTIONAL length_measure;
allowance_bottom:       OPTIONAL length_measure;END_ENTITY;

ENTITY bottom_and_side_rough_milling                                     (* m0 *)
SUBTYPE OF (bottom_and_side_milling);
WHERE
  WR1: EXISTS(SELF.allowance_side) AND (SELF.allowance_side>=0.0);
  WR2: EXISTS(SELF.allowance_bottom) AND (SELF.allowance_bottom>=0.0);
END_ENTITY;

ENTITY bottom_and_side_finish_milling                                    (* m0 *)
SUBTYPE OF (bottom_and_side_milling);
END_ENTITY;

(* ********************************************************** *)
(* Drilling type operation                                     *)
(* ********************************************************** *)

ENTITY drilling_type_operation                                           (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(drilling_operation, boring_operation, back_boring,
  tapping, thread_drilling))
SUBTYPE OF (milling_machining_operation);
cutting_depth:              OPTIONAL length_measure;
previous_diameter:          OPTIONAL length_measure;
dwell_time_bottom:          OPTIONAL time_measure;
feed_on_retract:            OPTIONAL positive_ratio_measure;
its_machining_strategy:     OPTIONAL drilling_type_strategy;
END_ENTITY;

(* ********************************************************** *)
(* Drilling type strategy                                      *)
(* ********************************************************** *)

ENTITY drilling_type_strategy;                                           (* m0 *)
```

```
reduced_cut_at_start:       OPTIONAL positive_ratio_measure;
reduced_feed_at_start:      OPTIONAL positive_ratio_measure;
depth_of_start:             OPTIONAL length_measure;
reduced_cut_at_end:         OPTIONAL positive_ratio_measure;
reduced_feed_at_end:        OPTIONAL positive_ratio_measure;
depth_of_end:               OPTIONAL length_measure;
WHERE
WR1: EXISTS(depth_of_start) OR NOT (EXISTS(reduced_cut_at_start) OR
  EXISTS(reduced_feed_at_start));
WR2: EXISTS(depth_of_end) OR NOT (EXISTS(reduced_cut_at_end) OR
  EXISTS(reduced_feed_at_end));
END_ENTITY;

(* ********************************************************** *)
(* Drilling operation                                         *)
(* ********************************************************** *)

ENTITY drilling_operation                                              (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(drilling, center_drilling, counter_sinking,
  multistep_drilling))
SUBTYPE OF (drilling_type_operation);
END_ENTITY;

ENTITY drilling                                                        (* m0 *)
SUBTYPE OF (drilling_operation);
END_ENTITY;

ENTITY center_drilling                                                 (* m0 *)
SUBTYPE OF (drilling_operation);
END_ENTITY;

ENTITY counter_sinking                                                 (* m0 *)
SUBTYPE OF (drilling_operation);
END_ENTITY;

ENTITY multistep_drilling                                              (* m0 *)
SUBTYPE OF (drilling_operation);
retract_distance:       length_measure;
first_depth:            length_measure;
depth_of_step:          length_measure;
dwell_time_step:        OPTIONAL time_measure;
END_ENTITY;

(* ********************************************************** *)
(* Boring operation                                           *)
(* ********************************************************** *)

ENTITY boring_operation                                                (* m0 *)
ABSTRACT SUPERTYPE OF (ONEOF(boring, reaming))
SUBTYPE OF (drilling_type_operation);
spindle_stop_at_bottom:         BOOLEAN;
depth_of_testcut:               OPTIONAL length_measure;
waiting_position:               OPTIONAL cartesian_point;
END_ENTITY;

ENTITY boring                                                          (* m0 *)
SUBTYPE OF (boring_operation);
END_ENTITY;

ENTITY reaming                                                         (* m0 *)
 SUBTYPE OF (boring_operation);
```



```
END_ENTITY;

ENTITY back_boring                                                            (* m0 *)
SUBTYPE OF (drilling_type_operation);
WHERE
 WR1:  EXISTS(SELF.its_machine_functions.oriented_spindle_stop);
END_ENTITY;

ENTITY tapping                                                                (* m0 *)
SUBTYPE OF (drilling_type_operation);
compensation_chuck:             BOOLEAN;
END_ENTITY;

ENTITY thread_drilling                                                        (* m0 *)
SUBTYPE OF (drilling_type_operation);
helical_movement_on_forward:    BOOLEAN;
END_ENTITY;

END_SCHEMA; (* milling_schema *)
```

Annex B
(normative)

Short names of entities

Table 1 — Short names of entities

Entity names	Short names
ADAPTIVE_CONTROL	ADPCNT
AIR_STRATEGY	ARSTR
ALONG_PATH	ALNPTH
AP_RETRACT_ANGLE	APRTAN
AP_RETRACT_TANGENT	APRTTN
APPROACH_RETRACT_STRATEGY	APRTST
BACK_BORING	BCKBRN
BACKSIDE_COUNTERBORE	BCK0
BACKSIDE_COUNTERSINK	BCKCNT
BALL_ENDMILL	BLLEND
BIDIRECTIONAL	BDRCTN
BIDIRECTIONAL_CONTOUR	BDRCNT
BORING	BORING
BORING_OPERATION	BRNOPR
BORING_TOOL	BRNTL
BOTTOM_AND_SIDE_FINISH_MILLING	BASFM
BOTTOM_AND_SIDE_MILLING	BASM
BOTTOM_AND_SIDE_ROUGH_MILLING	BASRM
BULLNOSE_ENDMILL	BLL0
CENTER_DRILLING	CNTDRL
CENTER_MILLING	CNTMLL
CENTERDRILL	CNTRDR
COMBINED_DRILL_AND_REAMER	CDA0
COMBINED_DRILL_AND_TAP	CDAT
CONTOUR_BIDIRECTIONAL	CNTBDR
CONTOUR_PARALLEL	CNTPRL
CONTOUR_SPIRAL	CNTSPR

Entity names	Short names
COUNTER_SINKING	CNTSNK
COUNTERBORE	CNTRBR
COUNTERSINK	CNTRSN
DOVETAIL_MILL	DVTMLL
DRILL	DRILL
DRILLING	DRLLNG
DRILLING_OPERATION	DRLOPR
DRILLING_TYPE_OPERATION	DRTYOP
DRILLING_TYPE_STRATEGY	DRTYST
ENDMILL	ENDMLL
EXCHANGE_PALLET	EXGPLL
EXPLICIT_STRATEGY	EXPSTR
FACEMILL	FCMLL
FIVE_AXES_CONST_TILT_YAW	FACTY
FIVE_AXES_VAR_TILT_YAW	FAVTY
FREEFORM_OPERATION	FRFOPR
FREEFORM_STRATEGY	FRFSTR
INDEX_PALLET	INDPLL
INDEX_TABLE	INDTBL
LEADING_LINE_STRATEGY	LDLNST
LOAD_TOOL	LDTL
MILLING_CUTTER	MLLCTT
MILLING_CUTTING_TOOL	MLCTTL
MILLING_MACHINE_FUNCTIONS	MLMCFN
MILLING_MACHINING_OPERATION	MLMCOP
MILLING_TECHNOLOGY	MLLTCH
MILLING_TOOL_BODY	MLTLBD
MILLING_TYPE_OPERATION	MLTYOP
MULTISTEP_DRILLING	MLTDRL
PLANE_CC_STRATEGY	PLCCST
PLANE_CL_STRATEGY	PLCLST
PLANE_FINISH_MILLING	PLFNML

Entity names	Short names
PLANE_MILLING	PLNMLL
PLANE_ROUGH_MILLING	PLRGML
PLUNGE_HELIX	PLNHLX
PLUNGE_RAMP	PLNRMP
PLUNGE_STRATEGY	PLNSTR
PLUNGE_TOOLAXIS	PLNTLX
PLUNGE_ZIGZAG	PLNZGZ
PROCESS_MODEL	PRCMDL
PROCESS_MODEL_LIST	PRMDLS
REAMER	REAMER
REAMING	RMNG
SIDE_FINISH_MILLING	SDFNML
SIDE_MILL	SDM0
SIDE_MILLING	SDMLL
SIDE_ROUGH_MILLING	SDRGML
SPADE_DRILL	SPDDRL
T_SLOT_MILL	TSLML
TAP	TAP
TAPERED_DRILL	TPRDRL
TAPERED_ENDMILL	TPREND
TAPERED_REAMER	TPRRMR
TAPERED_TAP	TPRTP
TAPPING	TPPNG
THREAD_DRILLING	THRDRL
THREAD_MILL	THRMLL
THREADING_TOOL	THRTL
THREE_AXES_TILTED_TOOL	TATT
TOLERANCES	TLRNCS
TOOL_DIMENSION	TLDMN
TOOL_DIRECTION_FOR_MILLING	TDFM
TWIST_DRILL	TWSDRL
TWO5D_MILLING_OPERATION	TWMLOP

Entity names	Short names
TWO5D_MILLING_STRATEGY	TWMLST
UNIDIRECTIONAL	UNDRCT
UNLOAD_TOOL	UNLTL
USER_DEFINED_TOOL	USDFTL
UV_STRATEGY	UVSTR
WOODRUFF_KEYSEAT_MILL	WDKYML

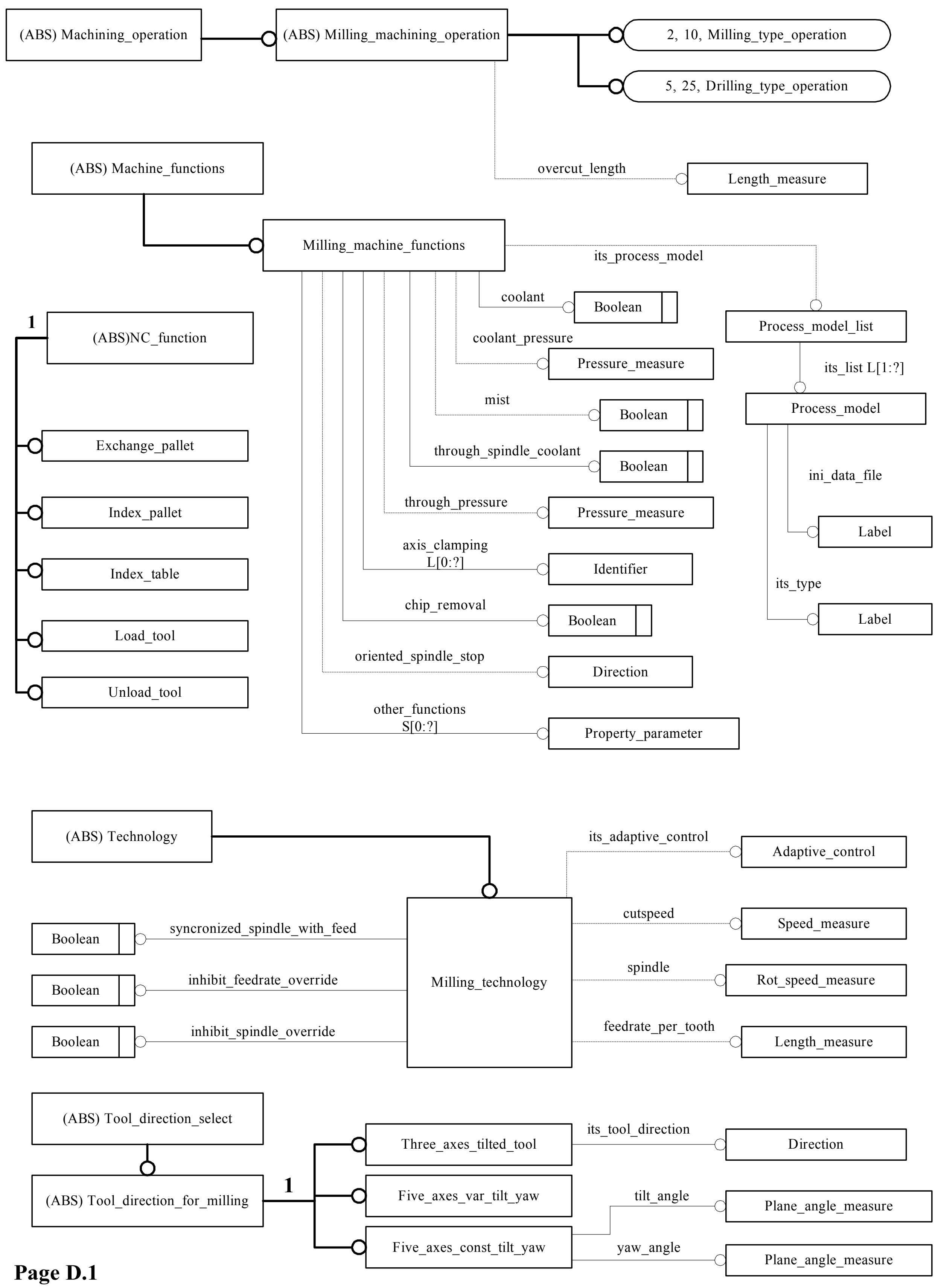
Annex C
(normative)

Implementation method specific requirements

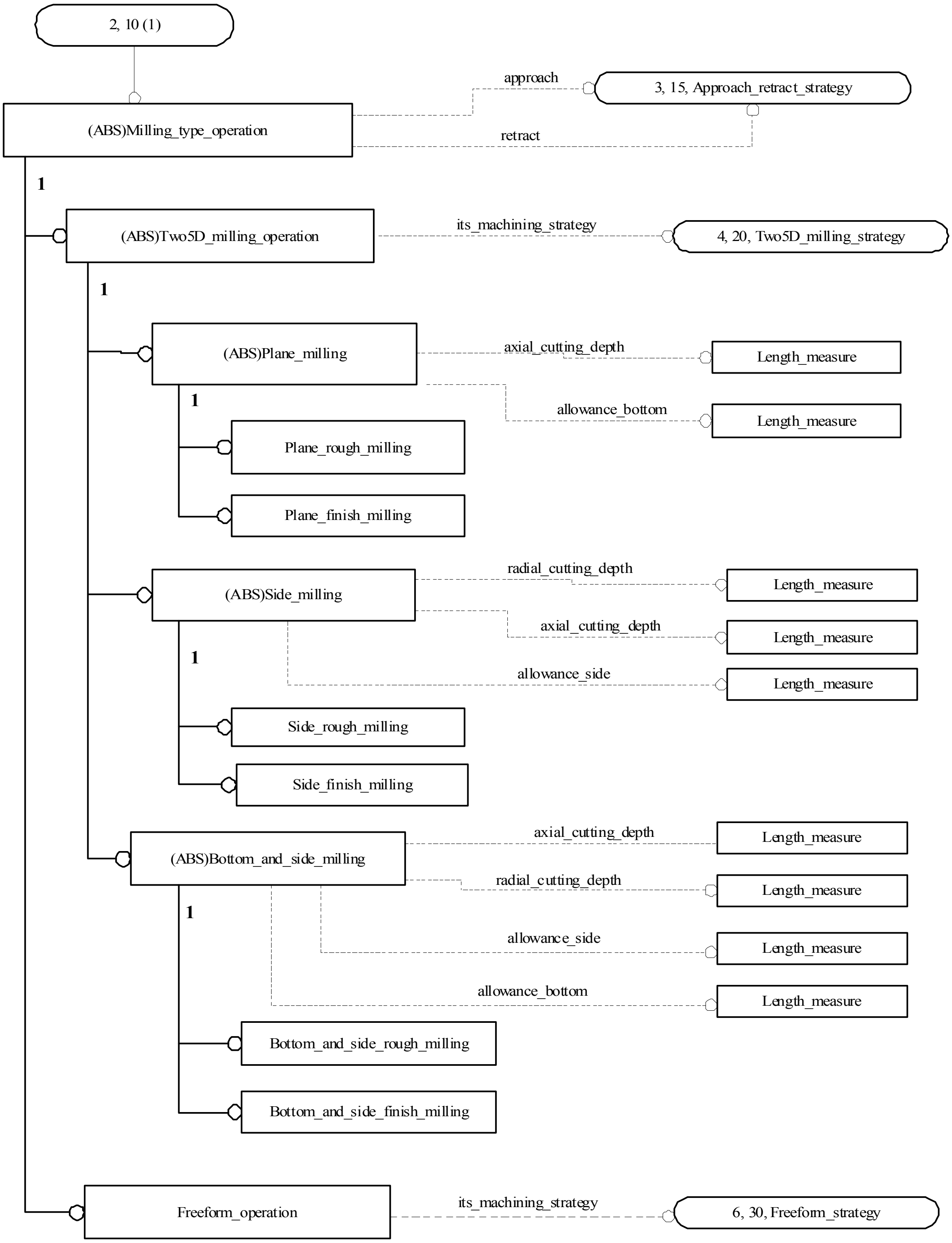
The implementation method defines what type of exchange behavior is required to this part of ISO 14649.
Conformance to this part of ISO 14649 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and annotated listing defined in Annex A of this part of ISO 14649. The header of the exchange structure shall identify use of this part of ISO 14649 by the schema name 'milling_schema'.

Annex D
(informative)

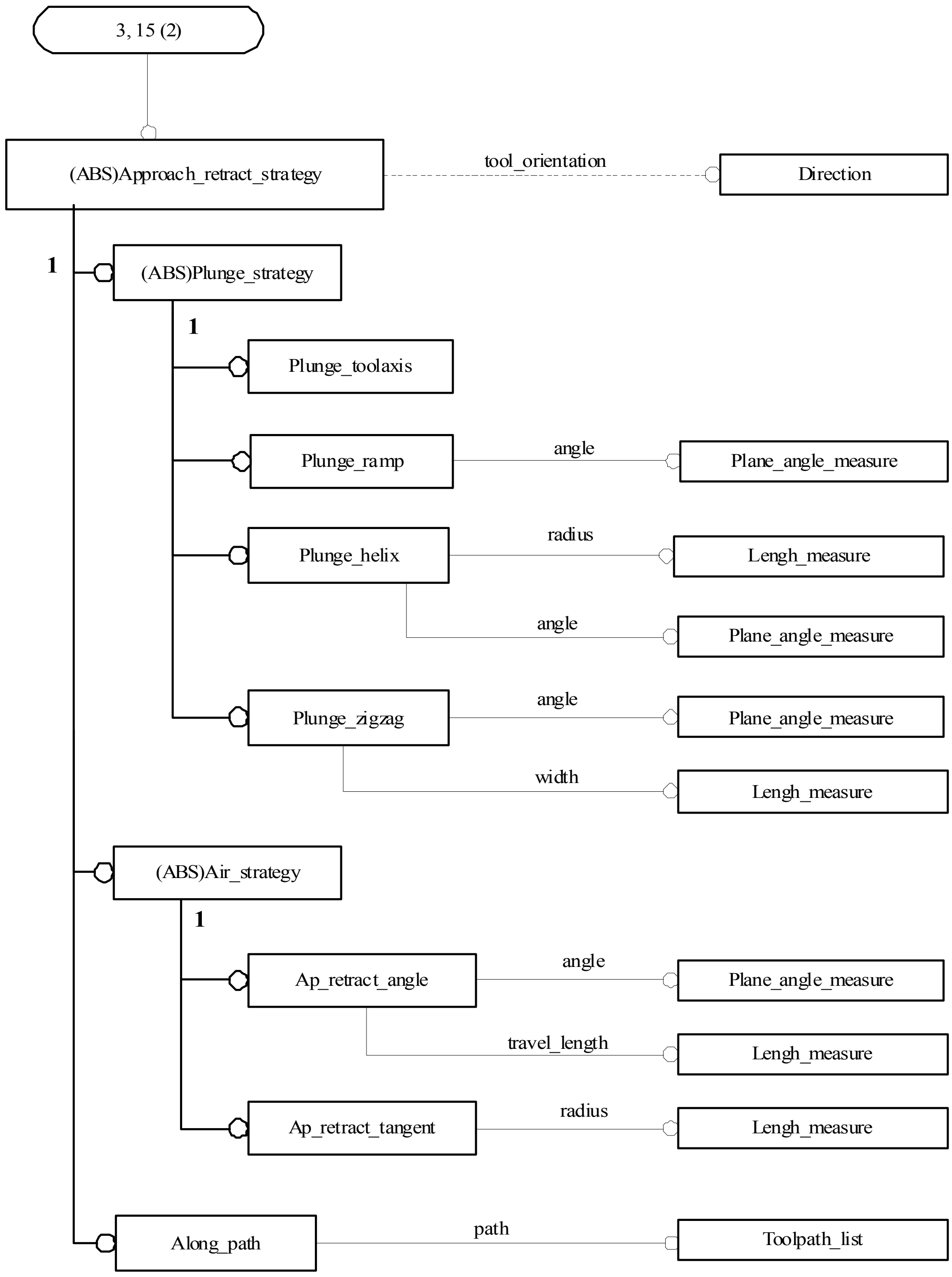
EXPRESS-G diagram


(ABS) Machining_operation
(ABS) Milling_machining_operation
2, 10, Milling_type_operation
5, 25, Drilling_type_operation
overcut_length
Length_measure
(ABS) Machine_functions
Milling_machine_functions
its_process_model
coolant
Boolean
Process_model_list
coolant_pressure
Pressure_measure
its_list L[1:?]
mist
Boolean
Process_model
through_spindle_coolant
Boolean
ini_data_file
through_pressure
Pressure_measure
Label
axis_clamping
L[0:?]
Identifier
its_type
chip_removal
Boolean
Label
oriented_spindle_stop
Direction
other_functions
S[0:?]
Property_parameter
1
(ABS)NC_function
Exchange_pallet
Index_pallet
Index_table
Load_tool
Unload_tool
(ABS) Technology
its_adaptive_control
Adaptive_control
cutspeed
Speed_measure
Boolean
syncronized_spindle_with_feed
spindle
Rot_speed_measure
Boolean
inhibit_feedrate_override
Milling_technology
feedrate_per_tooth
Length_measure
Boolean
inhibit_spindle_override
(ABS) Tool_direction_select
Three_axes_tilted_tool
its_tool_direction
Direction
(ABS) Tool_direction_for_milling
1
Five_axes_var_tilt_yaw
tilt_angle
Plane_angle_measure
Five_axes_const_tilt_yaw
yaw_angle
Plane_angle_measure

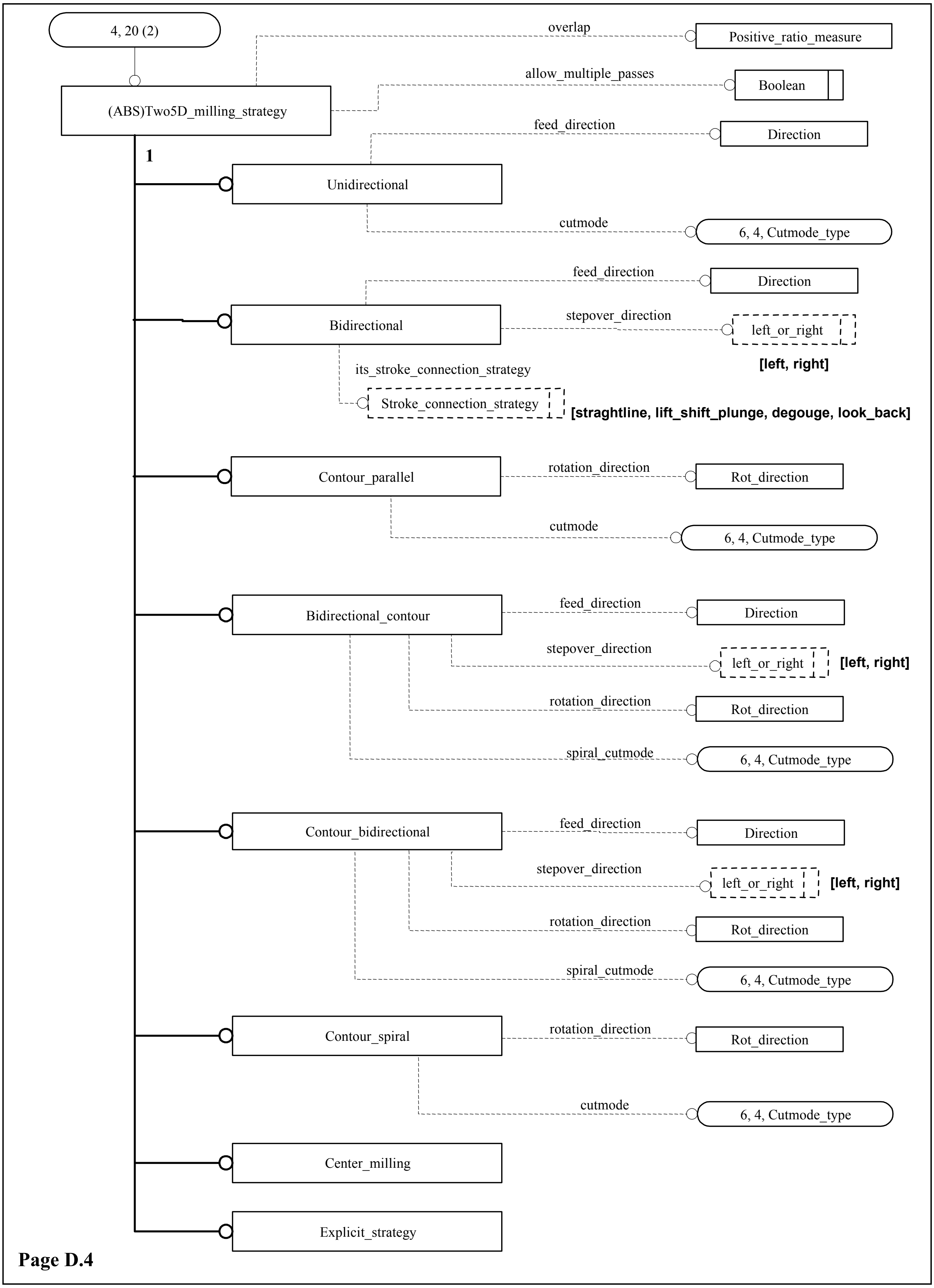
Page D.1

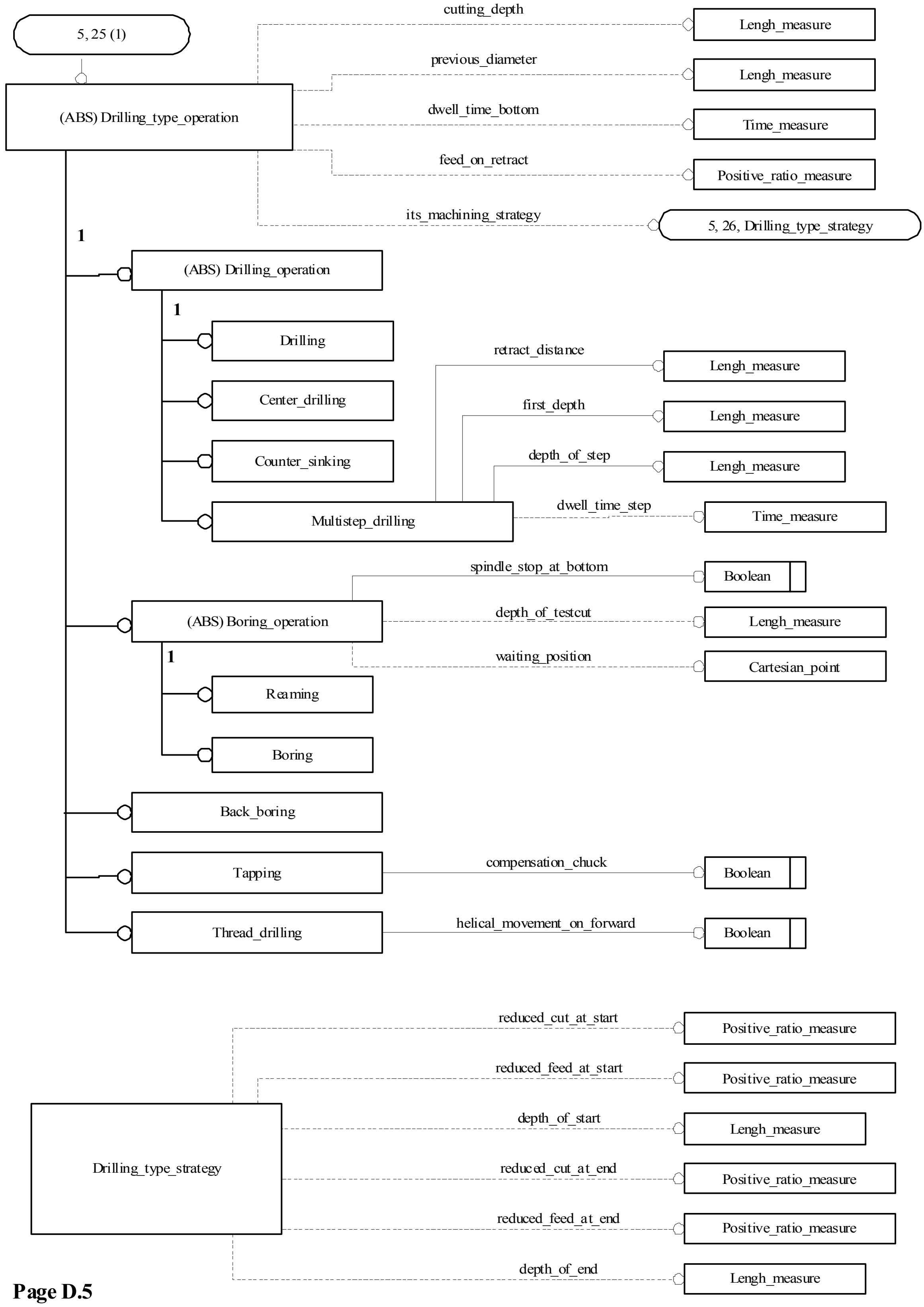

2, 10 (1)
(ABS)Milling_type_operation
approach
retract
3, 15, Approach_retract_strategy
1
(ABS)Two5D_milling_operation
its_machining_strategy
4, 20, Two5D_milling_strategy
1
(ABS)Plane_milling
axial_cutting_depth
Length_measure
allowance_bottom
Length_measure
1
Plane_rough_milling
Plane_finish_milling
(ABS)Side_milling
radial_cutting_depth
Length_measure
axial_cutting_depth
Length_measure
allowance_side
Length_measure
1
Side_rough_milling
Side_finish_milling
(ABS)Bottom_and_side_milling
axial_cutting_depth
Length_measure
radial_cutting_depth
Length_measure
allowance_side
Length_measure
allowance_bottom
Length_measure
1
Bottom_and_side_rough_milling
Bottom_and_side_finish_milling
Freeform_operation
its_machining_strategy
6, 30, Freeform_strategy

Page D.2

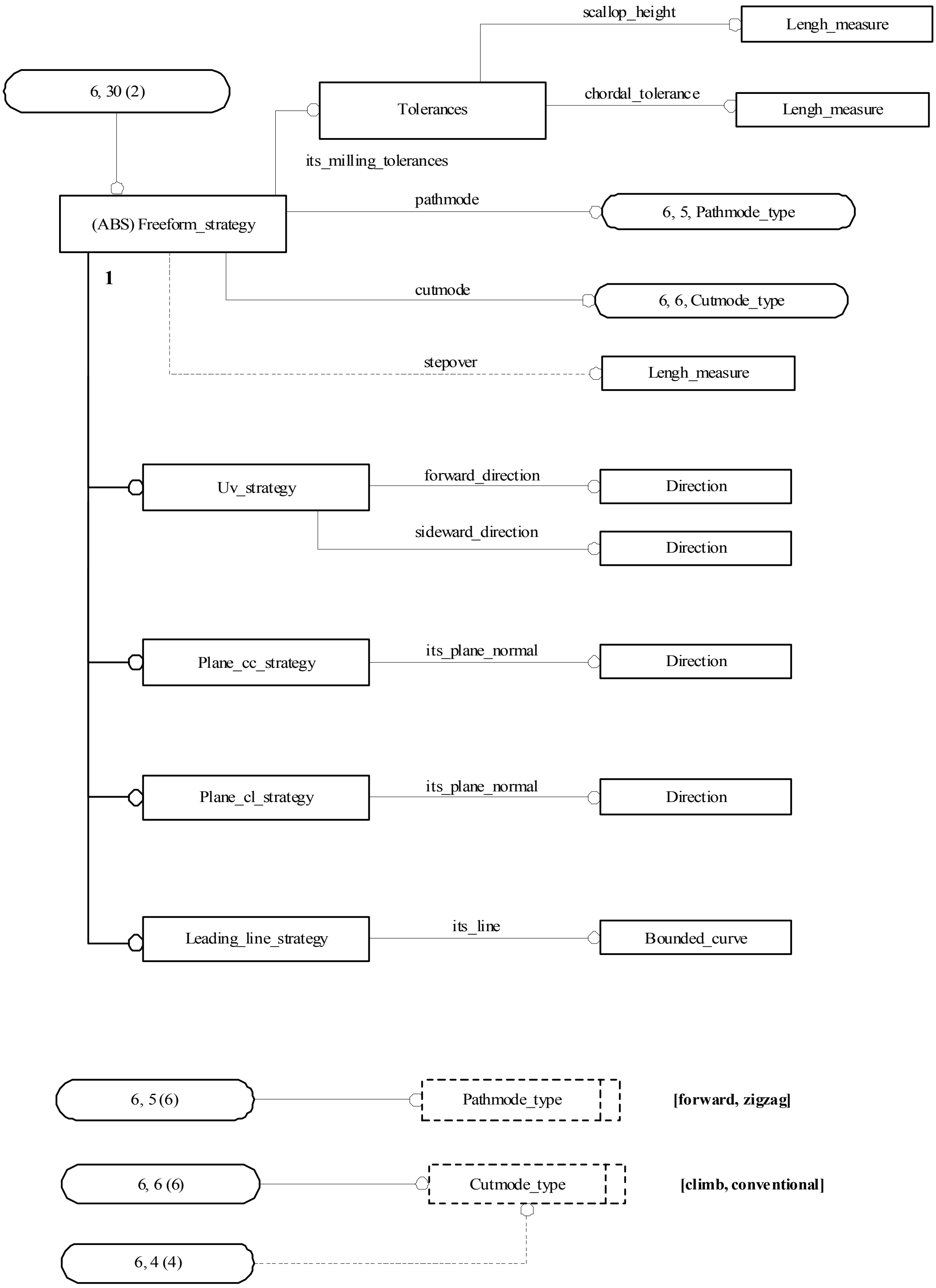

3, 15 (2)
(ABS)Approach_retract_strategy
tool_orientation
Direction
1
(ABS)Plunge_strategy
1
Plunge_toolaxis
Plunge_ramp
angle
Plane_angle_measure
Plunge_helix
radius
Lengh_measure
angle
Plane_angle_measure
Plunge_zigzag
angle
Plane_angle_measure
width
Lengh_measure
(ABS)Air_strategy
1
Ap_retract_angle
angle
Plane_angle_measure
travel_length
Lengh_measure
Ap_retract_tangent
radius
Lengh_measure
Along_path
path
Toolpath_list

Page D.3


4, 20 (2)
(ABS)Two5D_milling_strategy
overlap
Positive_ratio_measure
allow_multiple_passes
Boolean
feed_direction
Direction
1
Unidirectional
cutmode
6, 4, Cutmode_type
feed_direction
Direction
Bidirectional
stepover_direction
left_or_right
[left, right]
its_stroke_connection_strategy
Stroke_connection_strategy
[straghtline, lift_shift_plunge, degouge, look_back]
Contour_parallel
rotation_direction
Rot_direction
cutmode
6, 4, Cutmode_type
Bidirectional_contour
feed_direction
Direction
stepover_direction
left_or_right
[left, right]
rotation_direction
Rot_direction
spiral_cutmode
6, 4, Cutmode_type
Contour_bidirectional
feed_direction
Direction
stepover_direction
left_or_right
[left, right]
rotation_direction
Rot_direction
spiral_cutmode
6, 4, Cutmode_type
Contour_spiral
rotation_direction
Rot_direction
cutmode
6, 4, Cutmode_type
Center_milling
Explicit_strategy
Page D.4


5, 25 (1)
(ABS) Drilling_type_operation
cutting_depth
Lengh_measure
previous_diameter
Lengh_measure
dwell_time_bottom
Time_measure
feed_on_retract
Positive_ratio_measure
its_machining_strategy
5, 26, Drilling_type_strategy
1
(ABS) Drilling_operation
1
Drilling
Center_drilling
Counter_sinking
Multistep_drilling
retract_distance
Lengh_measure
first_depth
Lengh_measure
depth_of_step
Lengh_measure
dwell_time_step
Time_measure
spindle_stop_at_bottom
Boolean
(ABS) Boring_operation
depth_of_testcut
Lengh_measure
waiting_position
Cartesian_point
1
Reaming
Boring
Back_boring
Tapping
compensation_chuck
Boolean
Thread_drilling
helical_movement_on_forward
Boolean
Drilling_type_strategy
reduced_cut_at_start
Positive_ratio_measure
reduced_feed_at_start
Positive_ratio_measure
depth_of_start
Lengh_measure
reduced_cut_at_end
Positive_ratio_measure
reduced_feed_at_end
Positive_ratio_measure
depth_of_end
Lengh_measure

Page D.5


scallop_height
Lengh_measure
6, 30 (2)
Tolerances
chordal_tolerance
Lengh_measure
its_milling_tolerances
pathmode
(ABS) Freeform_strategy
6, 5, Pathmode_type
1
cutmode
6, 6, Cutmode_type
stepover
Lengh_measure
Uv_strategy
forward_direction
Direction
sideward_direction
Direction
Plane_cc_strategy
its_plane_normal
Direction
Plane_cl_strategy
its_plane_normal
Direction
Leading_line_strategy
its_line
Bounded_curve
6, 5 (6)
Pathmode_type
[forward, zigzag]
6, 6 (6)
Cutmode_type
[climb, conventional]
6, 4 (4)

Page D.6

Annex E
(informative)

Computer-interpretable listings

This annex provides a listing of the short names with their corresponding entity names and a listing of the EXPRESS specified in this Part of ISO 14649. No text or annotation is included. This annex is provided only in computer-interpretable form.

Annex F
(informative)

Sample NC programmes

F.1 Example 1

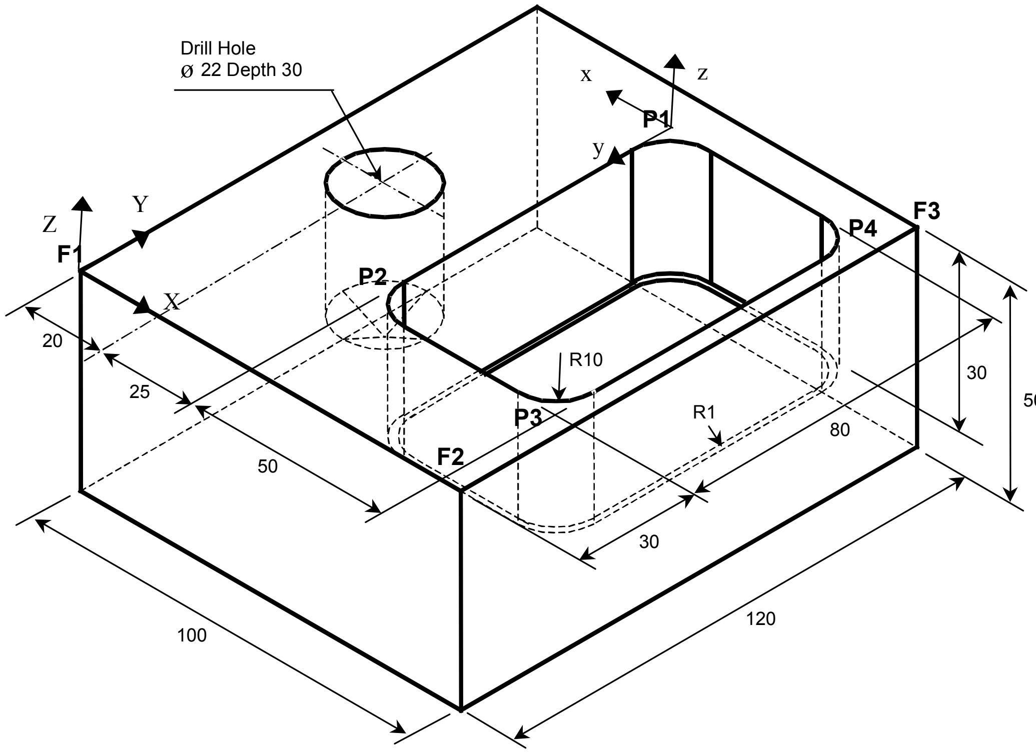


Fig. 25: Workpiece of the Example 1

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ISO 14649-11 EXAMPLE 1',
         'SIMPLE PRORGRAM WITH A PLANAR_FACE, A POCKET, AND A ROUND_HOLE'),
         '1');
FILE_NAME('EXAMPLE1.STP',
         '2002-02-02',
         ('YONG TAK HYUN','JOCHEN WOLF'),
         ('WZL, RWTH-AACHEN'),
         $,
         'ISO 14649',
         $);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;

DATA;
#1= PROJECT('EXECUTE EXAMPLE1',#2,(#4),$,$,$);
#2= WORKPLAN('MAIN WORKPLAN',(#10,#11,#12,#13,#14),$,#8,$);
#4= WORKPIECE('SIMPLE WORKPIECE',#6,0.010,$,$,$,(#66,#67,#68,#69));
#6= MATERIAL('ST-50','STEEL',(#7));
```

```
#7= PROPERTY_PARAMETER('E=200000N/M2');
#8= SETUP('SETUP1',#71,#62,(#9));
#9= WORKPIECE_SETUP(#4,#74,$,$,());
#10= MACHINING_WORKINGSTEP('WS FINISH PLANAR FACE1',#62,#16,#19,$);
#11= MACHINING_WORKINGSTEP('WS DRILL HOLE1',#62,#17,#20,$);
#12= MACHINING_WORKINGSTEP('WS REAM HOLE1',#62,#17,#21,$);
#13= MACHINING_WORKINGSTEP('WS ROUGH POCKET1',#62,#18,#22,$);
#14= MACHINING_WORKINGSTEP('WS FINISH POCKET1',#62,#18,#23,$);
#16= PLANAR_FACE('PLANAR FACE1',#4,(#19),#77,#63,#24,#25,$,());
#17= ROUND_HOLE('HOLE1 D=22MM',#4,(#20,#21),#81,#64,#58,$,#26);
#18= CLOSED_POCKET('POCKET1',#4,(#22,#23),#84,#65,(),$,#27,#35,#37,#28);
#19= PLANE_FINISH_MILLING($,$,'FINISH PLANAR FACE1',10.000,$,#39,#40,#41,$,
#60,#61,#42,2.500,$);
#20= DRILLING($,$,'DRILL HOLE1',10.000,$,#44,#45,#41,$,$,$,$,$,#46);
#21= REAMING($,$,'REAM HOLE1',10.000,$,#47,#48,#41,$,$,$,$,$,#49,.T.,$,$);
#22= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH POCKET1',15.000,$,#39,#50,#41
,$,$,$,#51,2.500,5.000,1.000,0.500);
#23= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH POCKET1',15.000,$,#39,#52,
#41,$,$,$,#53,2.000,10.000,$,$);
#24= LINEAR_PATH($,#54,#55);
#25= LINEAR_PROFILE($,#57);
#26= THROUGH_BOTTOM_CONDITION();
#27= PLANAR_POCKET_BOTTOM_CONDITION();
#28= GENERAL_CLOSED_PROFILE($,#59);
#29= TAPERED_ENDMILL(#30,4,$,.F.,$,$);
#30= TOOL_DIMENSION(20.000,$,$,$,1.500,$,$);
#31= TWIST_DRILL(#32,2,.RIGHT.,.F.,0.840);
#32= TOOL_DIMENSION(20.000,31.000,0.100,45.000,2.000,5.000,8.000);
#33= TAPERED_REAMER(#34,6,$,.F.,$,$);
#34= TOOL_DIMENSION(22.000,$,$,$,$,$,$);
#35= TOLERANCED_LENGTH_MEASURE(1.000,#36);
#36= PLUS_MINUS_VALUE(0.100,0.100,3);
#37= TOLERANCED_LENGTH_MEASURE(10.000,#38);
#38= PLUS_MINUS_VALUE(0.100,0.100,3);
#39= MILLING_CUTTING_TOOL('MILL 20MM',#29,(#125),80.000,$,$);
#40= MILLING_TECHNOLOGY(0.040,.TCP.,$,12.000,$,.F.,.F.,.F.,$);
#41= MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,());
#42= BIDIRECTIONAL(5.000,.T.,#43,.LEFT.,$);
#43= DIRECTION('STRATEGY PLANAR FACE1: 1.DIRECTION',(0.000,1.000,0.000));
#44= MILLING_CUTTING_TOOL('SPIRAL_DRILL_20MM',#31,(#126),90.000,$,$);
#45= MILLING_TECHNOLOGY(0.030,.TCP.,$,16.000,$,.F.,.F.,.F.,$);
#46= DRILLING_TYPE_STRATEGY(75.000,50.000,2.000,50.000,75.000,8.000);
#47= MILLING_CUTTING_TOOL('REAMER_22MM',#33,(#127),100.000,$,$);
#48= MILLING_TECHNOLOGY(0.030,.TCP.,$,18.000,$,.F.,.F.,.F.,$);
#49= DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#50= MILLING_TECHNOLOGY($,.TCP.,$,20.000,$,.F.,.F.,.F.,$);
#51= CONTOUR_BIDIRECTIONAL($,$,$,$,$,$);
#52= MILLING_TECHNOLOGY($,.TCP.,$,20.000,$,.F.,.F.,.F.,$);
#53= CONTOUR_PARALLEL(5.000,.T.,.CW.,.CONVENTIONAL.);
#54= TOLERANCED_LENGTH_MEASURE(120.000,#56);
#55= DIRECTION('COURSE OF TRAVEL DIRECTION',(0.000,1.000,0.000));
#56= PLUS_MINUS_VALUE(0.300,0.300,3);
#57= NUMERIC_PARAMETER('PROFILE LENGTH',100.000,'MM');
#58= TOLERANCED_LENGTH_MEASURE(22.000,#56);
#59= POLYLINE('CONTOUR OF POCKET1',(#121,#122,#123,#124,#121));
#60= PLUNGE_RAMP($,45.000);
#61= PLUNGE_RAMP($,45.000);
```

```
#62= ELEMENTARY_SURFACE('SECURITY PLANE',#73);
#63= ELEMENTARY_SURFACE('PLANAR FACE1-DEPTH PLANE',#80);
#64= ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE1',#83);
#65= ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET1',#94);
#66= CARTESIAN_POINT('CLAMPING_POSITION1',(0.000,20.000,25.000));
#67= CARTESIAN_POINT('CLAMPING_POSITION2',(100.000,20.000,25.000));
#68= CARTESIAN_POINT('CLAMPING_POSITION3',(0.000,100.000,25.000));
#69= CARTESIAN_POINT('CLAMPING_POSITION4',(100.000,100.000,25.000));
#71= AXIS2_PLACEMENT_3D('SETUP1',#95,#96,#97);
#73= AXIS2_PLACEMENT_3D('PLANE1',#98,#99,#100);
#74= AXIS2_PLACEMENT_3D('WORKPIECE',#101,#102,#103);
#77= AXIS2_PLACEMENT_3D('PLANAR FACE1',#104,#105,#106);
#80= AXIS2_PLACEMENT_3D('PLANAR FACE1',#107,#108,#109);
#81= AXIS2_PLACEMENT_3D('HOLE1',#110,#111,$);
#83= AXIS2_PLACEMENT_3D('HOLE1',#112,#113,#114);
#84= AXIS2_PLACEMENT_3D('POCKET1',#115,#116,#117);
#94= AXIS2_PLACEMENT_3D('POCKET1',#118,#119,#120);
#95= CARTESIAN_POINT('SETUP1:    LOCATION     ',(150.000,90.000,40.000));
#96= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#97= DIRECTION('                 REF_DIRECTION',(1.000,0.000,0.000));
#98= CARTESIAN_POINT('SECPLANE1: LOCATION     ',(0.000,0.000,30.000));
#99= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#100= DIRECTION('               REF_DIRECTION',(1.000,0.000,0.000));
#101= CARTESIAN_POINT('WORKPIECE1:LOCATION     ',(0.000,0.000,0.000));
#102= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#103= DIRECTION('                 REF_DIRECTION',(1.000,0.000,0.000));
#104= CARTESIAN_POINT('PLANAR FACE1:LOCATION   ',(0.000,0.000,5.000));
#105= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#106= DIRECTION('                 REF_DIRECTION',(1.000,0.000,0.000));
#107= CARTESIAN_POINT('PLANAR FACE1:DEPTH      ',(0.000,0.000,-5.000));
#108= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#109= DIRECTION('                 REF_DIRECTION',(1.000,0.000,0.000));
#110= CARTESIAN_POINT('HOLE1:      LOCATION     ',(20.000,60.000,0.000));
#111= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#112= CARTESIAN_POINT('HOLE1:      DEPTH        ',(0.000,0.000,-30.000));
#113= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#114= DIRECTION('                 REF_DIRECTION',(1.000,0.000,0.000));
#115= CARTESIAN_POINT('POCKET1:    LOCATION     ',(45.000,110.000,0.000));
#116= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#117= DIRECTION('                 REF_DIRECTION',(-1.000,0.000,0.000));
#118= CARTESIAN_POINT('POCKET1:    DEPTH     ',(0.000,0.000,-30.000));
#119= DIRECTION('                 AXIS         ',(0.000,0.000,1.000));
#120= DIRECTION('                 REF_DIRECTION',(1.000,0.000,0.000));
#121= CARTESIAN_POINT('P1',(0.000,0.000,0.000));
#122= CARTESIAN_POINT('P2',(0.000,80.000,0.000));
#123= CARTESIAN_POINT('P3',(-50.000,80.000,0.000));
#124= CARTESIAN_POINT('P4',(-50.000,0.000,0.000));
#125= CUTTING_COMPONENT(80.000,$,$,$);
#126= CUTTING_COMPONENT(90.000,$,$,$);
#127= CUTTING_COMPONENT(100.000,$,$,$);
ENDSEC;

END-ISO-10303-21;
```

F.2 Example 2

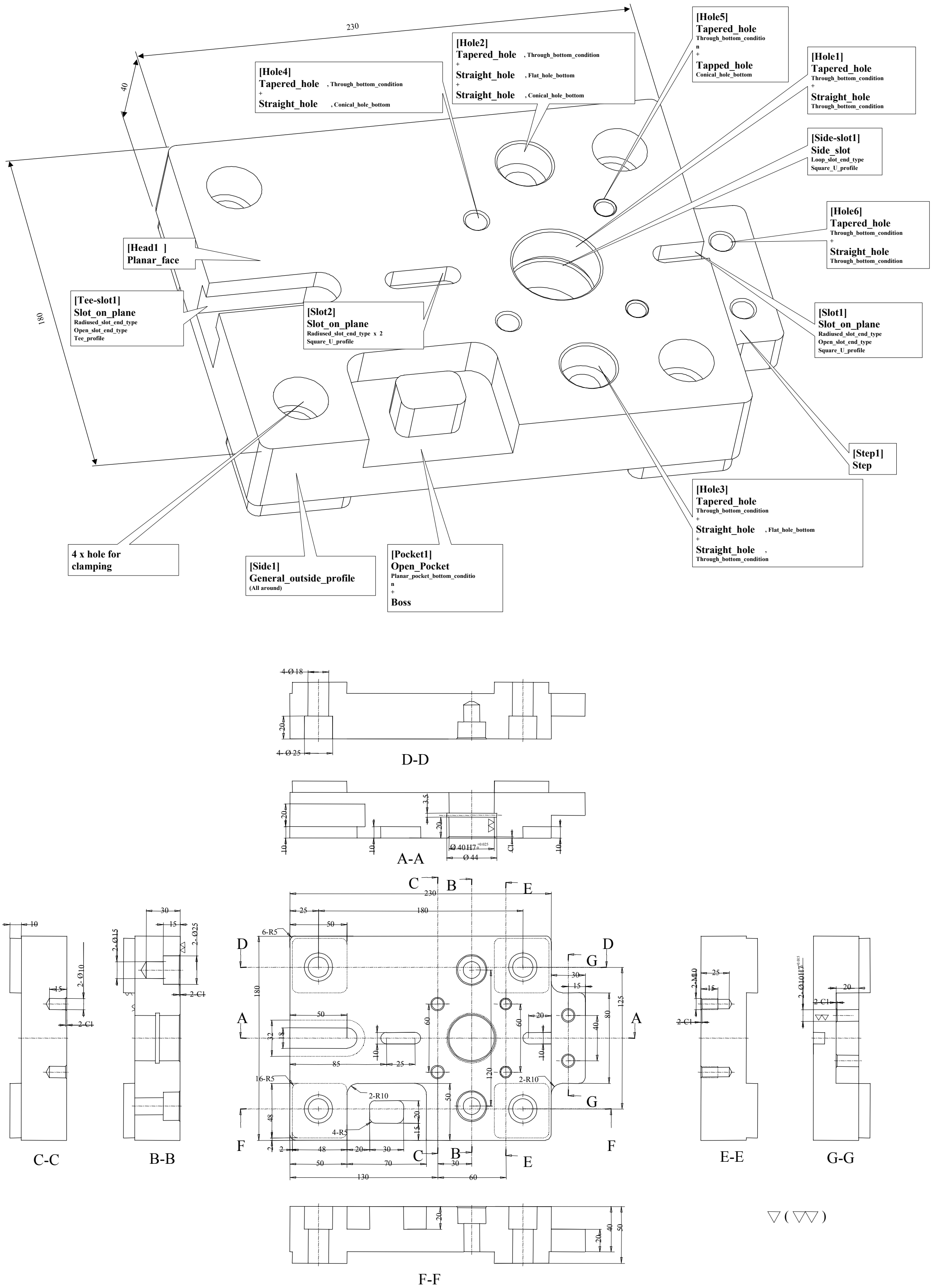


Fig. 26: Workpiece of the Example 2

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ISO 14649-11 EXAMPLE 2',
             'COMPLEX PRORGRAM WITH VARIOUS MANUFACTURING FEATURES'),'1');
FILE_NAME('EXAMPLE2.STP',
        '2002-02-02',
        ('JOCHEN WOLF','YONG TAK HYUN','C.SAKAMOTO'),
        ('WZL, RWTH-AACHEN','KOMATSU'),
        $,
        'ISO 14649',
        $);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;

DATA;
#1= PROJECT('EXECUTE_EXAMPLE2',#2,(#7),$,$,$);
#2= WORKPLAN('MAIN WORKPLAN',(#4,#5,#6),$,#14,$);
#4= WORKPLAN('WORKPLAN ROUGHING',(#17,#18,#19,#20,#21),$,$,$);
#5= WORKPLAN('WORKPLAN
DRILLING',(#23,#24,#25,#26,#27,#28,#29,#30,#31,#32,#33,#34,#35,#36,#37),$,$,$);
#6= WORKPLAN('WORKPLAN
FINISHING',(#39,#40,#41,#42,#43,#44,#45,#46,#47,#48,#49,#50,#51,#52,#53),$,$,$);
#7= WORKPIECE('PART 2',#13,0.01,$,$,$,(#9,#10,#11,#12));
#9= CARTESIAN_POINT('CLAMPING_POSITION1',(25.,25.,-20.));
#10= CARTESIAN_POINT('CLAMPING_POSITION2',(205.,25.,-20.));
#11= CARTESIAN_POINT('CLAMPING_POSITION3',(25.,155.,-20.));
#12= CARTESIAN_POINT('CLAMPING_POSITION4',(205.,155.,-20.));
#13= MATERIAL('FC200','CAST IRON ',());
#14= SETUP('SETUP OF WORKPIECE',#342,#333,(#16));
#16= WORKPIECE_SETUP(#7,#344,$,$,());
#17= MACHINING_WORKINGSTEP('ROUGHING HEAD1',#333,#60,#61,$);
#18= MACHINING_WORKINGSTEP('ROUGHING SIDE1',#333,#62,#63,$);
#19= MACHINING_WORKINGSTEP('ROUGHING STEP',#333,#64,#65,$);
#20= MACHINING_WORKINGSTEP('ROUGHING POCKET1',#333,#66,#67,$);
#21= MACHINING_WORKINGSTEP('FINISH POCKET1',#333,#66,#68,$);
#23= MACHINING_WORKINGSTEP('BORING1 HOLE1',#333,#100,#73,$);
#24= MACHINING_WORKINGSTEP('DRILLING HOLE2',#333,#101,#76,$);
#25= MACHINING_WORKINGSTEP('DRILLING HOLE3',#333,#102,#77,$);
#26= MACHINING_WORKINGSTEP('BORING HOLE2',#333,#101,#78,$);
#27= MACHINING_WORKINGSTEP('BORING HOLE3',#333,#102,#79,$);
#28= MACHINING_WORKINGSTEP('COUNTER SINKING HOLE2',#333,#101,#80,$);
#29= MACHINING_WORKINGSTEP('COUNTER SINKING HOLE3',#333,#102,#81,$);
#30= MACHINING_WORKINGSTEP('DRILLING HOLE4',#333,#103,#82,$);
#31= MACHINING_WORKINGSTEP('DRILLING HOLE5',#333,#104,#83,$);
#32= MACHINING_WORKINGSTEP('TAPPING HOLE5',#333,#104,#84,$);
#33= MACHINING_WORKINGSTEP('COUNTER SINKING HOLE4',#333,#103,#85,$);
#34= MACHINING_WORKINGSTEP('COUNTER SINKING HOLE5',#333,#104,#86,$);
#35= MACHINING_WORKINGSTEP('DRILLING HOLE6',#333,#105,#87,$);
#36= MACHINING_WORKINGSTEP('REAMING HOLE6',#333,#105,#88,$);
#37= MACHINING_WORKINGSTEP('COUNTER SINKING HOLE6',#333,#105,#89,$);
#39= MACHINING_WORKINGSTEP('FINISH SIDE-SLOT     ',#333,#123,#90,$);
#40= MACHINING_WORKINGSTEP('BORING HOLE1         ',#333,#100,#74,$);
#41= MACHINING_WORKINGSTEP('COUNTER SINKING HOLE1',#333,#100,#75,$);
#42= MACHINING_WORKINGSTEP('ROUGHING1 T-SLOT     ',#333,#124,#91,$);
#43= MACHINING_WORKINGSTEP('ROUGHING2 T-SLOT     ',#333,#124,#92,$);
```

```
#44= MACHINING_WORKINGSTEP('FINISH1 T-SLOT       ',#333,#124,#93,$);
#45= MACHINING_WORKINGSTEP('FINISH2 T-SLOT       ',#333,#124,#94,$);
#46= MACHINING_WORKINGSTEP('ROUGHING SLOT1       ',#333,#125,#95,$);
#47= MACHINING_WORKINGSTEP('SLOT2_PLUNGE_DRILLING',#333,#126,#96,$);
#48= MACHINING_WORKINGSTEP('ROUGHING SLOT2       ',#333,#126,#97,$);
#49= MACHINING_WORKINGSTEP('FINISH SLOT1       ',#333,#125,#98,$);
#50= MACHINING_WORKINGSTEP('FINISH SLOT2         ',#333,#126,#99,$);
#51= MACHINING_WORKINGSTEP('FINISH STEP          ',#333,#64,#72,$);
#52= MACHINING_WORKINGSTEP('FINISH HEAD1         ',#333,#60,#69,$);
#53= MACHINING_WORKINGSTEP('FINISH SIDE1         ',#333,#62,#70,$);
#60= PLANAR_FACE('HEAD1                                  ',#7,(#61,#69),#348,#334,#178,#179,$,());
#61= PLANE_ROUGH_MILLING($,$,'ROUGHING HEAD1            ',10.,$,#226,#229,#230,$,$,$,#184,$,$);
#62= GENERAL_OUTSIDE_PROFILE('SIDE1                     ',#7,(#63,#70),#352,#335,#185,#186);
#63= SIDE_ROUGH_MILLING($,$,'ROUGH SIDE1
',10.,$,#231,#234,#235,$,$,$,#191,4.,15.,0.5);
#64= STEP('STEP1',#7,(#71,#72),#355,#336,#192,$,());
#65= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING STEP1
',65.,$,#231,#234,#235,$,#196,#197,#198,10.,18.,1.5,2.);
#66= OPEN_POCKET('POCKET1',#7,(#67,#68),#359,#337,(#199),$,#202,$,$,#203,$);
#67= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING
POCKET1',10.,$,#237,#241,#235,$,$,$,#205,5.,10.,0.5,0.5);
#68= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH POCKET1
',10.,$,#237,#241,#235,$,$,$,#206,10.,5.,$,$);
#69= PLANE_FINISH_MILLING($,$,'FINISH HEAD1
',10.,$,#226,#229,#230,$,#207,#208,#209,2.5,$);
#70= SIDE_FINISH_MILLING($,$,'FINISH SIDE1              ',10.,$,#231,#234,#230,$,$,$,#211,1.5,1.,$);
#71= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING STEP1
',65.,$,#231,#234,#235,$,#196,#197,#198,10.,18.,1.5,2.);
#72= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH STEP1
',65.,$,#231,#234,#235,$,#196,#197,#198,5.,12.5,$,$);
#73= MULTISTEP_DRILLING($,$,'BORING1 HOLE1     ',20.,$,#242,#246,#230,$,48.,0.,$,$,$,5.,15.,10.,1.);
#74= BORING($,$,'BORING2 HOLE1                ',20.,$,#252,#256,#230,$,45.,36.,$,$,$,.T.,1.,$);
#75= COUNTER_SINKING($,$,'COUNTER SINKING HOLE1',30.,$,#257,#261,#230,$,1.,40.,$,$,$);
#76= DRILLING($,$,'DRILLING HOLE2             ',20.,$,#262,#251,#230,$,$,0.,$,$,$);
#77= DRILLING($,$,'DRILLING HOLE3             ',20.,$,#262,#251,#230,$,$,0.,$,$,$);
#78= BORING($,$,'BORING HOLE2                 ',20.,$,#266,#270,#230,$,15.,15.,$,$,$,.T.,1.,$);
#79= BORING($,$,'BORING HOLE3                 ',20.,$,#266,#270,#230,$,15.,15.,$,$,$,.T.,1.,$);
#80= COUNTER_SINKING($,$,'COUNTER SINKING HOLE2',30.,$,#271,#261,#230,$,1.,25.,$,$,$);
#81= COUNTER_SINKING($,$,'COUNTER SINKING HOLE3',30.,$,#271,#261,#230,$,1.,25.,$,$,$);
#82= DRILLING($,$,'DRILLING HOLE4             ',20.,$,#275,#251,#230,$,$,0.,$,$,$);
#83= MULTISTEP_DRILLING($,$,'DRILLING HOLE5    ',20.,$,#247,#251,#230,$,$,0.,$,$,$,5.,10.,10.,$);
#84= TAPPING($,$,'TAPPING HOLE5               ',10.,$,#279,#283,#230,$,15.,8.3,$,$,$,.T.);
#85= COUNTER_SINKING($,$,'COUNTER SINKING HOLE4',30.,$,#284,#261,#230,$,1.,10.,$,$,$);
#86= COUNTER_SINKING($,$,'COUNTER SINKING HOLE5',30.,$,#284,#261,#230,$,1.,10.,$,$,$);
#87= DRILLING($,$,'DRILLING HOLE6             ',10.,$,#288,#251,#230,$,22.,0.,$,$,$);
#88= REAMING($,$,'REAMING HOLE6               ',10.,$,#292,#296,#230,$,21.,9.,$,$,$,.F.,$,$);
#89= COUNTER_SINKING($,$,'COUTNER SINKING HOLE6',30.,$,#284,#261,#230,$,1.,10.,$,$,$);
#90= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'SIDE SLOT
',40.,$,#297,#301,#230,$,#219,#220,#221,$,$,$,$);
#91= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING1 T-SLOT
',10.,$,#302,#306,#230,$,$,#197,$,5.,$,1.,0.5);
#92= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING2 T-SLOT
',10.,$,#307,#311,#230,$,$,$,$,11.,$,1.,0.5);
#93= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISHING1 T-SLOT ',10.,$,#312,#236,#230,$,$,$,$,$,$,$,$);
#94= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISHING2 T-SLOT ',10.,$,#313,#317,#230,$,$,$,$,$,$,$,$);
#95= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING SLOT1
',10.,$,#318,#322,#230,$,$,#197,#222,$,$,1.,0.5);
```

```
#96= DRILLING($,$,'DRILLING SLOT2',10.,$,#288,#251,#230,$,5.,0.,$,$,$);
#97= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGHING SLOT2
',10.,$,#318,#322,#230,$,#196,#197,#223,$,$,1.,0.5);
#98= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISHING SLOT1
',10.,$,#323,#327,#230,$,#196,#197,#224,5.,5.,$,$);
#99= BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH SLOT2
',10.,$,#323,#327,#230,$,#196,#197,#225,5.,6.,$,$);
#100= COMPOUND_FEATURE('COMPOUND FEATURE HOLE1',#7,(),#363,(#106,#107));
#101= COMPOUND_FEATURE('COMPOUND_FEATURE_HOLE2',#7,(),#363,(#108,#109,#110));
#102= COMPOUND_FEATURE('COMPOUND_FEATURE_HOLE3',#7,(),#370,(#111,#112,#110));
#103= GENERAL_PATTERN('PATTERN_HOLE4',#7,(),#374,#113,(#377,#376));
#104= GENERAL_PATTERN('PATTERN_HOLE5',#7,(),#380,#116,(#383,#382));
#105= GENERAL_PATTERN('PATTERN_HOLE6',#7,(),#386,#120,(#389,#388));
#106= ROUND_HOLE('HOLE1 STRAIGHT              ',#7,(#73,#74),#368,#538,#539,$,#215);
#107= ROUND_HOLE('HOLE1 TAPERED               ',#7,(#75),#368,#539,#542,#212,#215);
#108= ROUND_HOLE('HOLE2 FLAT BOTTOM           ',#7,(#78),#369,#545,#546,$,#214);
#109= ROUND_HOLE('HOLE2 CONICAL BOTTOM        ',#7,(#76),#369,#548,#549,$,#213);
#110= ROUND_HOLE('HOLE2 TAPERED               ',#7,(#80),#369,#551,#552,#212,#216);
#111= ROUND_HOLE('HOLE3 FLAT BOTTOM           ',#7,(#79),#373,#555,#556,$,#214);
#112= ROUND_HOLE('HOLE2 THROUGH BOTTOM       ',#7,(#77),#373,#558,#559,$,#215);
#113= COMPOUND_FEATURE('COMPOUND_FEATURE_HOLE4',#7,(),#379,(#114,#115));
#114= ROUND_HOLE('HOLE4 CONICAL BOTTOM        ',#7,(#82),#379,#566,#567,$,#213);
#115= ROUND_HOLE('HOLE4 TAPERED               ',#7,(#85),#379,#569,#570,#212,#214);
#116= COMPOUND_FEATURE('COMPOUND_FEATURE_HOLE5',#7,(),#385,(#117,#118,#119));
#117= ROUND_HOLE('HOLE5 CONICAL BOTTOM        ',#7,(#83),#385,#575,#576,$,#213);
#118= ROUND_HOLE('TAPM10                      ',#7,(#84),#385,#578,#579,$,#214);
#119= ROUND_HOLE('HOLE5 TAPERED               ',#7,(#86),#385,#582,#583,#212,#214);
#120= COMPOUND_FEATURE('COMPOUND_FEATURE_HOLE6',#7,(),#391,(#121,#122));
#121= ROUND_HOLE('HOLE6 CONICAL BOTTOM        ',#7,(#87,#88),#391,#588,#589,$,#213);
#122= ROUND_HOLE('HOLE6 TAPERED               ',#7,(#89),#391,#591,#592,#212,#214);
#123= SLOT('SIDE-SLOT1',#7,(#90),#395,#338,#127,#132,(#218));
#124= SLOT('TEE_SLOT1 ',#7,(#91,#92,#93,#94),#394,#339,#137,#143,(#216,#217));
#125= SLOT('SLOT1     ',#7,(#95,#68),#392,#340,#156,#162,(#216,#217));
#126= SLOT('SLOT2     ',#7,(#96,#97,#99),#393,#341,#167,#173,(#217,#217));
#127= COMPLETE_CIRCULAR_PATH(#128,#130);
#128= AXIS2_PLACEMENT_3D('SIDE1:COURSE OF TRAVEL',#129,$,$);
#129= CARTESIAN_POINT('SIDE1:PLACEMENT',(0.,0.,-20.));
#130= TOLERANCED_LENGTH_MEASURE(44.,#131);
#131= PLUS_MINUS_VALUE(0.1,0.1,3);
#132= SQUARE_U_PROFILE(#400,#133,#135,0.,#135,0.);
#133= TOLERANCED_LENGTH_MEASURE(3.5,#134);
#134= PLUS_MINUS_VALUE(0.1,0.1,3);
#135= TOLERANCED_LENGTH_MEASURE(0.1,#136);
#136= PLUS_MINUS_VALUE(0.02,0.02,3);
#137= GENERAL_PATH(#138,#140);
#138= AXIS2_PLACEMENT_3D('TEE1:COURSE OF TRAVEL',#139,$,$);
#139= CARTESIAN_POINT('TEE1:PLACEMENT',(0.,0.,-10.));
#140= POLYLINE('TEE_SLOT1',(#141,#142));
#141= CARTESIAN_POINT('TEE_SLOT1',(0.,0.,0.));
#142= CARTESIAN_POINT('TEE_SLOT1',(50.,0.,0.));
#143= TEE_PROFILE($,60.,60.,#144,#146,#148,#150,#152,#154);
#144= TOLERANCED_LENGTH_MEASURE(32.,#145);
#145= PLUS_MINUS_VALUE(0.1,0.1,3);
#146= TOLERANCED_LENGTH_MEASURE(20.,#147);
#147= PLUS_MINUS_VALUE(0.1,0.1,3);
#148= TOLERANCED_LENGTH_MEASURE(0.2,#149);
#149= PLUS_MINUS_VALUE(0.01,0.01,3);
```

```
#150= TOLERANCED_LENGTH_MEASURE(20.,#151);
#151= PLUS_MINUS_VALUE(0.1,0.1,3);
#152= TOLERANCED_LENGTH_MEASURE(0.1,#153);
#153= PLUS_MINUS_VALUE(0.02,0.02,3);
#154= TOLERANCED_LENGTH_MEASURE(0.1,#155);
#155= PLUS_MINUS_VALUE(0.02,0.02,3);
#156= GENERAL_PATH(#157,#159);
#157= AXIS2_PLACEMENT_3D('SLOT1:COURSE OF TRAVEL',#158,$,$);
#158= CARTESIAN_POINT('SLOT1:PLACEMENT',(0.,0.,0.));
#159= POLYLINE('SLOT1',(#160,#161));
#160= CARTESIAN_POINT('SLOT1',(0.,0.,0.));
#161= CARTESIAN_POINT('SLOT1',(-20.,0.,0.));
#162= SQUARE_U_PROFILE($,#163,#165,0.,#165,0.);
#163= TOLERANCED_LENGTH_MEASURE(10.,#164);
#164= PLUS_MINUS_VALUE(0.1,0.1,3);
#165= TOLERANCED_LENGTH_MEASURE(0.1,#166);
#166= PLUS_MINUS_VALUE(0.02,0.02,3);
#167= GENERAL_PATH(#168,#170);
#168= AXIS2_PLACEMENT_3D('SLOT2:COURSE OF TRAVEL',#169,$,$);
#169= CARTESIAN_POINT('SLOT2:PLACEMENT',(0.,0.,0.));
#170= POLYLINE('SLOT2',(#171,#172));
#171= CARTESIAN_POINT('SLOT2',(0.,0.,0.));
#172= CARTESIAN_POINT('SLOT2',(25.,0.,0.));
#173= SQUARE_U_PROFILE($,#174,#176,0.,#176,0.);
#174= TOLERANCED_LENGTH_MEASURE(10.,#134);
#175= PLUS_MINUS_VALUE(0.1,0.1,3);
#176= TOLERANCED_LENGTH_MEASURE(0.1,#136);
#177= PLUS_MINUS_VALUE(0.02,0.02,3);
#178= LINEAR_PATH($,#180,#181);
#179= LINEAR_PROFILE($,#183);
#180= TOLERANCED_LENGTH_MEASURE(180.,#182);
#181= DIRECTION('COURSE OF TRAVEL DIRECTION',(0.,1.,0.));
#182= PLUS_MINUS_VALUE(0.3,0.3,3);
#183= NUMERIC_PARAMETER('PROFILE LENGTH',260.,'MM');
#184= UNIDIRECTIONAL(0.2,.T.,#483,.CONVENTIONAL.);
#185= LINEAR_PATH($,#187,#188);
#186= LINEAR_PROFILE($,#190);
#187= TOLERANCED_LENGTH_MEASURE(-50.,#189);
#188= DIRECTION('COURSE OF TRAVEL DIRECTION',(0.,0.,-1.));
#189= PLUS_MINUS_VALUE(0.3,0.3,3);
#190= NUMERIC_PARAMETER('PROFILE LENGTH',230.,'MM');
#191= CONTOUR_PARALLEL($,.T.,.CW.,.CLIMB.);
#192= LINEAR_PATH($,#193,#194);
#193= TOLERANCED_LENGTH_MEASURE(100.,#195);
#194= DIRECTION('COURSE OF TRAVEL DIRECTION',(0.,1.,0.));
#195= PLUS_MINUS_VALUE(0.3,0.3,3);
#196= PLUNGE_TOOLAXIS($);
#197= PLUNGE_TOOLAXIS($);
#198= BIDIRECTIONAL(5.,.T.,$,$,$);
#199= BOSS('POCKET1-BOSS1',#7,(),#359,#337,#200,$);
#200= GENERAL_CLOSED_PROFILE($,#201);
#201= COMPOSITE_CURVE('BOSS BOUNDARY POCKET1',(#484,#487,#494,#497,#502,#505,#510,#513),.F.);
#202= PLANAR_POCKET_BOTTOM_CONDITION();
#203= GENERAL_PROFILE($,#204);
#204= COMPOSITE_CURVE('OPEN BOUNDARY POCKET1',(#518,#521,#526,#529,#534),.F.);
#205= BIDIRECTIONAL_CONTOUR($,.T.,$,$,$,.CONVENTIONAL.);
#206= BIDIRECTIONAL_CONTOUR($,.T.,$,$,$,.CONVENTIONAL.);
```

```
#207= PLUNGE_RAMP($,45.);
#208= PLUNGE_RAMP($,45.);
#209= BIDIRECTIONAL(0.2,.T.,#210,$,$);
#210= DIRECTION('OPEN POCKET:FEED DIRECTION',(0.,1.,0.));
#211= CONTOUR_PARALLEL($,.T.,.CW.,.CLIMB.);
#212= ANGLE_TAPER(45.);
#213= CONICAL_HOLE_BOTTOM(30.,$);
#214= FLAT_HOLE_BOTTOM();
#215= THROUGH_BOTTOM_CONDITION();
#216= OPEN_SLOT_END_TYPE();
#217= RADIUSED_SLOT_END_TYPE();
#218= LOOP_SLOT_END_TYPE();
#219= PLUNGE_HELIX($,25.,45.);
#220= PLUNGE_HELIX($,25.,45.);
#221= CONTOUR_PARALLEL($,$,$,.CLIMB.);
#222= CENTER_MILLING(5.,$);
#223= CENTER_MILLING(5.,$);
#224= CENTER_MILLING(4.,$);
#225= CENTER_MILLING(4.,$);
#226= MILLING_CUTTING_TOOL('MILL 36MM',#228,(#227),75.,#482,32.);
#227= CUTTING_COMPONENT(20.,#328,$,1230.,#229);
#228= FACEMILL(#329,8,.RIGHT.,.F.,$);
#229= MILLING_TECHNOLOGY(0.5,.TCP.,14.,$,0.5,.T.,.F.,.F.,$);
#230= MILLING_MACHINE_FUNCTIONS(.F.,$,$,.F.,$,(),.T.,$,$,());
#231= MILLING_CUTTING_TOOL('ENDMILL 25',#233,(#232),72.,#482,20.);
#232= CUTTING_COMPONENT(30.,#330,$,1200.,#236);
#233= ENDMILL(#331,4,.RIGHT.,.F.,$);
#234= MILLING_TECHNOLOGY($,.CCP.,5.5,40.,$,.T.,.F.,.F.,$);
#235= MILLING_MACHINE_FUNCTIONS(.F.,$,$,.F.,$,(),.T.,$,$,());
#236= MILLING_TECHNOLOGY($,.TCP.,$,50.,0.2,.T.,.F.,.F.,$);
#237= MILLING_CUTTING_TOOL('ENDMILL 12',#239,(#238),54.,#482,8.);
#238= CUTTING_COMPONENT(30.,#332,$,1000.,#241);
#239= ENDMILL(#240,3,.RIGHT.,.F.,$);
#240= TOOL_DIMENSION(12.,$,$,$,$,$,$);
#241= MILLING_TECHNOLOGY(0.024,.TCP.,$,120.,$,.T.,.F.,.F.,$);
#242= MILLING_CUTTING_TOOL('BORING TOOL 36',#244,(#243),200.,#482,32.);
#243= CUTTING_COMPONENT(30.,#332,$,1200.,#246);
#244= BORING_TOOL(#245,3,.RIGHT.,.T.,$,.F.);
#245= TOOL_DIMENSION(36.,11.,0.,4.3,0.3,$,$);
#246= MILLING_TECHNOLOGY($,.TCP.,2.3,$,0.1,.T.,.F.,.F.,$);
#247= MILLING_CUTTING_TOOL('DRILL 8',#249,(#248),70.,#482,8.);
#248= CUTTING_COMPONENT(30.,#332,$,1230.,#251);
#249= TWIST_DRILL(#250,8,.RIGHT.,$,$);
#250= TOOL_DIMENSION(8.,31.,$,$,$,$,$);
#251= MILLING_TECHNOLOGY(0.08,.TCP.,1.6,$,$,.T.,.F.,.F.,$);
#252= MILLING_CUTTING_TOOL('SPADE DRILL 40',#254,(#253),160.,#482,32.);
#253= CUTTING_COMPONENT(30.,#328,$,1200.,#256);
#254= SPADE_DRILL(#255,4,.RIGHT.,.F.,$);
#255= TOOL_DIMENSION(40.,20.,0.,13.5,0.8,$,$);
#256= MILLING_TECHNOLOGY(0.005,.TCP.,2.3,$,$,.T.,.F.,.F.,$);
#257= MILLING_CUTTING_TOOL('COUNTERSINK 40',#259,(#258),70.,#482,20.);
#258= CUTTING_COMPONENT(30.,#332,$,1400.,#261);
#259= COUNTERSINK(#260,2,.RIGHT.,.F.,$,2.7);
#260= TOOL_DIMENSION(45.,45.,$,28.,$,$,$);
#261= MILLING_TECHNOLOGY(0.09,.TCP.,1.2,$,$,.T.,.F.,.F.,$);
#262= MILLING_CUTTING_TOOL('DRILL 15',#264,(#263),165.,$,15.);
#263= CUTTING_COMPONENT(30.,#332,$,1350.,#251);
```

```
#264= TWIST_DRILL(#265,2,.LEFT.,$,$);
#265= TOOL_DIMENSION(15.,31.,$,$,$,$,$);
#266= MILLING_CUTTING_TOOL('BORING TOOL 25',#268,(#267),180.,#482,25.);
#267= CUTTING_COMPONENT(30.,#328,$,1000.,#270);
#268= BORING_TOOL(#269,2,.RIGHT.,.T.,$,.F.);
#269= TOOL_DIMENSION(25.,7.,$,4.5,0.35,$,$);
#270= MILLING_TECHNOLOGY(0.005,.TCP.,2.1,$,$,.T.,.F.,.F.,$);
#271= MILLING_CUTTING_TOOL('COUNTERSINK 25',#273,(#272),60.,#482,20.);
#272= CUTTING_COMPONENT(30.,#332,$,1400.,#261);
#273= COUNTERSINK(#274,2,.RIGHT.,.F.,$,2.27);
#274= TOOL_DIMENSION(30.,45.,$,18.,$,$,$);
#275= MILLING_CUTTING_TOOL('DRILL 10',#277,(#276),90.,#482,10.);
#276= CUTTING_COMPONENT(30.,#332,$,1280.,#251);
#277= TWIST_DRILL(#278,2,.RIGHT.,$,$);
#278= TOOL_DIMENSION(10.,31.,$,$,$,$,$);
#279= MILLING_CUTTING_TOOL('TAP M10',#281,(#280),63.,#482,7.);
#280= CUTTING_COMPONENT(30.,#332,$,200.,#283);
#281= TAP(#282,$,.RIGHT.,$,$);
#282= TOOL_DIMENSION(10.,$,$,$,$,$,$);
#283= MILLING_TECHNOLOGY(0.035809,.TCP.,0.9,$,$,.T.,.F.,.F.,$);
#284= MILLING_CUTTING_TOOL('COUNTERSINK 10',#286,(#285),50.,#482,10.);
#285= CUTTING_COMPONENT(30.,#332,$,1400.,#261);
#286= COUNTERSINK(#287,2,.RIGHT.,.F.,$,1.84);
#287= TOOL_DIMENSION(15.,45.,$,8.,$,$,$);
#288= MILLING_CUTTING_TOOL('DRILL 9',#290,(#289),80.,#482,9.);
#289= CUTTING_COMPONENT(30.,#332,$,1230.,#251);
#290= TWIST_DRILL(#291,2,.RIGHT.,$,$);
#291= TOOL_DIMENSION(9.,31.,$,$,$,$,$);
#292= MILLING_CUTTING_TOOL('REAMER 10',#294,(#293),100.,#482,8.);
#293= CUTTING_COMPONENT(30.,#328,$,980.,#296);
#294= REAMER(#295,6,.LEFT.,$,$);
#295= TOOL_DIMENSION(10.,20.,0.3,70.,0.5,1.5,3.6);
#296= MILLING_TECHNOLOGY(0.08,.TCP.,1.2,$,$,.T.,.F.,.F.,$);
#297= MILLING_CUTTING_TOOL('T-SLOT MILL',#299,(#298),62.,#482,15.);
#298= CUTTING_COMPONENT(30.,#328,$,1120.,#301);
#299= T_SLOT_MILL(#300,4,.RIGHT.,.F.,$,3.5);
#300= TOOL_DIMENSION(35.,$,$,$,$,$,$);
#301= MILLING_TECHNOLOGY(0.0165,.TCP.,5.5,$,$,.T.,.F.,.F.,$);
#302= MILLING_CUTTING_TOOL('ENDMILL 16',#304,(#303),54.,#482,10.);
#303= CUTTING_COMPONENT(30.,#328,$,1110.,#306);
#304= ENDMILL(#305,3,.RIGHT.,.F.,$);
#305= TOOL_DIMENSION(16.,$,$,$,$,$,$);
#306= MILLING_TECHNOLOGY(0.024,.TCP.,$,120.,$,.T.,.F.,.F.,$);
#307= MILLING_CUTTING_TOOL('T-SLOT MILL 25',#309,(#308),63.,#482,10.);
#308= CUTTING_COMPONENT(30.,#328,$,1230.,#311);
#309= T_SLOT_MILL(#310,4,.RIGHT.,.F.,$,18.);
#310= TOOL_DIMENSION(25.,$,$,$,$,$,$);
#311= MILLING_TECHNOLOGY($,.TCP.,$,70.,0.18,.T.,.F.,.F.,$);
#312= MILLING_CUTTING_TOOL('ENDMILL 18',#233,(#232),64.,#482,15.);
#313= MILLING_CUTTING_TOOL('T-SLOT MILL 32MM',#315,(#314),72.,#482,10.);
#314= CUTTING_COMPONENT(30.,#226,$,1120.,#317);
#315= T_SLOT_MILL(#316,4,.NEUTRAL.,.F.,$,20.);
#316= TOOL_DIMENSION(32.,$,$,$,$,$,$);
#317= MILLING_TECHNOLOGY(0.02,.TCP.,$,70.,$,.T.,.F.,.F.,$);
#318= MILLING_CUTTING_TOOL('ENDMILL 8',#320,(#319),54.,#482,8.);
#319= CUTTING_COMPONENT(30.,#230,$,1000.,#322);
#320= ENDMILL(#321,3,.RIGHT.,.F.,$);
```

```
#321= TOOL_DIMENSION(8.,$,$,$,$,$,$);
#322= MILLING_TECHNOLOGY($,.TCP.,3.,$,0.1,.T.,.F.,.F.,$);
#323= MILLING_CUTTING_TOOL('ENDMILL 10',#325,(#324),54.,#482,8.);
#324= CUTTING_COMPONENT(30.,#230,$,1000.,#327);
#325= ENDMILL(#326,3,.RIGHT.,.F.,$);
#326= TOOL_DIMENSION(10.,$,$,$,$,$,$);
#327= MILLING_TECHNOLOGY($,.TCP.,$,120.,0.1,.T.,.F.,.F.,$);
#328= MATERIAL('TIN','TIN',());
#329= TOOL_DIMENSION(125.,$,$,$,$,$,$);
#330= MATERIAL('TIN','TIN',());
#331= TOOL_DIMENSION(18.,$,$,$,$,$,$);
#332= MATERIAL('EMO5CO5','HSS/CO',());
#333= ELEMENTARY_SURFACE('SECURITIY PLANE',#347);
#334= ELEMENTARY_SURFACE('PLANAR FACE1-DEPTH PLANE',#351);
#335= ELEMENTARY_SURFACE('SIDE1-DEPTH',#354);
#336= ELEMENTARY_SURFACE('STEP1-DEPTH',#358);
#337= ELEMENTARY_SURFACE('POCKET1-DEPTH',#362);
#338= ELEMENTARY_SURFACE('SLOT1:DEPTH',#396);
#339= ELEMENTARY_SURFACE('SLOT2:DEPTH',#397);
#340= ELEMENTARY_SURFACE('TEE1:DEPTH',#398);
#341= ELEMENTARY_SURFACE('SIDE1:DEPTH',#399);
#342= AXIS2_PLACEMENT_3D('SETUP1',#401,#402,#403);
#344= AXIS2_PLACEMENT_3D('PL_WORKPIECE_EXAMPLE2',#404,#405,#406);
#347= AXIS2_PLACEMENT_3D('PL_MAIN_SECPLANE',#407,#408,#409);
#348= AXIS2_PLACEMENT_3D('PL_HEAD1',#410,#411,#412);
#351= AXIS2_PLACEMENT_3D('PLANAR FACE1',#413,#414,#415);
#352= AXIS2_PLACEMENT_3D('SIDE1-PLACEMENT',#416,#417,#418);
#354= AXIS2_PLACEMENT_3D('SIDE1-DEPTH',#419,#420,#421);
#355= AXIS2_PLACEMENT_3D('STEP1-PLACEMENT',#422,#423,#424);
#358= AXIS2_PLACEMENT_3D('STEP1-DEPTH',#425,#426,#427);
#359= AXIS2_PLACEMENT_3D('POCKET1-PLACEMENT',#428,#429,#430);
#362= AXIS2_PLACEMENT_3D('POCKET1-DEPTH',#431,#432,#433);
#363= AXIS2_PLACEMENT_3D('CF_HOLE1',#434,#435,#436);
#368= AXIS2_PLACEMENT_3D('HOLE1',#437,#438,#439);
#369= AXIS2_PLACEMENT_3D('HOLE2',#440,#435,#436);
#370= AXIS2_PLACEMENT_3D('CF_HOLE3',#441,#442,#443);
#373= AXIS2_PLACEMENT_3D('HOLE3',#444,#442,#443);
#374= AXIS2_PLACEMENT_3D('GP_HOLE4',#445,#446,#447);
#376= AXIS2_PLACEMENT_3D('HOLE4_NEAR_REGION',#448,#446,#447);
#377= AXIS2_PLACEMENT_3D('HOLE4_NEAR_POCKET',#449,#446,#446);
#379= AXIS2_PLACEMENT_3D('HOLE4',#450,#446,#446);
#380= AXIS2_PLACEMENT_3D('GP_HOLE5',#451,#452,#453);
#382= AXIS2_PLACEMENT_3D('HOLE5_NEAR_REGION',#454,#452,#453);
#383= AXIS2_PLACEMENT_3D('HOLE5_NEAR_POCKET',#455,#452,#453);
#385= AXIS2_PLACEMENT_3D('HOLE5',#456,#452,#453);
#386= AXIS2_PLACEMENT_3D('GP_HOLE6',#457,#458,#459);
#388= AXIS2_PLACEMENT_3D('HOLE6_NEAR_REGION',#460,#458,#459);
#389= AXIS2_PLACEMENT_3D('HOLE6_NEAR_POCKET',#461,#458,#459);
#391= AXIS2_PLACEMENT_3D('HOLE6',#462,#458,#459);
#392= AXIS2_PLACEMENT_3D('SLOT1',#463,#464,#465);
#393= AXIS2_PLACEMENT_3D('SLOT2',#466,#467,#468);
#394= AXIS2_PLACEMENT_3D('TEE_SLOT',#469,#470,#471);
#395= AXIS2_PLACEMENT_3D('SIDE_SLOT',#472,#473,#474);
#396= AXIS2_PLACEMENT_3D('SLOT1:DEPTH',#475,$,$);
#397= AXIS2_PLACEMENT_3D('SLOT2:DEPTH',#476,$,$);
#398= AXIS2_PLACEMENT_3D('TEE_SLOT:DEPTH',#477,$,$);
#399= AXIS2_PLACEMENT_3D('SIDE_SLOT:DEPTH',#478,$,$);
```

```
#400= AXIS2_PLACEMENT_3D('SIDE_SLOT:SQUARE',#479,#480,#481);
#401= CARTESIAN_POINT('SETUP1:    LOCATION',(123.,123.,45.));
#402= DIRECTION('                     AXIS',(0.,0.,1.));
#403= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#404= CARTESIAN_POINT('WORKPIECE1:LOCATION',(0.,0.,0.));
#405= DIRECTION('                     AXIS',(0.,0.,1.));
#406= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#407= CARTESIAN_POINT('SECPLANE1: LOCATION',(0.,0.,30.));
#408= DIRECTION('                     AXIS',(0.,0.,1.));
#409= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#410= CARTESIAN_POINT('PLANAR FACE : LOCATION',(0.,0.,5.));
#411= DIRECTION('                     AXIS',(0.,0.,1.));
#412= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#413= CARTESIAN_POINT('PLANAR FACE1:DEPTH LOCATION',(0.,0.,-5.));
#414= DIRECTION('                   AXIS          ',(0.,0.,1.));
#415= DIRECTION('                   REF_DIRECTION ',(1.,0.,0.));
#416= CARTESIAN_POINT('SIDE1 PROFILE:LOCATION',(0.,0.,0.));
#417= DIRECTION('                     AXIS',(0.,0.,1.));
#418= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#419= CARTESIAN_POINT('SIDE1 PROFILE:DEPTH',(0.,0.,-50.));
#420= DIRECTION('                     AXIS',(0.,0.,1.));
#421= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#422= CARTESIAN_POINT('STEP1 PROFILE:LOCATION',(230.,40.,0.));
#423= DIRECTION('                     AXIS',(0.,0.,1.));
#424= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#425= CARTESIAN_POINT('STEP1 PROFILE:DEPTH',(0.,0.,-20.));
#426= DIRECTION('                     AXIS',(0.,0.,1.));
#427= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#428= CARTESIAN_POINT('POCKET1 PROFILE:LOCATION',(50.,0.,0.));
#429= DIRECTION('                     AXIS',(0.,0.,1.));
#430= DIRECTION('            REF_DIRECTION',(1.,0.,0.));
#431= CARTESIAN_POINT('POCKET1 PROFILE:DEPTH',(0.,0.,-20.));
#432= DIRECTION('                     AXIS',(0.,0.,1.));
#433= DIRECTION('          REF_DIRECTION',(1.,0.,0.));
#434= CARTESIAN_POINT('',(160.,150.,0.));
#435= DIRECTION('',(0.,0.,1.));
#436= DIRECTION('',(1.,0.,0.));
#437= CARTESIAN_POINT('',(0.,0.,0.));
#438= DIRECTION('',(0.,0.,1.));
#439= DIRECTION('',(1.,0.,0.));
#440= CARTESIAN_POINT('',(0.,0.,0.));
#441= CARTESIAN_POINT('',(160.,30.,0.));
#442= DIRECTION('',(0.,0.,1.));
#443= DIRECTION('',(1.,0.,0.));
#444= CARTESIAN_POINT('',(0.,0.,0.));
#445= CARTESIAN_POINT('',(130.,120.,0.));
#446= DIRECTION('',(0.,0.,1.));
#447= DIRECTION('',(1.,0.,0.));
#448= CARTESIAN_POINT('',(0.,0.,0.));
#449= CARTESIAN_POINT('',(0.,-60.,0.));
#450= CARTESIAN_POINT('',(0.,0.,0.));
#451= CARTESIAN_POINT('',(190.,120.,0.));
#452= DIRECTION('',(0.,0.,1.));
#453= DIRECTION('',(1.,0.,0.));
#454= CARTESIAN_POINT('',(0.,0.,0.));
#455= CARTESIAN_POINT('',(0.,-60.,0.));
#456= CARTESIAN_POINT('',(0.,0.,0.));
```

```
#457= CARTESIAN_POINT('',(245.,110.,0.));
#458= DIRECTION('',(0.,0.,1.));
#459= DIRECTION('',(1.,0.,0.));
#460= CARTESIAN_POINT('',(0.,0.,0.));
#461= CARTESIAN_POINT('',(0.,-40.,0.));
#462= CARTESIAN_POINT('',(0.,0.,0.));
#463= CARTESIAN_POINT('',(230.,90.,0.));
#464= DIRECTION('',(0.,0.,1.));
#465= DIRECTION('',(1.,0.,0.));
#466= CARTESIAN_POINT('',(85.,90.,0.));
#467= DIRECTION('',(0.,0.,1.));
#468= DIRECTION('',(1.,0.,0.));
#469= CARTESIAN_POINT('',(0.,90.,0.));
#470= DIRECTION('',(0.,0.,1.));
#471= DIRECTION('',(1.,0.,0.));
#472= CARTESIAN_POINT('',(160.,90.,0.));
#473= DIRECTION('',(0.,0.,1.));
#474= DIRECTION('',(1.,0.,0.));
#475= CARTESIAN_POINT('SLOT1:DEPTH',(0.,0.,-10.));
#476= CARTESIAN_POINT('SLOT2:DEPTH',(0.,0.,-10.));
#477= CARTESIAN_POINT('TEE1:DEPTH',(0.,0.,-10.));
#478= CARTESIAN_POINT('SIDE1:DEPTH',(0.,0.,-10.));
#479= CARTESIAN_POINT('SIDE1:SQUARE U PROFILE',(-22.,0.,-20.));
#480= DIRECTION('',(1.,0.,0.));
#481= DIRECTION('',(0.,0.,-1.));
#482= DIRECTION('X',(1.,0.,0.));
#483= DIRECTION('',(1.,0.,0.));
#484= POLYLINE('TEST1',(#485,#486));
#485= CARTESIAN_POINT('',(25.,15.,0.));
#486= CARTESIAN_POINT('',(45.,15.,0.));
#487= COMPOSITE_CURVE_SEGMENT(.CONT_SAME_GRADIENT.,.T.,#488);
#488= TRIMMED_CURVE('TEST2',#489,(PARAMETER_VALUE(270.)),(PARAMETER_VALUE(0.)),.T.,.PARAMETER.);
#489= CIRCLE('',#490,5.);
#490= AXIS2_PLACEMENT_3D('',#491,#492,#493);
#491= CARTESIAN_POINT('',(45.,20.,0.));
#492= DIRECTION('',(0.,0.,1.));
#493= DIRECTION('',(1.,0.,0.));
#494= POLYLINE('',(#495,#496));
#495= CARTESIAN_POINT('',(50.,20.,0.));
#496= CARTESIAN_POINT('',(50.,30.,0.));
#497= COMPOSITE_CURVE_SEGMENT(.CONT_SAME_GRADIENT.,.T.,#498);
#498= TRIMMED_CURVE('',#499,(PARAMETER_VALUE(0.)),(PARAMETER_VALUE(90.)),.T.,.PARAMETER.);
#499= CIRCLE('',#500,5.);
#500= AXIS2_PLACEMENT_3D('',#501,#492,#493);
#501= CARTESIAN_POINT('',(45.,30.,0.));
#502= POLYLINE('',(#503,#504));
#503= CARTESIAN_POINT('',(45.,35.,0.));
#504= CARTESIAN_POINT('',(25.,35.,0.));
#505= COMPOSITE_CURVE_SEGMENT(.CONT_SAME_GRADIENT.,.T.,#506);
#506= TRIMMED_CURVE('',#507,(PARAMETER_VALUE(90.)),(PARAMETER_VALUE(180.)),.T.,.PARAMETER.);
#507= CIRCLE('',#508,5.);
#508= AXIS2_PLACEMENT_3D('',#509,#492,#493);
#509= CARTESIAN_POINT('',(25.,30.,0.));
#510= POLYLINE('',(#511,#512));
#511= CARTESIAN_POINT('',(20.,30.,0.));
#512= CARTESIAN_POINT('',(20.,20.,0.));
#513= COMPOSITE_CURVE_SEGMENT(.CONT_SAME_GRADIENT.,.T.,#514);
```

```
#514= TRIMMED_CURVE('',#515,(PARAMETER_VALUE(180.)),(PARAMETER_VALUE(270.)),.T.,.PARAMETER.);
#515= CIRCLE('',#516,5.);
#516= AXIS2_PLACEMENT_3D('',#517,#492,#493);
#517= CARTESIAN_POINT('',(25.,20.,0.));
#518= POLYLINE('',(#519,#520));
#519= CARTESIAN_POINT('',(0.,0.,0.));
#520= CARTESIAN_POINT('',(0.,40.,0.));
#521= COMPOSITE_CURVE_SEGMENT(.CONT_SAME_GRADIENT.,.T.,#522);
#522= TRIMMED_CURVE('',#523,(PARAMETER_VALUE(180.)),(PARAMETER_VALUE(90.)),.F.,.PARAMETER.);
#523= CIRCLE('',#524,10.);
#524= AXIS2_PLACEMENT_3D('',#525,#492,#493);
#525= CARTESIAN_POINT('',(10.,40.,0.));
#526= POLYLINE('',(#527,#528));
#527= CARTESIAN_POINT('',(10.,50.,0.));
#528= CARTESIAN_POINT('',(60.,50.,0.));
#529= COMPOSITE_CURVE_SEGMENT(.CONT_SAME_GRADIENT.,.T.,#530);
#530= TRIMMED_CURVE('',#531,(PARAMETER_VALUE(90.)),(PARAMETER_VALUE(0.)),.F.,.PARAMETER.);
#531= CIRCLE('',#532,10.);
#532= AXIS2_PLACEMENT_3D('',#533,#492,#493);
#533= CARTESIAN_POINT('',(60.,40.,0.));
#534= POLYLINE('',(#535,#536));
#535= CARTESIAN_POINT('',(70.,40.,0.));
#536= CARTESIAN_POINT('',(70.,0.,0.));
#537= TOLERANCED_LENGTH_MEASURE(40.,#780);
#538= TOLERANCED_LENGTH_MEASURE(40.,#780);
#539= TOLERANCED_LENGTH_MEASURE(40.,#781);
#541= TOLERANCED_LENGTH_MEASURE(0.5,#781);
#542= TOLERANCED_LENGTH_MEASURE(41.,#782);
#544= TOLERANCED_LENGTH_MEASURE(30.,#782);
#545= TOLERANCED_LENGTH_MEASURE(15.,#782);
#546= TOLERANCED_LENGTH_MEASURE(22.,#781);
#548= TOLERANCED_LENGTH_MEASURE(30.,#781);
#549= TOLERANCED_LENGTH_MEASURE(22.,#781);
#551= TOLERANCED_LENGTH_MEASURE(0.5,#781);
#552= TOLERANCED_LENGTH_MEASURE(23.,#781);
#554= TOLERANCED_LENGTH_MEASURE(40.,#781);
#555= TOLERANCED_LENGTH_MEASURE(15.,#781);
#556= TOLERANCED_LENGTH_MEASURE(22.,#781);
#558= TOLERANCED_LENGTH_MEASURE(40.,#781);
#559= TOLERANCED_LENGTH_MEASURE(22.,#781);
#563= TOLERANCED_LENGTH_MEASURE(0.5,#781);
#564= TOLERANCED_LENGTH_MEASURE(23.,#781);
#565= TOLERANCED_LENGTH_MEASURE(15.,#781);
#566= TOLERANCED_LENGTH_MEASURE(15.,#781);
#567= TOLERANCED_LENGTH_MEASURE(10.,#781);
#569= TOLERANCED_LENGTH_MEASURE(0.5,#781);
#570= TOLERANCED_LENGTH_MEASURE(11.,#781);
#574= TOLERANCED_LENGTH_MEASURE(25.,#781);
#575= TOLERANCED_LENGTH_MEASURE(25.,#781);
#576= TOLERANCED_LENGTH_MEASURE(9.,#781);
#578= TOLERANCED_LENGTH_MEASURE(15.5,#781);
#579= TOLERANCED_LENGTH_MEASURE(8.3,#781);
#581= TOLERANCED_LENGTH_MEASURE(1.75,#781);
#582= TOLERANCED_LENGTH_MEASURE(0.5,#781);
#583= TOLERANCED_LENGTH_MEASURE(11.,#781);
#587= TOLERANCED_LENGTH_MEASURE(20.,#781);
#588= TOLERANCED_LENGTH_MEASURE(20.,#781);
```

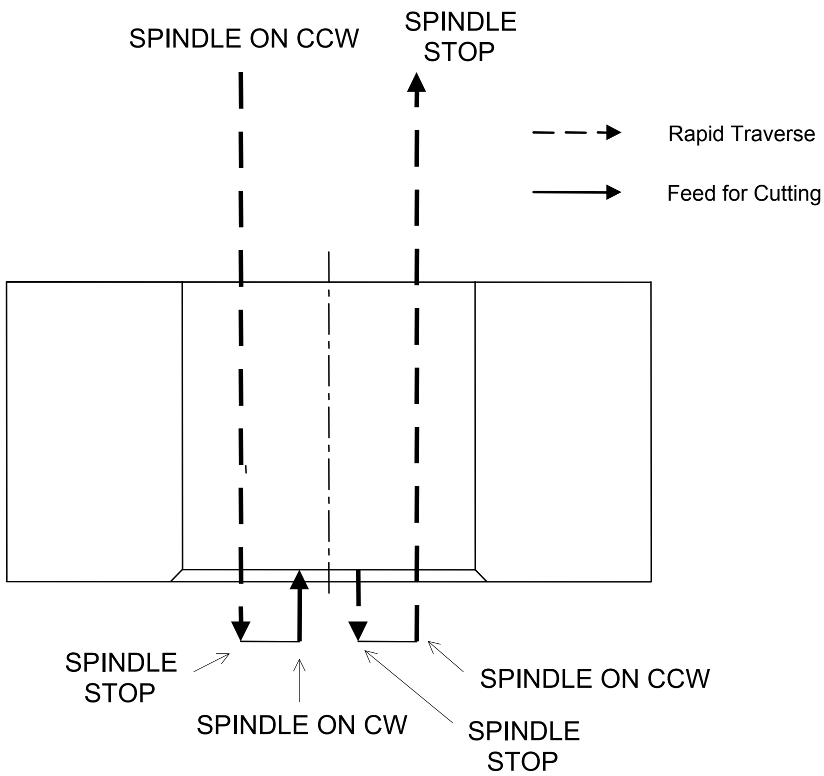
```
#589= TOLERANCED_LENGTH_MEASURE(10.,#783);
#591= TOLERANCED_LENGTH_MEASURE(0.5,#783);
#592= TOLERANCED_LENGTH_MEASURE(11.,#783);
#780= PLUS_MINUS_VALUE(0.5,0.,3);
#781= PLUS_MINUS_VALUE(0.025,0.,3);
#782= PLUS_MINUS_VALUE(0.1,0.1,3);
#783= PLUS_MINUS_VALUE(0.015,0.,3);
ENDSEC;

END-ISO-10303-21;
```

F.3 Example 3

Example of Tool_path_list

Backside Counterbore Using a Special Tool Holder



[Specification of the special tool holder for backside_counterbore]
CCW rotation of the spindle will retract the tool edge to the inside of the tool holder.

Fig. 27: Back boring operation of the Example 3

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ISO 14649-11 EXAMPLE 3','SIMPLE PRORGRAM WITH TOOLPATH'),'1');
FILE_NAME('EXAMPLE3.STP',
        '2001-08-24',
        ('YONG TAK HYUN','C.SAKAMOTO'),
        ('WZL, RWTH-AACHEN','KOMATSU'),
        $,
        'ISO 14649',
        $);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;
DATA;
#1= WORKPIECE('EXAMPLE3_WORKPIECE',#2,$,$,$,$,$);
#2= MATERIAL('FC','200',());
#3= ROUND_HOLE('HOLE12',#1,(),#10,#11,#15,#8,#9);
#4= COUNTERSUNK_HOLE('HOLE1',#1,(),#6,(#5,#3));
#5= ROUND_HOLE('HOLE11',#1,(),#25,#26,#15,$,#9);
#6= AXIS2_PLACEMENT_3D('CUNTERSUNK_HOLE_PLACEMENT',#7,$,$);
```

```
#7= CARTESIAN_POINT('HOLE1_COORDINATE',(160.,90.,0.));
#8= ANGLE_TAPER(135.);
#9= THROUGH_BOTTOM_CONDITION();
#10= AXIS2_PLACEMENT_3D('HOLE11_PLACEMET',#14,$,$);
#11= ELEMENTARY_SURFACE('DEPTH OF HOLE 12',#12);
#12= AXIS2_PLACEMENT_3D('DEPTH OF HOLE 12',#13,$,$);
#13= CARTESIAN_POINT('HOLE 12:DEPTH:LOCATION',(0.,0.,-1.));
#14= CARTESIAN_POINT('HOLE 11:DEPTH:LOCATION',(0.,0.,-39.));
#15= TOLERANCED_LENGTH_MEASURE(40.,#16);
#16= PLUS_MINUS_VALUE(0.2,0.2,3);
#17= PROJECT('ISO14649',#18,(#1),$,$,$);
#18= WORKPLAN('EXAMPLE3_WORKPLAN',(#30),#21,#22,$);
#19= SURFACE_TEXTURE_PARAMETER(25.,'R','ISO4287','MAX',(#20));
#20= MACHINED_SURFACE(#5,.SIDE.);
#21= CHANNEL('CHANNEL1');
#22= SETUP('SETUP1',$,#41,(#23));
#23= WORKPIECE_SETUP(#1,#45,$,$,());
#24= CARTESIAN_POINT('HOLE 12:LOCATION',(0.,0.,0.));
#25= AXIS2_PLACEMENT_3D('HOLE11_PLACEMENT',#24,$,$);
#26= ELEMENTARY_SURFACE('DEPTH OF HOLE',#27);
#27= AXIS2_PLACEMENT_3D('DEPTH OF HOLE 11',#28,$,$);
#28= CARTESIAN_POINT('HOLE 11:LOCATION',(0.,0.,-39.));
#29= BACK_BORING(#31,$,'BACKSIDE_COUNTERBORING1',$,$,#47,#37,#38,$,$,$,$,$,$);
#30= MACHINING_WORKINGSTEP('BACKSIDE_COUNTERBORING1',#56,#3,#29,$);
#31= TOOLPATH_LIST((#32,#33,#34,#35,#36));
#32= CUTTER_LOCATION_TRAJECTORY(.T.,.APPROACH.,.RAPID.,#39,$,.T.,#51,$,$);
#33= CUTTER_LOCATION_TRAJECTORY(.T.,.TRAJECTORY_PATH.,$,$,$,.T.,#54,$,$);
#34= FEEDSTOP(.T.,.CONTACT.,$,$,$,0.5);
#35= CUTTER_LOCATION_TRAJECTORY(.T.,.TRAJECTORY_PATH.,.RAPID.,$,$,.T.,#59,$,$);
#36= CUTTER_LOCATION_TRAJECTORY(.T.,.LIFT.,.RAPID.,#39,$,.T.,#60,$,$);
#37= MILLING_TECHNOLOGY(0.006,.TCP.,$,50.,$,.F.,.F.,.F.,$);
#38= MILLING_MACHINE_FUNCTIONS(.F.,$,$,.F.,$,(),.T.,$,$,());
#39= MILLING_TECHNOLOGY($,$,$,-8.3,$,.F.,.F.,.F.,$);
#40= AXIS2_PLACEMENT_3D('SECPLANE 0',#42,#43,#44);
#41= PLANE('SECURITY PLANE',#40);
#42= CARTESIAN_POINT('SECPLANE0:LOCATION',(0.,0.,70.));
#43= DIRECTION('AXIS',(0.,0.,1.));
#44= DIRECTION('REF DIRECTION',(1.,0.,0.));
#45= AXIS2_PLACEMENT_3D('EXAMPLE3_WORKPIECE_POSITION',#46,$,$);
#46= CARTESIAN_POINT('WORKPIECE_COORINATE',(-4.672,76.863,-88.668));
#47= MILLING_CUTTING_TOOL('SPECIAL_BACKSIDE_COUNTERBORE_50MM',#49,(#48),400.,$,$);
#48= CUTTING_COMPONENT(400.,$,$,$,$);
#49= USER_DEFINED_TOOL(#50,$,$,$,$,'SPECIAL_BACKSIDE_COUNTERBORE1');
#50= TOOL_DIMENSION(50.,$,$,$,$,$,$);
#51= POLYLINE('APPROACH_LINE1',(#52,#53));
#52= CARTESIAN_POINT('APPROACH_START_POINT',(0.,0.,50.));
#53= CARTESIAN_POINT('APPROACH_END_POINT',(0.,0.,-48.));
#54= POLYLINE('CUTTING_LINE1',(#53,#55));
#55= CARTESIAN_POINT('CUTTING_START_POINT',(0.,0.,-45.));
#56= ELEMENTARY_SURFACE('SECURITY_PLANE1',#57);
#57= AXIS2_PLACEMENT_3D('SECURITY_PLACEMENT1',#58,$,$);
#58= CARTESIAN_POINT('SECURITY_LOCATION1',(0.,0.,50.));
#59= POLYLINE('RETURN_LINE1',(#55,#53));
#60= POLYLINE('RETURN_LINE2',(#53,#52));
ENDSEC;
END-ISO-10303-21;
```

Index

A

Adaptive control.......6
Air strategy.......11
Along path.......13
Approach retract angle.......11
Approach retract strategy.......8
Approach retract tangent.......12

B

Back boring.......28
Bidirectional and contour milling.......20
Bidirectional milling.......18
Boring.......28
Boring operation.......27
Bottom and side finish milling.......24
Bottom and side milling.......23
Bottom and side rough milling.......24

C

Center drilling.......27
Center milling.......21
Conformance class 1.......30
Conformance class 2.......31
Conformance requirement.......29
Contour and bidirectional milling.......20
Contour parallel milling.......19
Contour spiral milling.......21
Counter sinking.......27
Cutmode type.......14

D

Drilling.......26
Drilling operation.......26
Drilling type operation.......24
Drilling type strategy.......26

E

Exchange pallet.......2
Explicit strategy.......22

F

Five axes constant tilt and yaw.......4
Five axes variable tilt and yaw.......4
Freeform operation.......13
Freeform strategy.......13

H

Header and References.......2

I

Index pallet.......3
Index table.......3

L

Leading line strategy.......16
Left or right.......19
Load tool.......3

M

Milling machine functions.......6
Milling machining operation.......4
Milling technology.......5
Milling type operation.......7
Multistep drilling.......27

N

NC functions for milling.......2

P

Pathmode type.......14
Plane cutter contact strategy.......15
Plane cutter location strategy.......16
Plane finish milling.......23
Plane milling.......22
Plane rough milling.......22
Plunge helix.......10
Plunge ramp.......9
Plunge strategy.......8
Plunge zigzag.......10
Plunge_tool axis.......8
Process model.......7
Process model list.......7

R

Reaming.......28

S

Side finish milling.......23
Side milling.......23
Side rough milling.......23
Stroke connection strategy.......19

T

Tapping.......29
Thread drilling.......29
Three axes tilted tool.......4
Tolerances.......14
Tool direction for milling.......3
Two5D milling operation.......17
Two5D milling strategy.......17

U

Unidirectional milling.......18
Unload tool.......3
UV strategy.......15