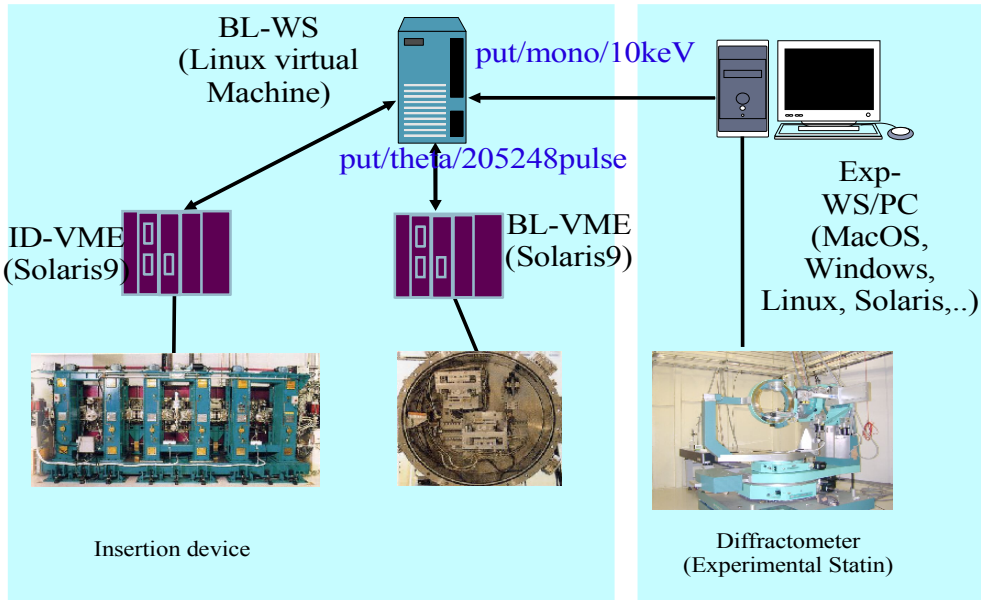


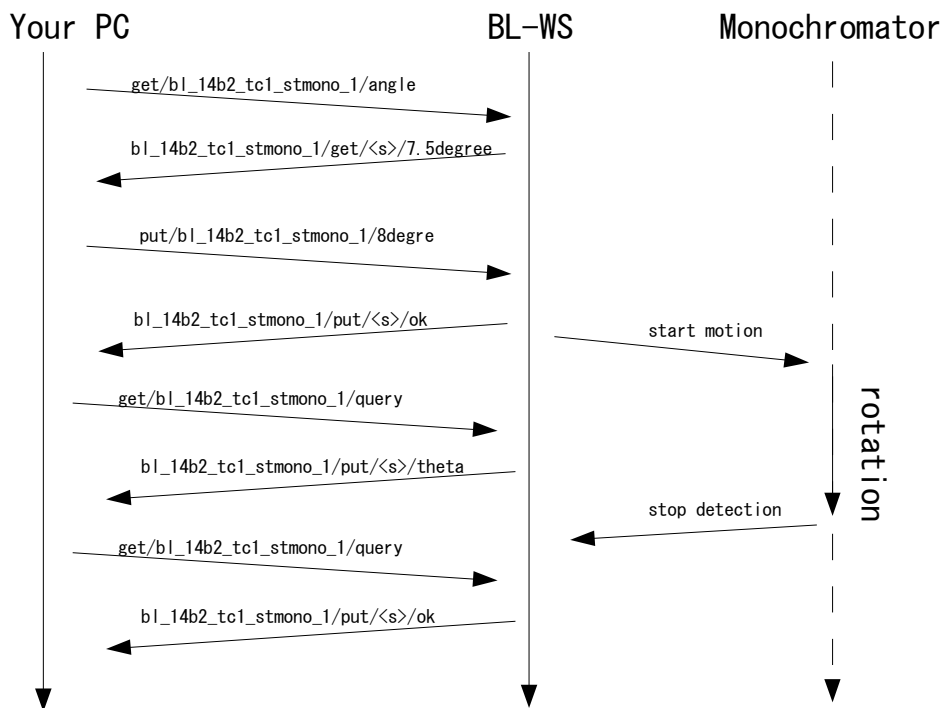
1. Beamline Control System at SPring-8

- o MADOCA (Message And Database Oriented Control Architecture) Framework
 Message Exchange based control system



Message format is S/V/O/C (usually S is automatically appended by the system).

- o Messages to control a monochromator @ BL14B2



2. Summary of messages for the monochromator and counters

V	O	C (send)	C (return)
put	bl_14b2_tc1_stmono_1	%fkev	ok / fail
put	bl_14b2_tc1_stmono_1	%fangstrome	ok / fail
put	bl_14b2_tc1_stmono_1	%fdegree	ok / fail
get	bl_14b2_tc1_stmono_1	energy	%fkev / fail
get	bl_14b2_tc1_stmono_1	wavelength	%fangstrome / fail
get	bl_14b2_tc1_stmono_1	angle	%fdegree / fail
get	bl_14b2_tc1_stmono_1	query	ok / theta / y1 / alpha1 / alpha2 / running / stoping / fail
put	bl_14b2_tc1_stmono_1	stop	ok / fail
put	bl_14b2_tc1_stmono_1	emstop	ok / fail
put	bl_14b2_tc1_stmono_1	recover	ok / fail
put	bl_14b2_tc1_stmono_1_netplane	111 / 311 /511	ok / fail
get	bl_14b2_tc1_stmono_1	netplane	si111 / si311 / si511 / fail

V	O	C (send)	C (return)
put	bl_14b2_st1_counter_1	clear	ok / fail
put	bl_14b2_st1_counter_1	%dsec	ok / fail
get	bl_14b2_st1_counter_1	query	inactive_%dcount... / counting_ %dcount... / fail
get	bl_14b2_st1_counter_1	count	%dcount_%dcount_%dcount_%dcount / fail

- Single message command for testing
tel lms <command>

Example

```
>tel lms get/bl_14b2_tc1_stmono_1/angle
tel lms::received message=bl_14b2_tc1_stmono_1/get/5400_tel lms_furukawa_vaios90s/12degree
>tel lms put/bl_14b2_tc1_stmono_1/8.1kev
tel lms::received message=bl_14b2_tc1_stmono_1/put/5408_tel lms_furukawa_vaios90s/ok
>tel lms get/bl_14b2_st1_counter_1/count
tel lms::received
message=bl_14b2_st1_counter_1/get/5414_tel lms_furukawa_vaios90s/0count_0count_0count_0cou
nt
>tel lms put/bl_14b2_st1_counter_1/clear
tel lms::received message=bl_14b2_st1_counter_1/put/5426_tel lms_furukawa_vaios90s/ok
>tel lms put/bl_14b2_st1_counter_1/1sec
tel lms::received message=bl_14b2_st1_counter_1/put/5434_tel lms_furukawa_vaios90s/ok
>tel lms get/bl_14b2_st1_counter_1/query
tel lms::received
message=bl_14b2_st1_counter_1/get/5440_tel lms_furukawa_vaios90s/counting_0count_0count_0c
ount_0count
>tel lms get/bl_14b2_st1_counter_1/query
tel lms::received
message=bl_14b2_st1_counter_1/get/5446_tel lms_furukawa_vaios90s/inactive_0count_0count_0c
ount_0count
```

Try several commands and consider message sequence to obtain XAFS spectra.

NOTE: Please do not change the netplane today.

- More details of control commands (in the SPring-8 site only)
http://madoca/for_staff_and_users/beamline/usr_svc/

3. Python

- Before starting communication with the MS, you need call `py2ms.open("<app_name>");`
- You can send/receive messages using following functions
 - `py2ms.send("<msg to be sent>")`
 - `py2ms.recv();`
- You have to close communication before terminating your program using
 - `py2ms.close();`
- Example
 - `import py2ms`
 - `py2ms.open("test");`
 - `py2ms.send("get/bl_14b2_tc1_stmono_1/energy");`
 - `recv = ms_recv();`
 - `print recv;`
 - `py2ms.close();`
- You can obtain each part of the message (S/V/O/C) by splitting the message using `string.split()` function
 - `import string`
 - `recv = ms_recv();`
 - `re = string.split(recv, "/");`
 - `print "S=%s V=%s O=%s C=%s" % {re[0], re[1], re[2], re[3]};`

Try to make a XAFS measurement program using python.

4. Command Interpreter(CI)

- CI is used to control individual axis of the beamline components like the monochromator's two theta axis)
- The CI refers a configuration file which describes how an abstract message is interpreted and how to decompose it into primitive messages.
- You can describe the sequence to obtain XAFS spectra in the configuration file.
- Example

```
1: SUBSECTION FUNCTION
2: apply put/count_%dsec_%dtimes
3:
4: function $i [ 0 ]
5: while [ $i < target(2) ]
6:   function bl_14b2_st1_counter_1 [ "clear" ]
7:   function bl_14b2_st1_counter_1 [ target(1) ] sec
8:   sequence
9:   print [ count_1 ] counts
10:  print [ new_line() ]
11:  function $i [ $i + 1 ]
12: endwhile
13: ENDSUBSECTION
```

- 1: Declaration of new abstract message section
- 2: Abstract message, %d, %f and %s denote a integer number, real number and string, respectively.
- 4: Initialization of internal variable. "\$" denotes an internal variable.
- 6: send "put/bl_14b2_st1_counter_1/clear"
- 7: send "put/bl_14b2_st1_counter_1/1sec", time length is described in the second part of C in the abstract message from client.
- 8: Wait for counting end
- 9: Display on screen result. The variable "count_1" is described in other section.

Try to make a XAFS measurement sequence into the configuration file.