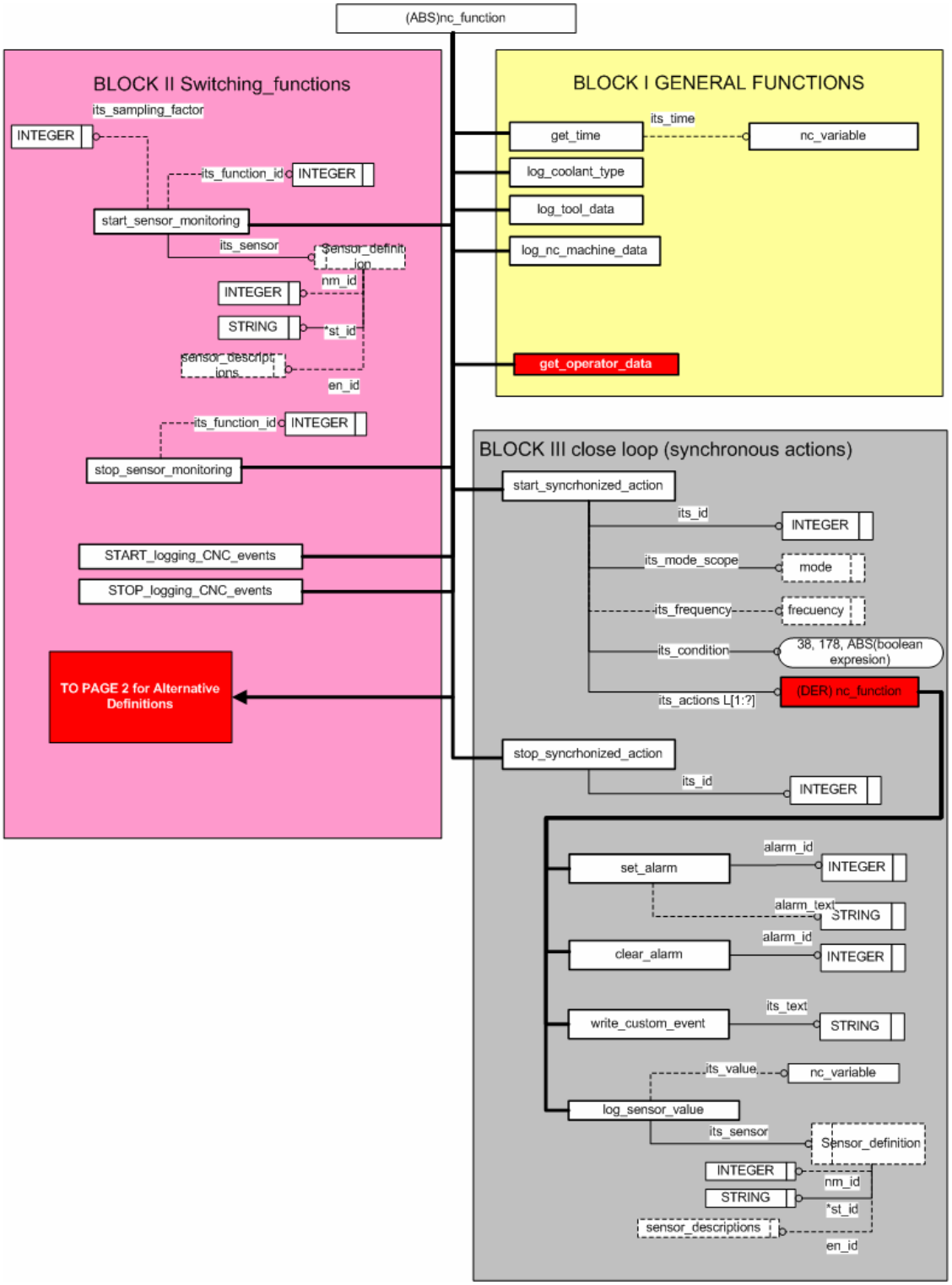
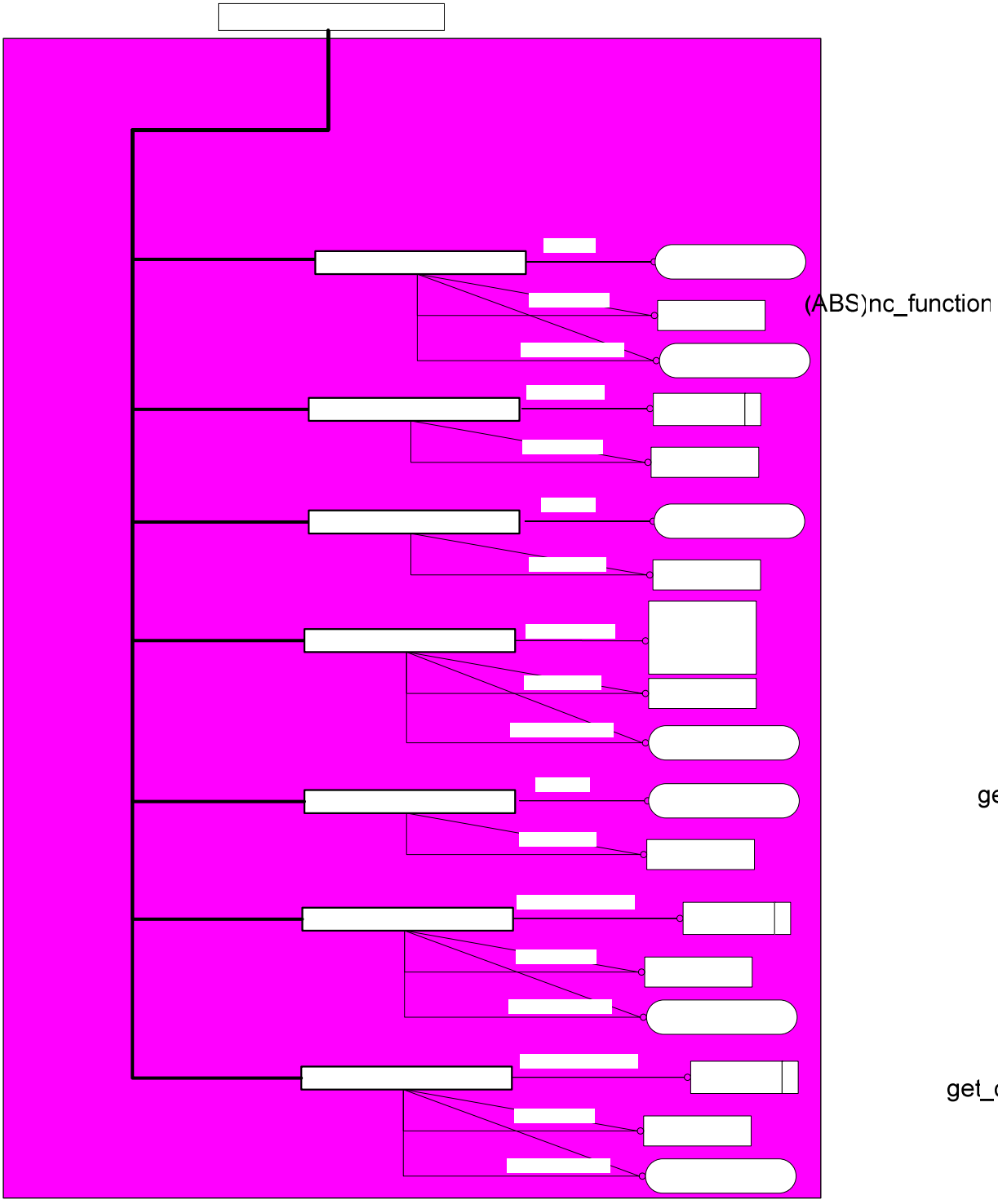
General ARM v 2.0 (approach p. 1 of 2)



General ARM v 2.0 (approach p. 2 of 2)



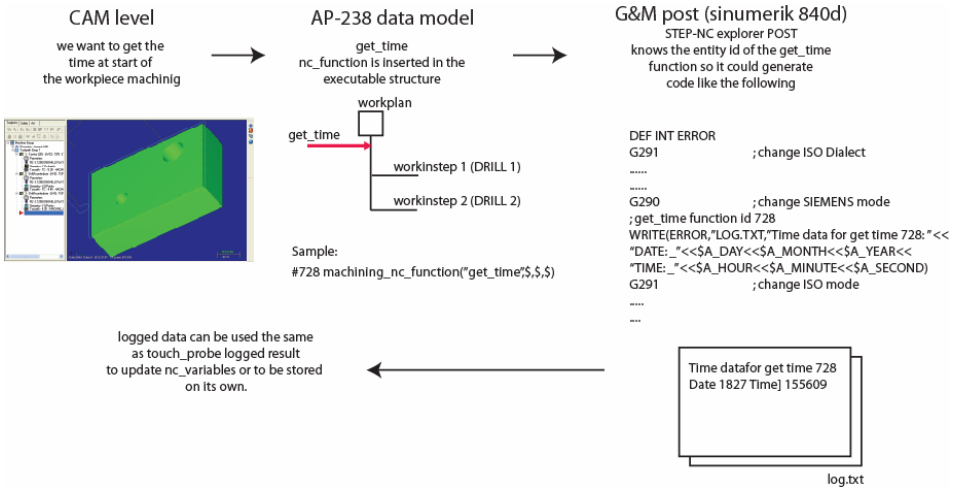
BLOCK I. GENERAL Functions.

GET_TIME

Timing functions can be used to get absolute value times, or the time employed by the CNC to accomplish certain tasks while machining. As an example, the first proposed function is the get_time function.

Get time to record Absolute value in milliseconds.

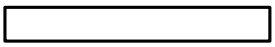
(the nc_variable has been marked as optional to indicate *that the collected time could be simply logged into a file or assigned if needed to an nc variable to be processed by a CAM step application*). This function could be used as follows :



its_time

Other functions to collect more general data not directly related to machine execution.

LOG_TOOL_DATA



It could be also be as a switching ON/OFF function to automatically detect TOOL changes and LOG tool data each time a new tool is used, to avoid having to program a log_tool_data each time a tool change has been programmed in the AP-238 file….

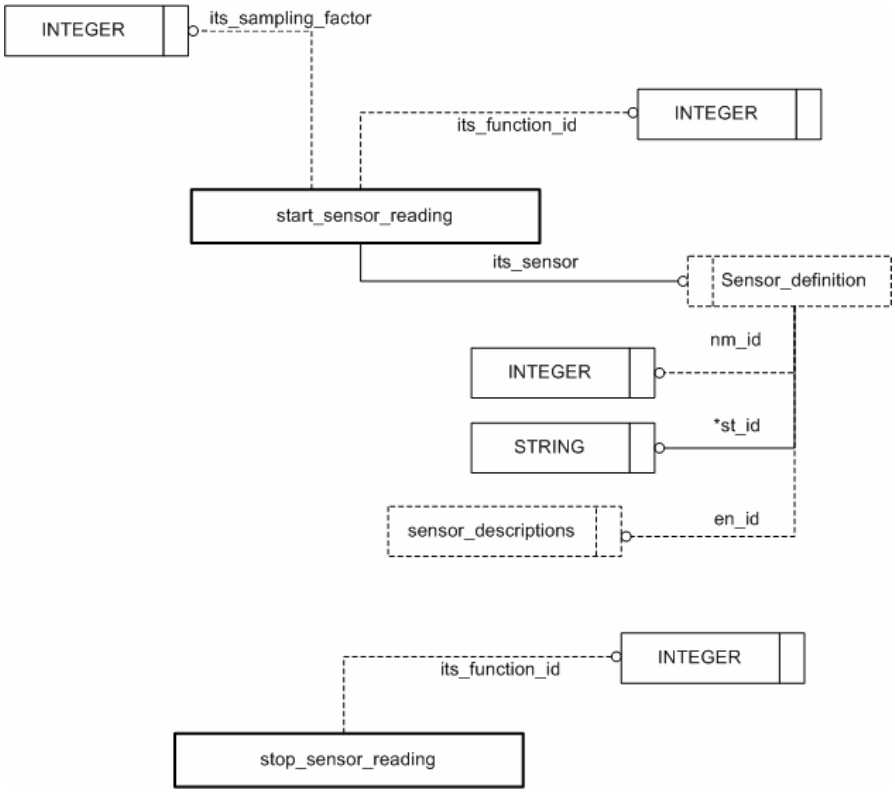
LOG_NC_MACHINE_DATA

To log specific nc_machine data as machine serial number, model and brand. May be also it would be possible to access other internal or system variables, initial or setup conditions, or predefined machine characteristics.

log_nc_machine_data

BLOCK II. Reading Sensor (internal NC variables) Values.

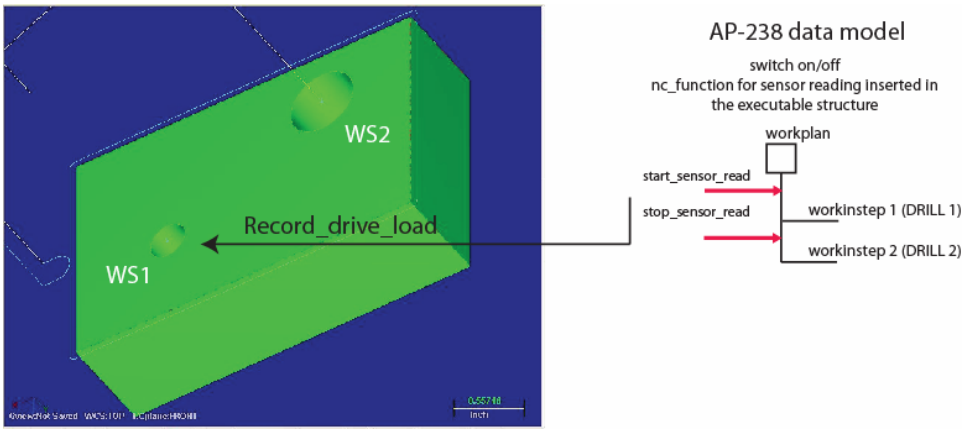
Read_sensor value nc_function as a swithing off/on function:



(1) Sample: Recording % of Drive Load during Workingstep1 Execution

The next sample, tries to show how through the CAM system the % of drive load (It could be configured by axis) could be done, resulting in an AP-238 nc_functions inserted before and after the workingstep (or as part of a workplan including the nc_function and the related workingstep). The POST processor depending on the capabilities of the selected machine, will have to set the synchronous actions to read and log the internal register values.

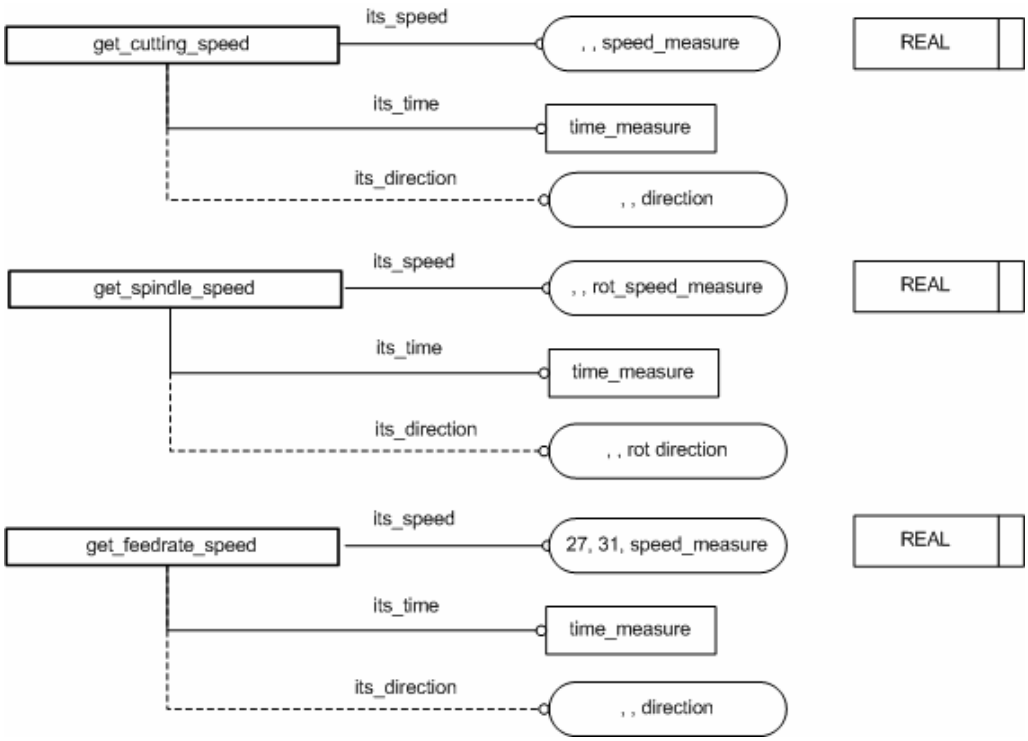
The attribute its_function_id servers to match the starting and ending function, and the sampling_factor attribute of the start rsensor_recording could be used to adjust the number of samples to be recorded (machine cycle per sample)



Another option is to specify different functions for each value, and in this case it could be useful to provide a data structure for each of these functions

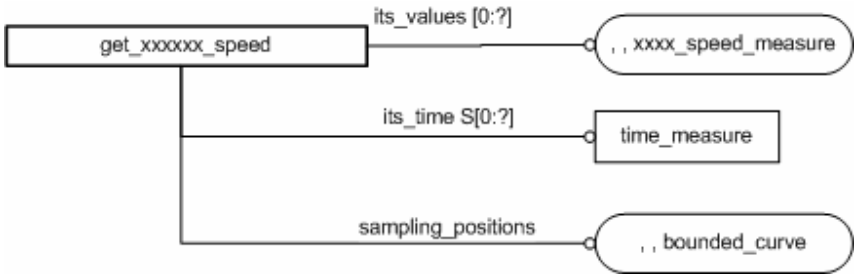
1. Speeds Single Values: Cutting Speed, Spindle Speed and FeedRate Speed. [for these measurements: CNC axis Speeds, CNC Axis/Bearing/Spindle Acelerations (?)]

(measuring single value/ structuring the log file)



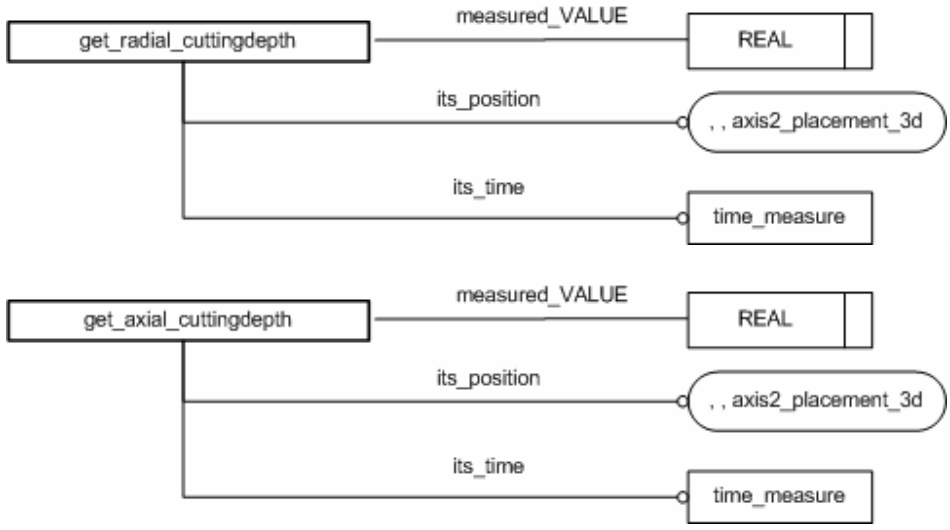
Minimum data to collect is the value of the measure, and the sampling time. As optional parameter, MAYBE it could be included if it is important the direction (?).

[measuring several (bounded quantity) of points]

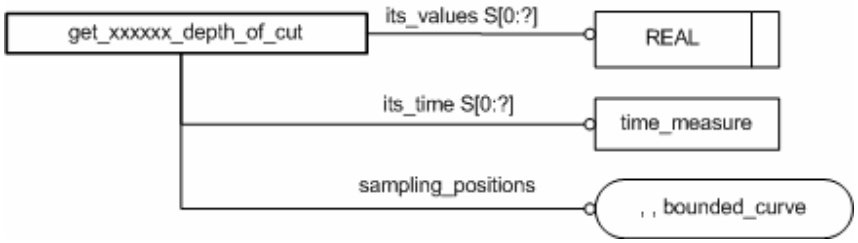


2. Depth of Cut (if it's realistic to think it could be really measured while the cut is being done). It could be separated into RADIAL and AXIAL.

[measuring single value/LOG data File structure]



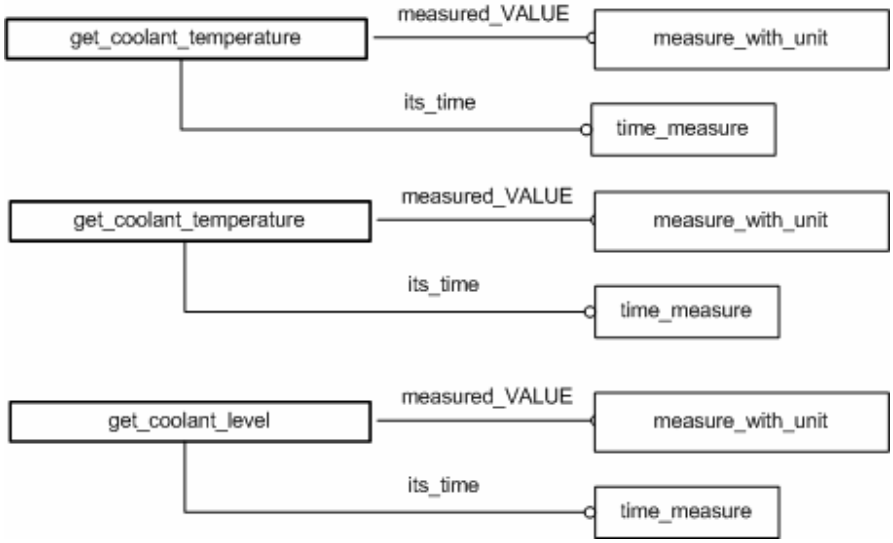
[measuring several (bounded quantity) of points]



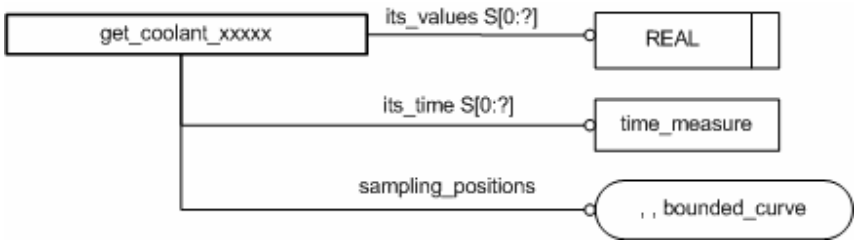
A bounded_curve structure could be used to indicate the points where we want to record the depth of cut…

3. Coolant Pressures, Level and Temperatures (?).

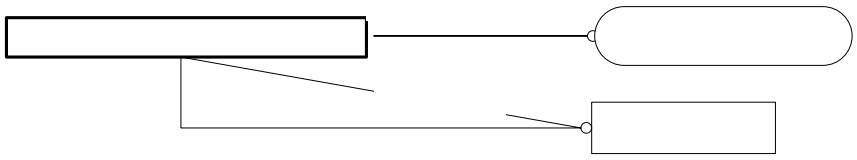
[measuring single value/LOG data File structure]



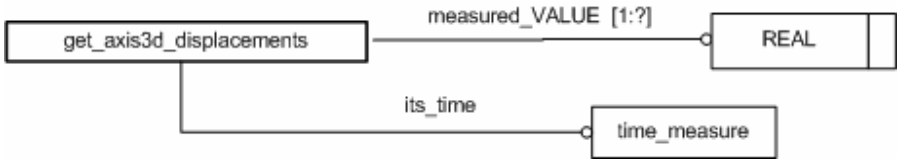
[measuring several (bounded quantity) of points]



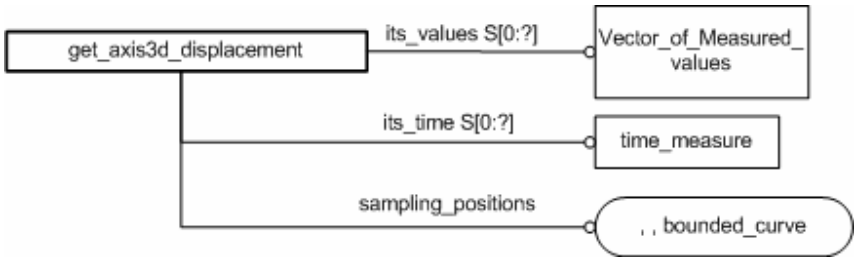
4. Positions (relative/absolute):
   4.1 – Interpolated Positions by the CNC
   4.2 – Positions as given by the encoders
   4.3 – Axis 3D Displacements (tolerances)



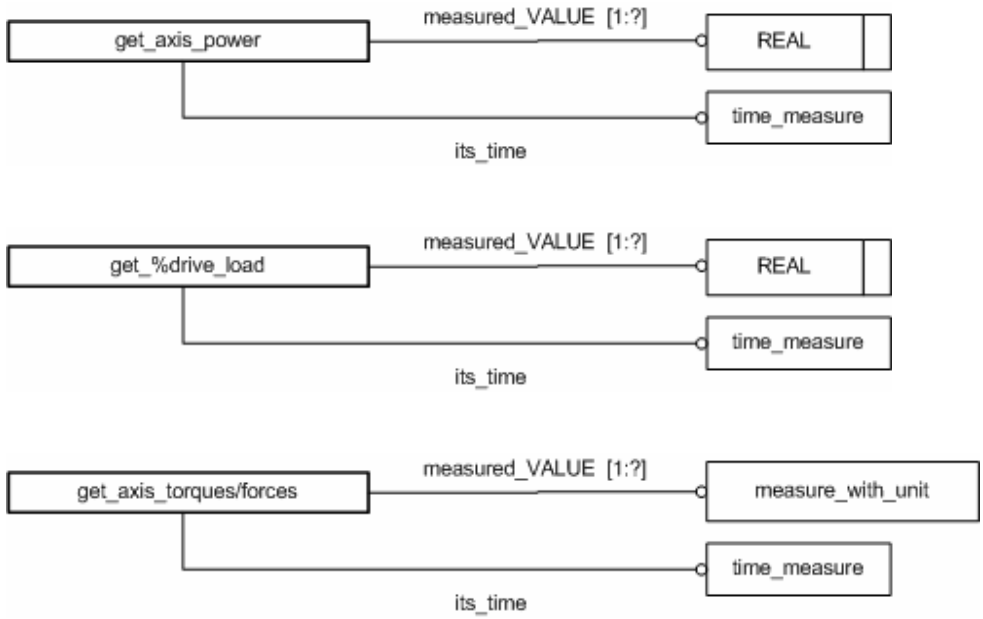
[measuring single value/LOG data File structure]



[measuring several (bounded quantity) of points]

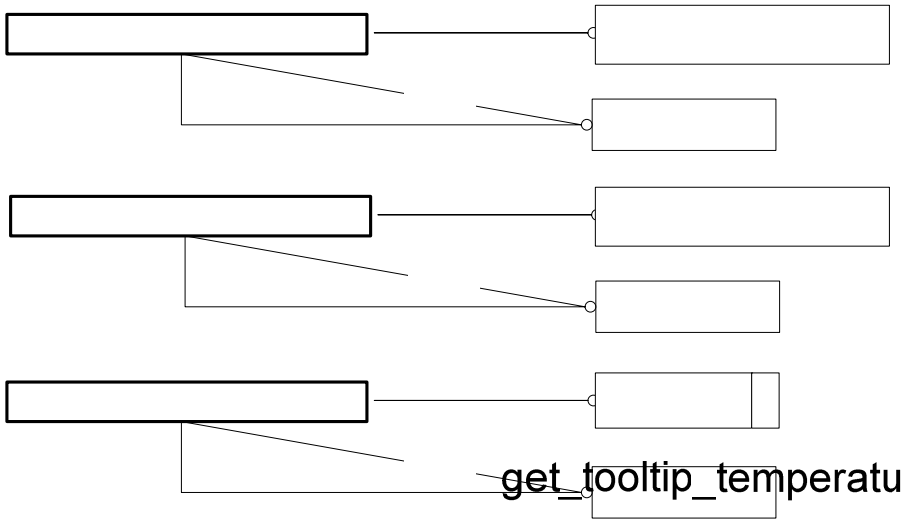


5,6 & 7. Motor Currents or Powers (by Axis/Spindle/channel); % of Drive Loads. ; Machine Forces and Torques



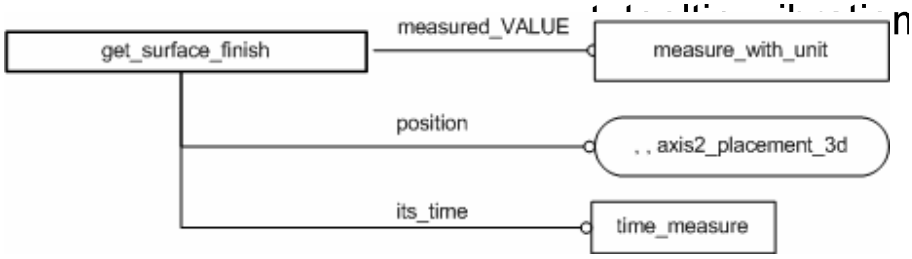
8,9,10. Tool Tip Temperature (Cutting Operation Temperature(?) ; Tool Tip Vibration (or Acoustic Emision).

Indirect measures to detect also Tool wear (?), or including specific tool sensor wear (?).



The measured value could be simply and REAL Value or a measure_with_unit value to indicate the unit of the measure.

10. **Surface Finish (Roughness)**

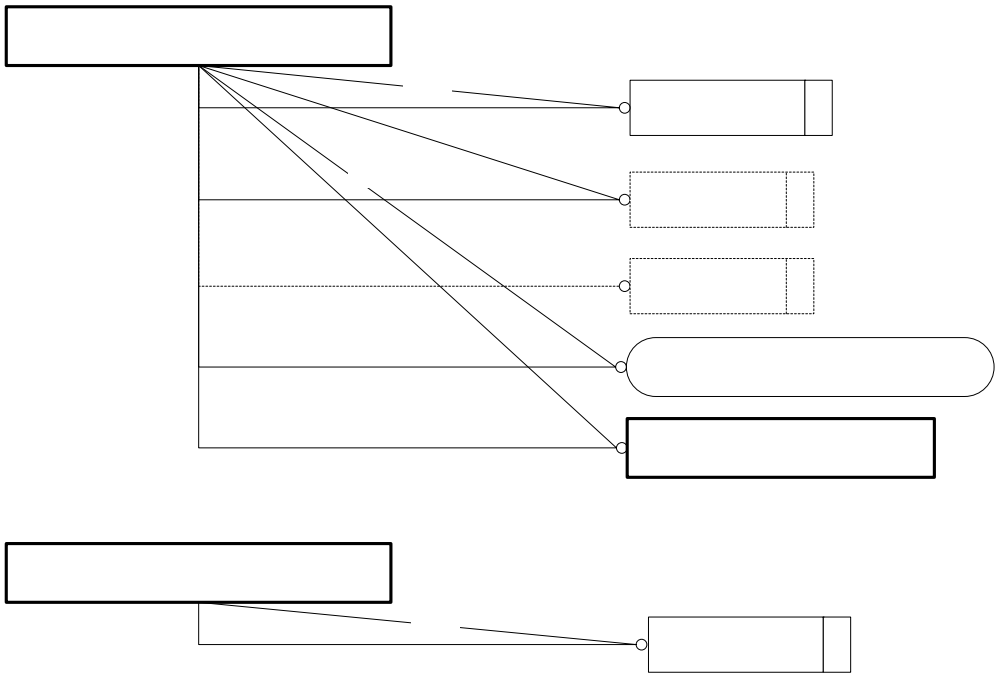


Position for the measured value is OK as axis2_placement3d or as a vector [3] of Cartesian points?.

BLOCK III. A Model for close loop functions

Close loop operations, like: "if workingstep time is greater than a value, do something", can be realized at least in siemens architecture with the aid of **"synchronous motion actions"**.

These are action that can be programmed and cancel anywhere in the G&M code to be evaluated during the CNC interpolation cycles and execute the selected actions including subroutine program call to technological cycles.



Main problems and limited functionality with this approach:

(1) its_id attribute identifies the synchronize action to match the stop synchronized action function.

(2) its_mode_scope attribute: it is important to distinguish between actions active in automatic mode (local synchronous actions) and actions active in all machine modes (even in manual operation mode)or static synchronous actions.
{'AUTOMATIC MODE','ALL MODES'}

(3) its_frequency attribute could be used to indicate when the condition has to be evaluated. If it is not

start_syncrhonized_action

its_id

its_mode_scope

its_frequency

provided, next table shows the possible values as defined for sinumerik 840d. SO the enumeration could at least include this word: {'NONE','WHENEVER','FROM','WHEN','EVERY'}

(4) its_condition attribute limits the conditions to be evaluated to the ones that can be defined in AP-238 language by now: Boolean expressions. The main problem here, the same as with the read sensor nc_functions, is how to identify ("standardize the sensor/register selection)

(5) its_action attribute must be limited to alist of nc_functions to perform data recording accions ….

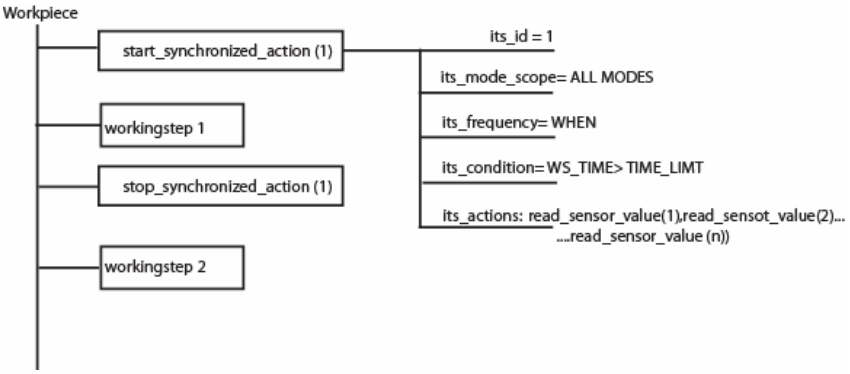
Table 2-1 Effect of frequency vocabulary words

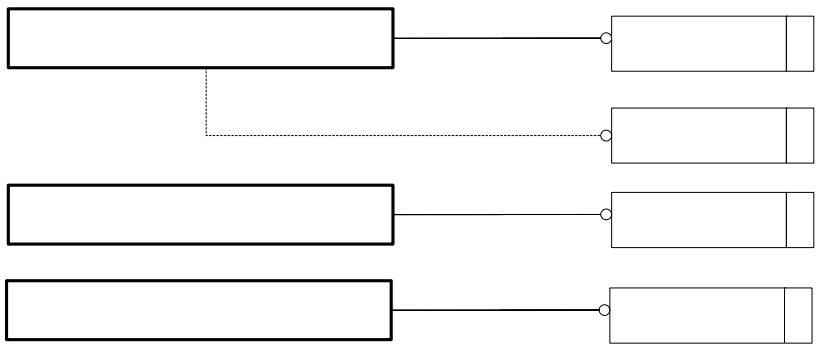
Vocabulary word	Scanning frequency
None	If no scanning frequency is programmed, then the action is executed **cyclically** in every interpolation cycle.
WHENEVER	The associated action/technology cycle is executed cyclically in every interpolation cycle provided that **the condition is fulfilled**.
FROM	If the **condition has been fulfilled once**, the action/technology cycle is executed **cyclically** in every interpolation cycle for as long as the synchronized action remains active.
WHEN	As soon as the **condition has been fulfilled**, the action/technology cycle is executed **once**. Once the action has been executed a single time, the condition is no longer checked.
EVERY	The action/technology cycle is activated **once** if the condition if fulfilled. The action/technology cycle is executed every time the **condition changes from the "FALSE" to the "TRUE" state**. In contrast to vocabulary word WHEN, checking of the condition continues after execution of the action/cycle until the synchronized action is deleted or disabled.

Example:

CAM LEVEL: for WS1, we want to check if execution time is greater than an hour and if so, record/log the X,Y,Z axis positions, the X,Y,Z drive load and set an alarm …

AP-238 High Level:





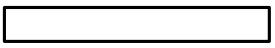
set_alarm

clear_alarm

log_event

OTHERS BLOCK I. GENERAL Functions.

LOG_COOLANT_TYPE



If it is wanted to have data about the characteristics of the coolant used.

GET_OPERATOR

The idea is to be able to set write protection (authorization changes) to machining parameters in some/all part of the AP-238 program.

1. This means having a kind of set operator authorization level **set_EDIT_acces_level** nc_function, to specifiy that from this point of the program to be able to edit the AP-238 file and make changes in the program parameters, the operator must have the appropriate authorization level.

2. This authorization level could be active for the rest of the program, or until another set_authorization_level nc_function resets or changes the programmed authorization level

3. This implies defining:

   a. Standard Access Levels for example 1-10
   b. Which are the Higher (1) and the Lower (10) access levels ?
   c. Default Level (?)
   d. How to assign operators according to their machining experience different acces levels.

With this approach for protecting change/editing the AP-238 file, the get_operator_data, could follow 2 approaches:

1. A log_operator_data, that could log operator data each time a user "logs in the HMI system".
2. A get_operator_data used to inmediatly collect data about the machining operator …

A possible data structure for the Operator_data, harmonized with person_and_organization as defined in another parts could be as follows: