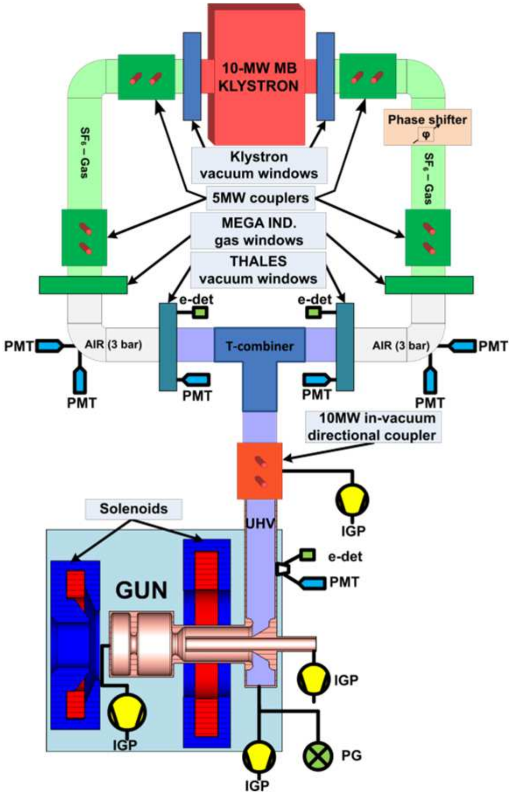
SMAC

The **S**tate **M**achine based **A**utomatic **C**onditioning application for PITZ

Introduction

- The Photo Injector Test Facility at DESY in Zeuthen **(PITZ)** was built to test and to optimize high brightness electron sources for Free-Electron Lasers (FELs) (e.g. FLASH, XFEL).
- In order to achieve high accelerating gradients and long RF pulse lengths in the RF gun cavities, an extensive and safe RF conditioning is required.
- A State Machine based Automatic Conditioning application **(SMAC)** was developed to automate the RF conditioning processes, allowing for greater efficiency and performance optimization.



The PITZ RF setup

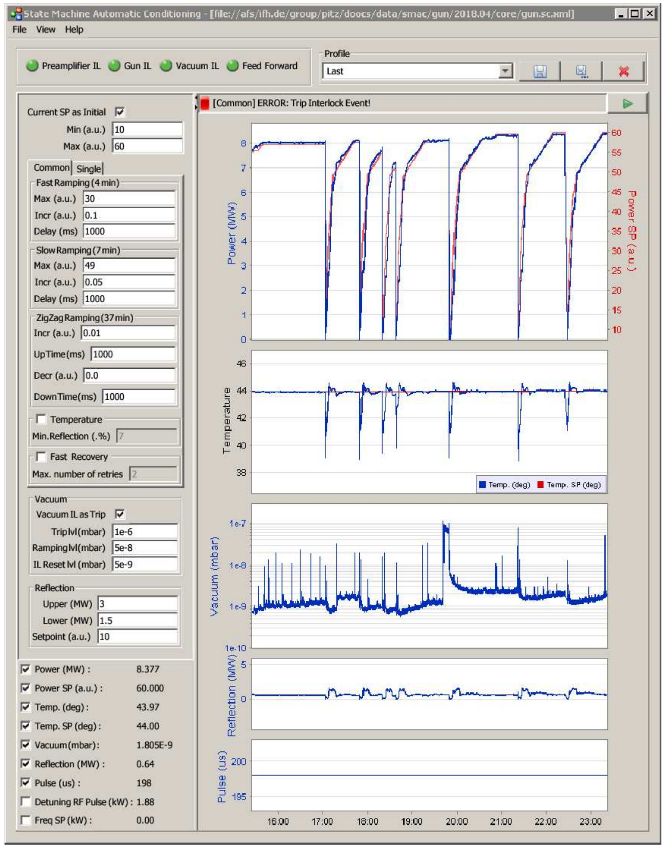
SMAC History

- The Automatic Conditioning Program (ACP) was originally implemented by Michael Winde as a DOOCS server + LabView GUI
- In 2009 I started a new project SMAC with the following main goals
  - Provide users with the ability to change logic w/o having to recompile the application.
  - Provide users with a more user-friendly graphical interface.
  - Add support for single running instance.
  - Multiplatform support, ease of maintenance, extensibility and integration into the PITZ control system.
  - High performance (using multithreading, event-driven approaches).
  - The first release and use at PITZ took place in 2010.

SMAC History

- The first version only included RF Ramping (for Gun) and Magnet Sweeping algorithms.
- It was later extended with support for Booster and T-Combiner cavities.
- In 2012, it was extended with a temperature control algorithm based on an implementation done by Marek Otevrel (originally in MatLab).
- In 2015, it was extended with the “cold-startup” algorithm implemented by Yves Renier (originally in MatLab)

SMAC History



State Machine Automatic Conditioning - [file://afs/ifh.de/group/pitz/doocs/data/smac/gun/2018.04/core/gun.sc.xml]

Profile: Last

[Common] ERROR: Trip Interlock Event!

Current SP as Initial ☑
Min (a.u.) 10
Max (a.u.) 60

Common | Single

Fast Ramping (4 min)
Max (a.u.) 30
Incr (a.u.) 0.1
Delay (ms) 1000

Slow Ramping (7 min)
Max (a.u.) 49
Incr (a.u.) 0.05
Delay (ms) 1000

ZigZag Ramping (37 min)
Incr (a.u.) 0.01
UpTime(ms) 1000
Decr (a.u.) 0.0
DownTime(ms) 1000

☐ Temperature
Min.Reflection (.%) 7

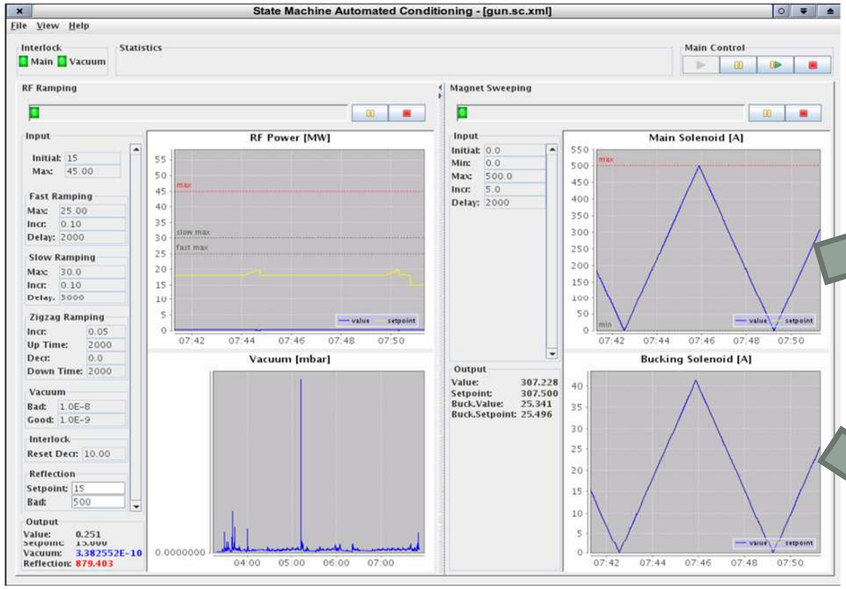
☐ Fast Recovery
Max. number of retries 2

Vacuum
Vacuum IL as Trip ☑
Trip lvl(mbar) 1e-6
Ramping lvl(mbar) 5e-8
IL Reset lvl (mbar) 5e-9

Reflection
Upper (MW) 3
Lower (MW) 1.5
Setpoint (a.u.) 10

☑ Power (MW) : 8.377
☑ Power SP (a.u.) : 60.000
☑ Temp. (deg) : 43.97
☑ Temp. SP (deg) : 44.00
☑ Vacuum(mbar): 1.805E-9
☑ Reflection (MW) : 0.64
☑ Pulse (us) : 198
☐ Detuning RF Pulse (kW) : 1.88
☐ Freq SP (kW) : 0.00

latest version, SMAC for Gun



State Machine Automated Conditioning - [gun.sc.xml]

RF Ramping

Input
Initial: 15
Max: 45.00

Fast Ramping
Max: 25.00
Incr: 0.10
Delay: 2000

Slow Ramping
Max: 30.0
Incr: 0.10
Delay: 3000

Zigzag Ramping
Incr: 0.05
Up Time: 2000
Decr: 0.0
Down Time: 2000

Vacuum
Bad: 1.0E-8
Good: 1.0E-9

Interlock
Reset Decr: 10.00

Reflection
Setpoint: 15
Bad: 500

Output
Value: 0.251
Setpoint: 15.000
Vacuum: 3.382552E-10
Reflection: 879.403

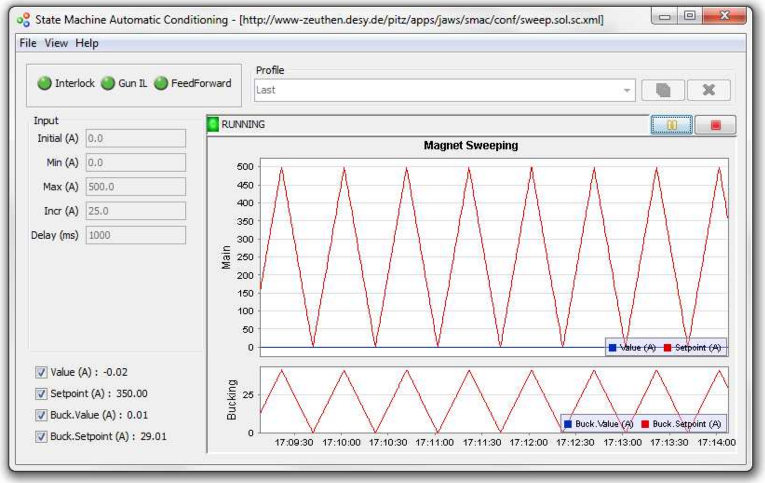
Magnet Sweeping

Input
Initial: 0.0
Min: 0.0
Max: 500.0
Incr: 5.0
Delay: 2000

Output
Value: 307.228
Setpoint: 307.500
Buck.Value: 25.341
Buck.Setpoint: 25.496

first version

latest version, SMAC for Solenoid



State Machine Automatic Conditioning - [http://www-zeuthen.desy.de/pitz/apps/jaws/smac/conf/sweep.sol.sc.xml]

Profile: Last

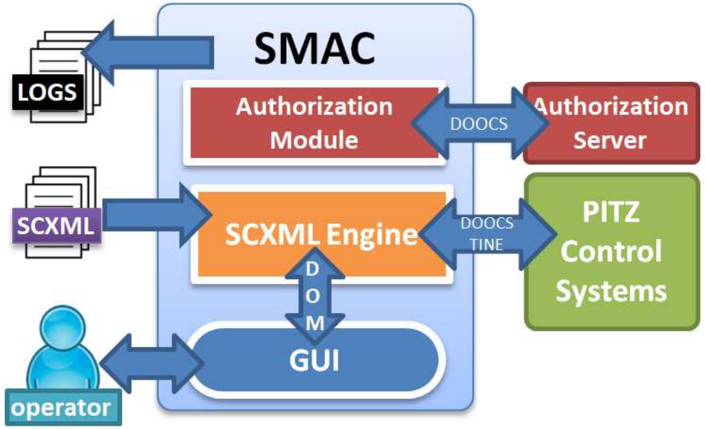
Input
Initial (A) 0.0
Min (A) 0.0
Max (A) 500.0
Incr (A) 25.0
Delay (ms) 1000

RUNNING

☑ Value (A) : -0.02
☑ Setpoint (A) : 350.00
☑ Buck.Value (A) : 0.01
☑ Buck.Setpoint (A) : 29.01

Technical Description

- SMAC is written in Java and uses State Chart XML (SCXML) as the finite-state machine execution environment based on Harel state-charts.
- Application employs the DOOCS and TINE for the communication with the control systems of PITZ.

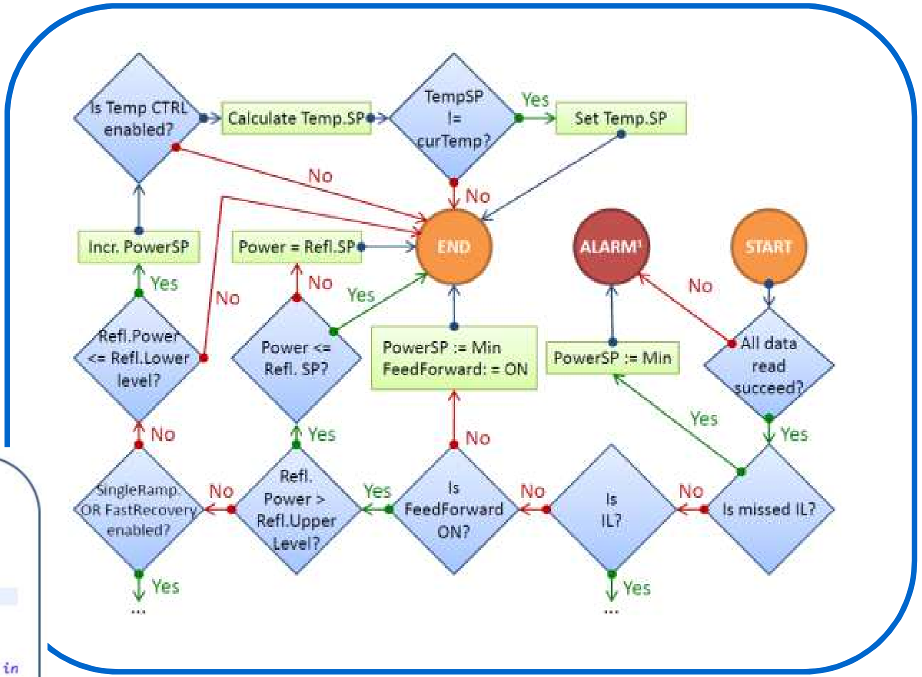


The overall structure and data flow of SMAC application.

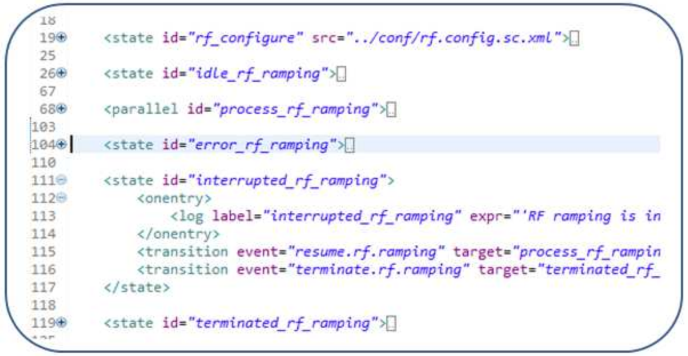
- Communication between GUI and SCXML processing layer (state machine) is performed via Document Object Model (DOM) events.
- The authorization module guaranties that only one instance of SMAC is working at the same time.

SMAC: SCXML

- SCXML engine capable of executing a state machine defined using a SCXML document that describes the application flow.
- Specified by W3C the SCXML is becoming a standard way to represent state charts in XML.
- The whole application flow is visible by looking at the SCXML files.



Segment of the RF ramping flow diagram.

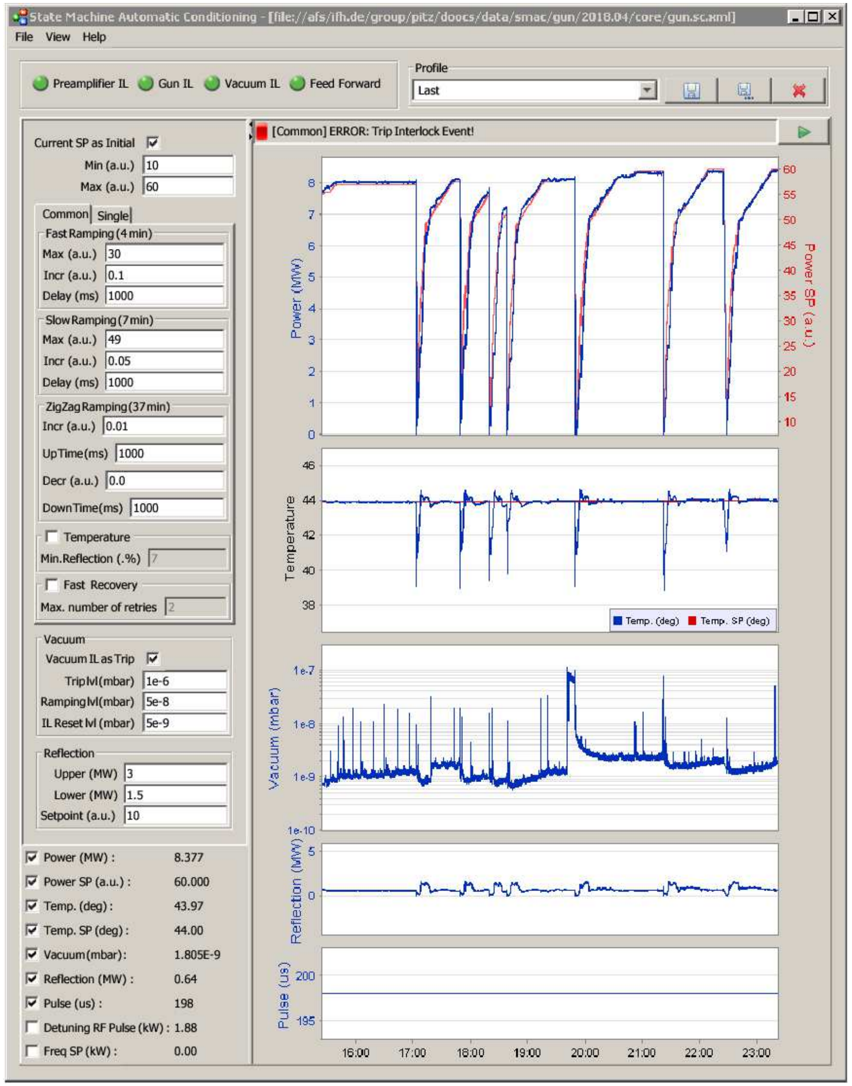


```
<state id="rf_configure" src="../conf/rf.config.sc.xml">

<state id="idle_rf_ramping">

<parallel id="process_rf_ramping">

<state id="error_rf_ramping">

<state id="interrupted_rf_ramping">
    <onentry>
        <log label="interrupted_rf_ramping" expr="'RF ramping is in
    </onentry>
    <transition event="resume.rf.ramping" target="process_rf_rampin
    <transition event="terminate.rf.ramping" target="terminated_rf_
</state>

<state id="terminated_rf_ramping">
```

Segment of SCXML code.

SMAC: GUI

- The GUI is created by using the Java Swing toolkit.
- The GUI provides the user control of the conditioning process and relevant monitoring data.
- Error and warning messages are constantly displayed giving the user information on the current status (e.g. anomalous forward power, communications error).
- The profile panel allows operator to pre-configure the conditioning settings in order to quickly apply them to a new run.



State Machine Automatic Conditioning - [file://afs/ifh.de/group/pitz/doocs/data/smac/gun/2018.04/core/gun.sc.xml]

● Preamplifier IL ● Gun IL ● Vacuum IL ● Feed Forward

Profile: Last

[Common] ERROR: Trip Interlock Event!

Current SP as Initial ☑
Min (a.u.) 10
Max (a.u.) 60

Common | Single

Fast Ramping (4 min)
Max (a.u.) 30
Incr (a.u.) 0.1
Delay (ms) 1000

Slow Ramping (7 min)
Max (a.u.) 49
Incr (a.u.) 0.05
Delay (ms) 1000

ZigZag Ramping (37 min)
Incr (a.u.) 0.01
UpTime(ms) 1000
Decr (a.u.) 0.0
DownTime(ms) 1000

☐ Temperature
Min.Reflection (.%) 7

☐ Fast Recovery
Max. number of retries 2

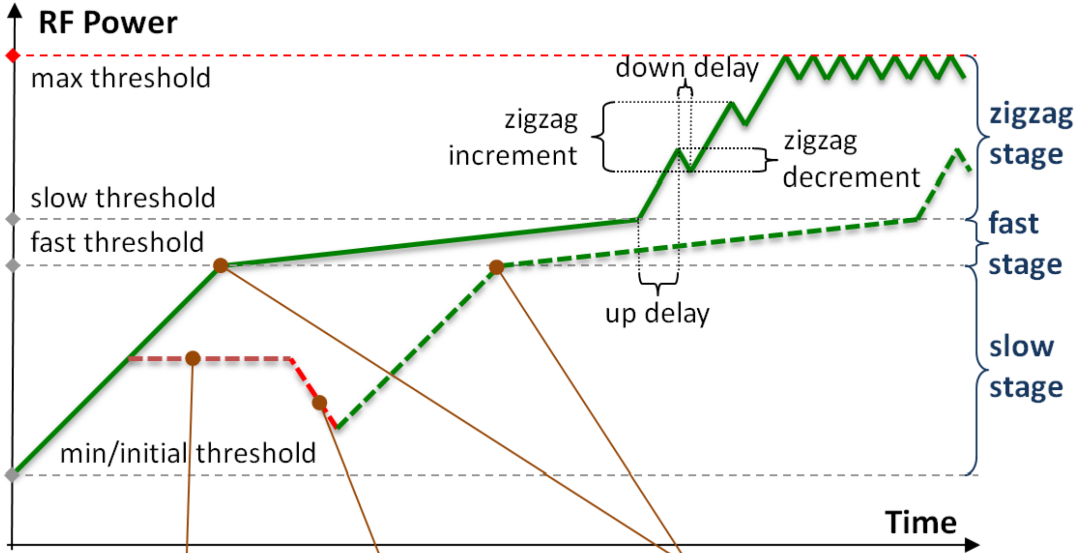
Vacuum
Vacuum IL as Trip ☑
Trip lvl(mbar) 1e-6
Ramping lvl(mbar) 5e-8
IL Reset lvl (mbar) 5e-9

Reflection
Upper (MW) 3
Lower (MW) 1.5
Setpoint (a.u.) 10

☑ Power (MW) : 8.377
☑ Power SP (a.u.) : 60.000
☑ Temp. (deg) : 43.97
☑ Temp. SP (deg) : 44.00
☑ Vacuum(mbar): 1.805E-9
☑ Reflection (MW) : 0.64
☑ Pulse (us) : 198
☐ Detuning RF Pulse (kW) : 1.88
☐ Freq SP (kW) : 0.00

Conditioning strategy

Common mode of the RF power ramping: consists of several stages (“fast”, “slow” and “zigzag”) with different RF ramping speeds.



- vacuum level is too low
- reflected power level is too high

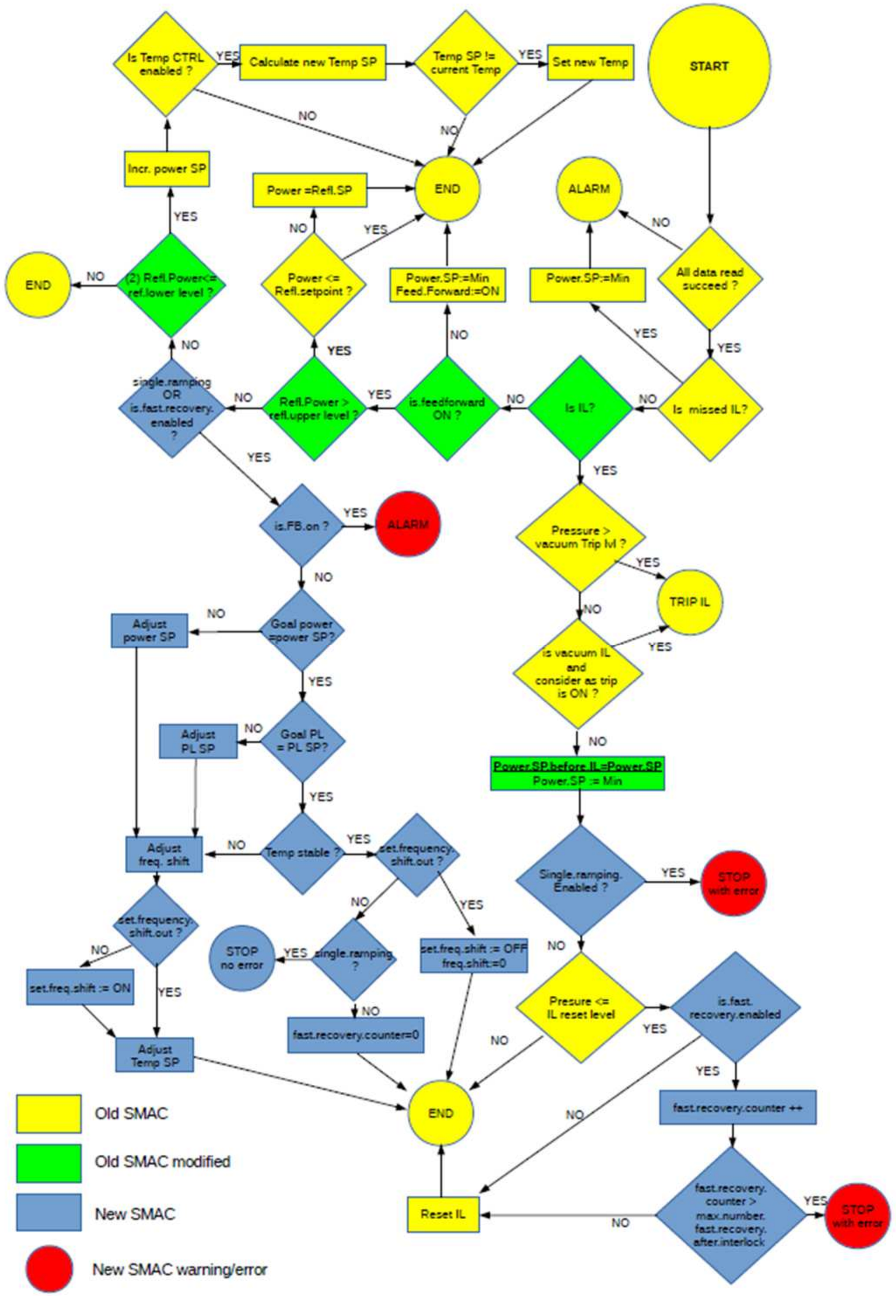
- interlock event
- reflected power level is too high

- regular operation

- The conditioning algorithm consists of gradually increasing the RF power and the RF pulse length but keeping a low rate of vacuum spikes in the cavity in order to prevent any damage from break-downs.
- Currently, SMAC implements two ramping modes, namely, **Single** and **Common**.

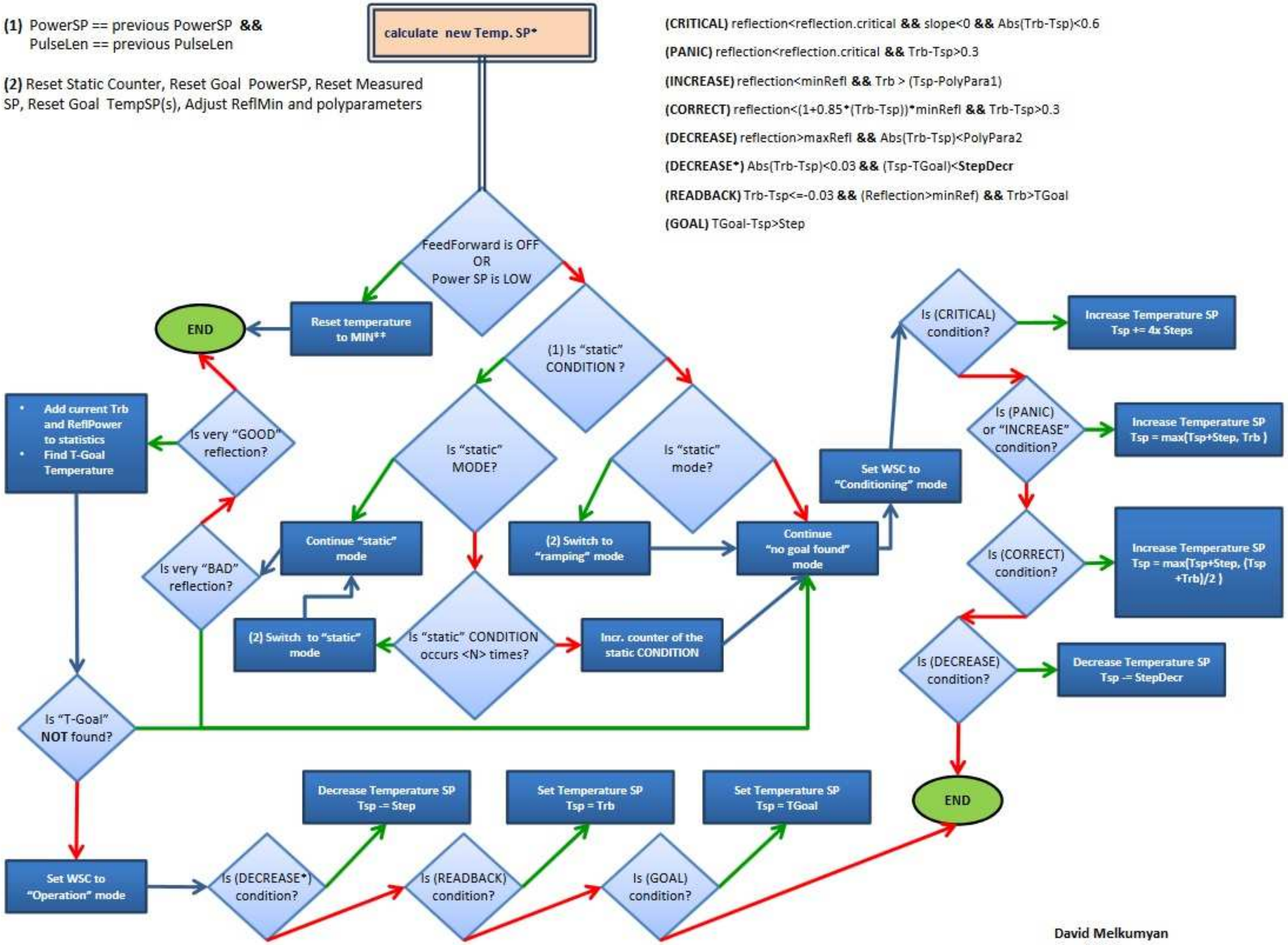
Flowchart

RF Ramping
Simplified version



Flowchart Temperature Control

Simplified version



(1) PowerSP == previous PowerSP **&&**
PulseLen == previous PulseLen

(2) Reset Static Counter, Reset Goal PowerSP, Reset Measured SP, Reset Goal TempSP(s), Adjust ReflMin and polyparameters

(CRITICAL) reflection<reflection.critical **&&** slope<0 **&&** Abs(Trb-Tsp)<0.6

(PANIC) reflection<reflection.critical **&&** Trb-Tsp>0.3

(INCREASE) reflection<minRefl **&&** Trb > (Tsp-PolyPara1)

(CORRECT) reflection<(1+0.85*(Trb-Tsp))*minRefl **&&** Trb-Tsp>0.3

(DECREASE) reflection>maxRefl **&&** Abs(Trb-Tsp)<PolyPara2

(DECREASE*) Abs(Trb-Tsp)<0.03 **&&** (Tsp-TGoal)<**StepDecr**

(READBACK) Trb-Tsp<=-0.03 **&&** (Reflection>minRef) **&&** Trb>TGoal

(GOAL) TGoal-Tsp>Step

calculate new Temp. SP*

FeedForward is OFF
OR
Power SP is LOW

Reset temperature
to MIN**

END

(1) Is "static"
CONDITION ?

Is "static"
MODE?

Is "static"
mode?

Continue "static"
mode

(2) Switch to
"ramping" mode

Continue
"no goal found"
mode

Set WSC to
"Conditioning" mode

Is very "GOOD"
reflection?

- Add current Trb and ReflPower to statistics
- Find T-Goal Temperature

Is very "BAD"
reflection?

(2) Switch to "static"
mode

Is "static" CONDITION
occurs <N> times?

Incr. counter of the
static CONDITION

Is "T-Goal"
NOT found?

Set WSC to
"Operation" mode

Is (DECREASE*)
condition?

Decrease Temperature SP
Tsp -= Step

Is (READBACK)
condition?

Set Temperature SP
Tsp = Trb

Is (GOAL)
condition?

Set Temperature SP
Tsp = TGoal

Is (CRITICAL)
condition?

Increase Temperature SP
Tsp += 4x Steps

Is (PANIC)
or "INCREASE"
condition?

Increase Temperature SP
Tsp = max(Tsp+Step, Trb)

Is (CORRECT)
condition?

Increase Temperature SP
Tsp = max(Tsp+Step, (Tsp +Trb)/2)

Is (DECREASE)
condition?

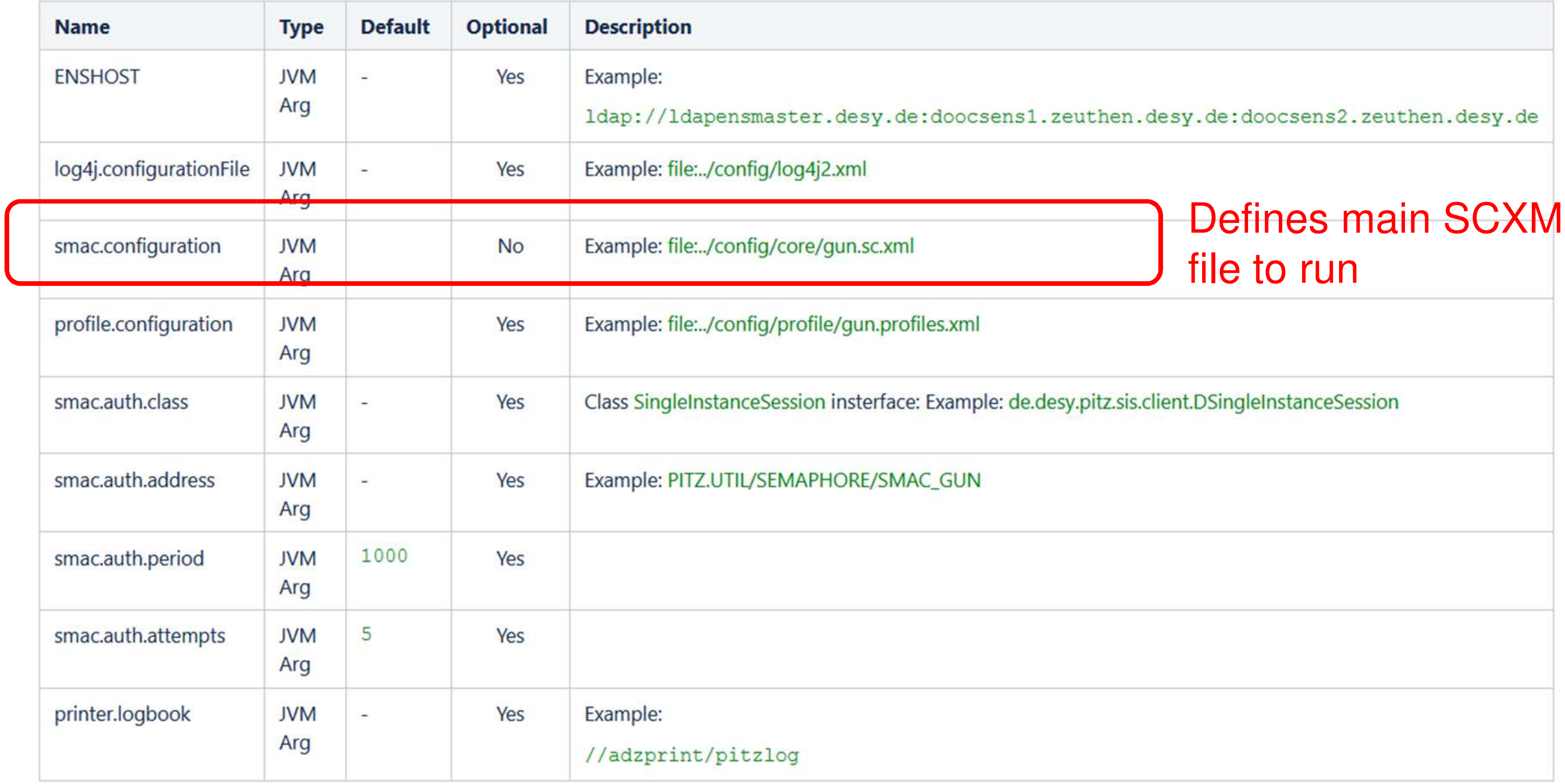
Decrease Temperature SP
Tsp -= StepDecr

END

David Melkumyan
Apr. 2014

Configuration

- Application settings (via Java VM arguments)



Name	Type	Default	Optional	Description
ENSHOST	JVM Arg	-	Yes	Example: `ldap://ldapensmaster.desy.de:doocsens1.zeuthen.desy.de:doocsens2.zeuthen.desy.de`
log4j.configurationFile	JVM Arg	-	Yes	Example: file:../config/log4j2.xml
smac.configuration	JVM Arg		No	Example: file:../config/core/gun.sc.xml
profile.configuration	JVM Arg		Yes	Example: file:../config/profile/gun.profiles.xml
smac.auth.class	JVM Arg	-	Yes	Class SingleInstanceSession interface: Example: de.desy.pitz.sis.client.DSingleInstanceSession
smac.auth.address	JVM Arg	-	Yes	Example: PITZ.UTIL/SEMAPHORE/SMAC_GUN
smac.auth.period	JVM Arg	1000	Yes	
smac.auth.attempts	JVM Arg	5	Yes	
printer.logbook	JVM Arg	-	Yes	Example: `//adzprint/pitzlog`

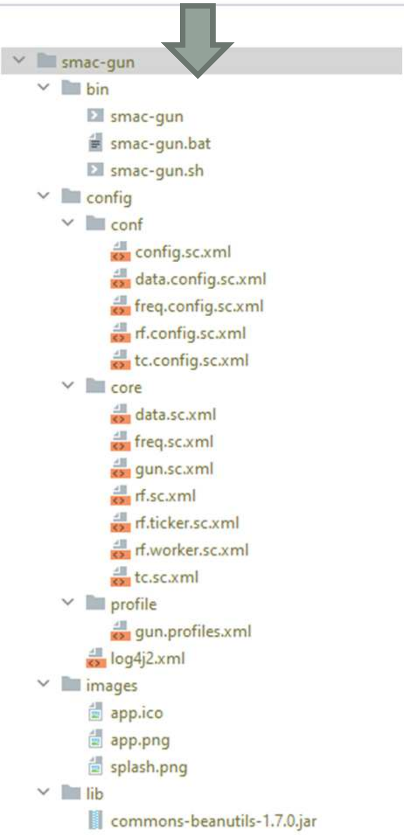
Defines main SCXML file to run

Installation

Download URLs:

SMAC for GUN

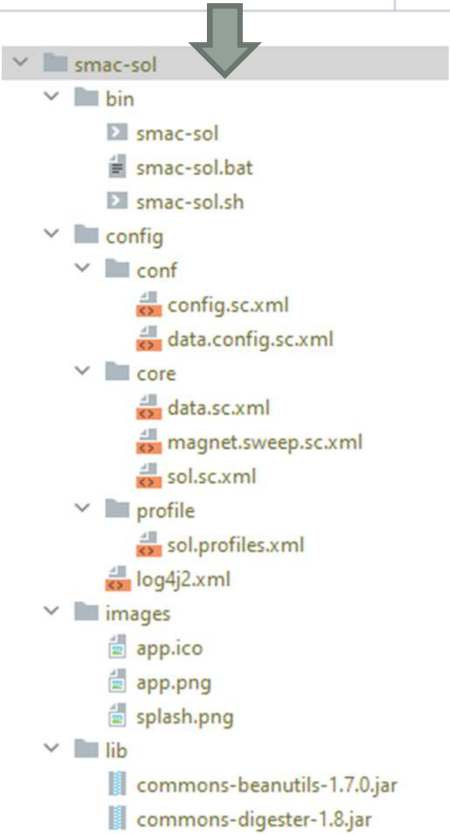
Windows	smac-gun-2021.04.zip	ZIP file
Linux	smac-gun-2021.04.tgz	TAR file
RPM	smac-gun-2021.04-06.x86_64.rpm	RPM package
Java FAT JAR	smac-gun-2021.04-all.jar	Java Fat Jar



```
smac-gun
  bin
    smac-gun
    smac-gun.bat
    smac-gun.sh
  config
    conf
      config.sc.xml
      data.config.sc.xml
      freq.config.sc.xml
      rf.config.sc.xml
      tc.config.sc.xml
    core
      data.sc.xml
      freq.sc.xml
      gun.sc.xml
      rf.sc.xml
      rf.ticker.sc.xml
      rf.worker.sc.xml
      tc.sc.xml
    profile
      gun.profiles.xml
    log4j2.xml
  images
    app.ico
    app.png
    splash.png
  lib
    commons-beanutils-1.7.0.jar
```

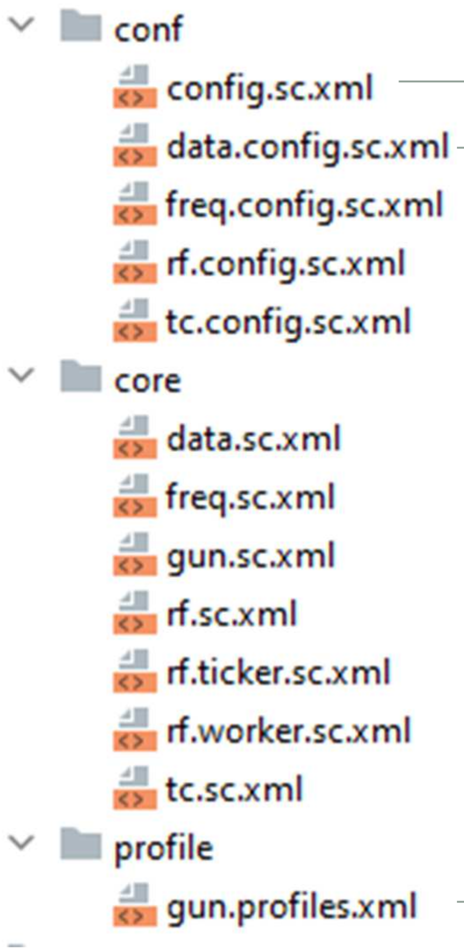
SMAC for Magnets

Windows	smac-sol-2021.10.zip	ZIP file
Linux	smac-sol-2021.10.tgz	TAR file
RPM	smac-sol-2021.10-22.x86_64.rpm	RPM package
Java FAT JAR	smac-sol-2021.10-all.jar	Java Fat Jar



```
smac-sol
  bin
    smac-sol
    smac-sol.bat
    smac-sol.sh
  config
    conf
      config.sc.xml
      data.config.sc.xml
    core
      data.sc.xml
      magnet.sweep.sc.xml
      sol.sc.xml
    profile
      sol.profiles.xml
    log4j2.xml
  images
    app.ico
    app.png
    splash.png
  lib
    commons-beanutils-1.7.0.jar
    commons-digester-1.8.jar
```

Configuration



- conf
  - config.sc.xml
  - data.config.sc.xml
  - freq.config.sc.xml
  - rf.config.sc.xml
  - tc.config.sc.xml
- core
  - data.sc.xml
  - freq.sc.xml
  - gun.sc.xml
  - rf.sc.xml
  - rf.ticker.sc.xml
  - rf.worker.sc.xml
  - tc.sc.xml
- profile
  - gun.profiles.xml

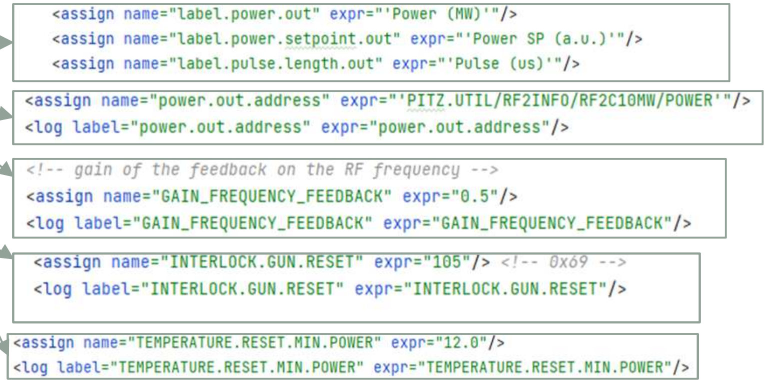
Labels for UI elements

Data points

Freq. control settings

RF control settings

Temp. control settings



```
    <assign name="label.power.out" expr="'Power (MW)'"/>
    <assign name="label.power.setpoint.out" expr="'Power SP (a.u.)'"/>
    <assign name="label.pulse.length.out" expr="'Pulse (us)'"/>
```

```
<assign name="power.out.address" expr="'PITZ.UTIL/RF2INFO/RF2C10MW/POWER'"/>
<log label="power.out.address" expr="power.out.address"/>
```

```
<!-- gain of the feedback on the RF frequency -->
<assign name="GAIN_FREQUENCY_FEEDBACK" expr="0.5"/>
<log label="GAIN_FREQUENCY_FEEDBACK" expr="GAIN_FREQUENCY_FEEDBACK"/>
```

```
 <assign name="INTERLOCK.GUN.RESET" expr="105"/> <!-- 0x69 -->
 <log label="INTERLOCK.GUN.RESET" expr="INTERLOCK.GUN.RESET"/>
```

```
<assign name="TEMPERATURE.RESET.MIN.POWER" expr="12.0"/>
<log label="TEMPERATURE.RESET.MIN.POWER" expr="TEMPERATURE.RESET.MIN.POWER"/>
```

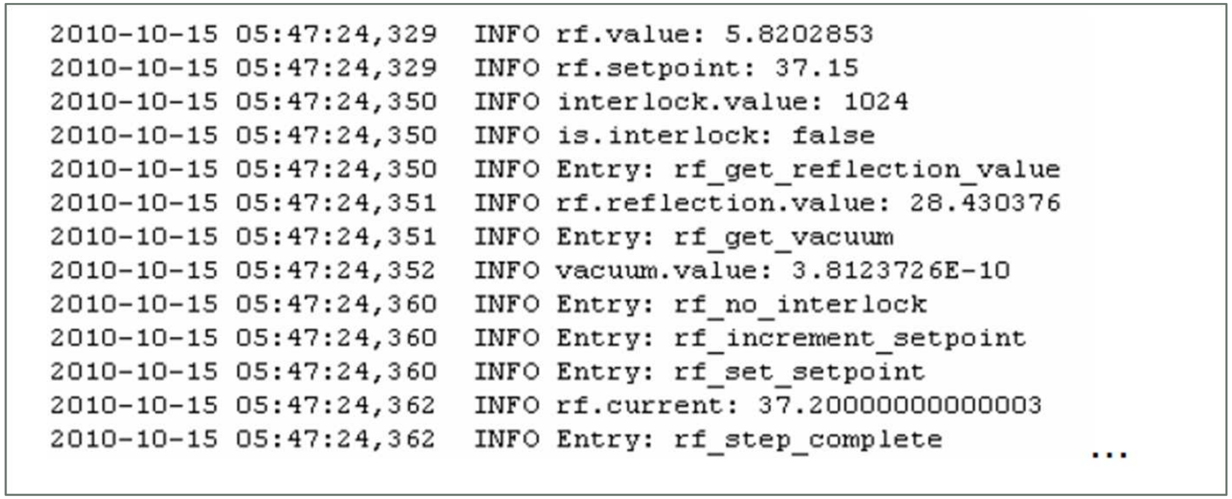
State machine configuration

Profiles (user settings)

Logging

Detailed information about conditioning process available also from log files

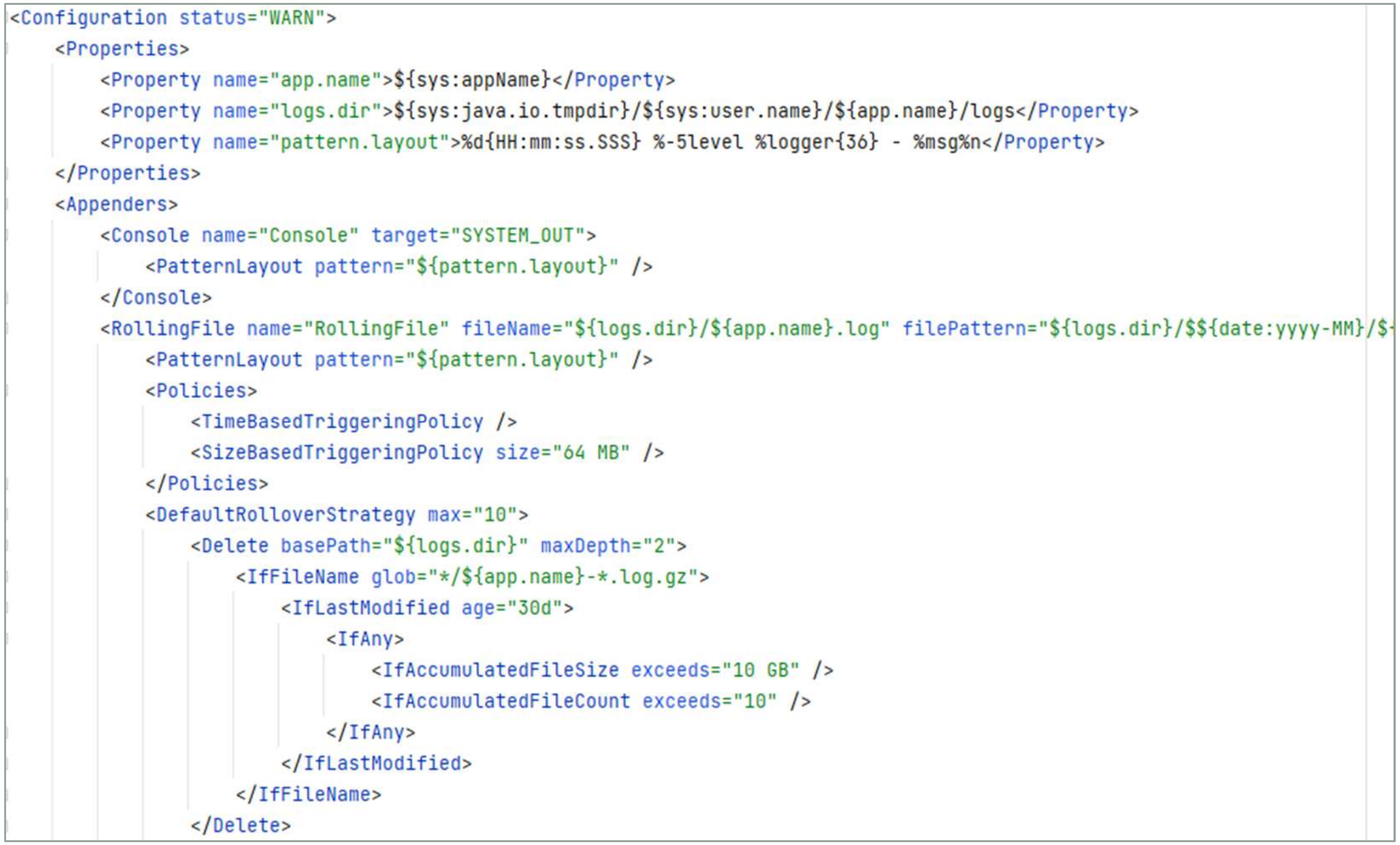
- millisecond precision
- Automatically archived (zip files)
- Contains data for up to several months of operation (depends on the logging level)
- Has different (configurable) levels of details (ERROR, WARN, INFO, DEBUG, TRACE)
- Very useful for debugging and troubleshooting (especially during development phase)



```
2010-10-15 05:47:24,329  INFO rf.value: 5.8202853
2010-10-15 05:47:24,329  INFO rf.setpoint: 37.15
2010-10-15 05:47:24,350  INFO interlock.value: 1024
2010-10-15 05:47:24,350  INFO is.interlock: false
2010-10-15 05:47:24,350  INFO Entry: rf_get_reflection_value
2010-10-15 05:47:24,351  INFO rf.reflection.value: 28.430376
2010-10-15 05:47:24,351  INFO Entry: rf_get_vacuum
2010-10-15 05:47:24,352  INFO vacuum.value: 3.8123726E-10
2010-10-15 05:47:24,360  INFO Entry: rf_no_interlock
2010-10-15 05:47:24,360  INFO Entry: rf_increment_setpoint
2010-10-15 05:47:24,360  INFO Entry: rf_set_setpoint
2010-10-15 05:47:24,362  INFO rf.current: 37.20000000000003
2010-10-15 05:47:24,362  INFO Entry: rf_step_complete
                                                             ...
```

Logging Configuration

./conf/log4j.xml



```
<Configuration status="WARN">
    <Properties>
        <Property name="app.name">${sys:appName}</Property>
        <Property name="logs.dir">${sys:java.io.tmpdir}/${sys:user.name}/${app.name}/logs</Property>
        <Property name="pattern.layout">%d{HH:mm:ss.SSS} %-5level %logger{36} - %msg%n</Property>
    </Properties>
    <Appenders>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="${pattern.layout}" />
        </Console>
        <RollingFile name="RollingFile" fileName="${logs.dir}/${app.name}.log" filePattern="${logs.dir}/$${date:yyyy-MM}/$
            <PatternLayout pattern="${pattern.layout}" />
            <Policies>
                <TimeBasedTriggeringPolicy />
                <SizeBasedTriggeringPolicy size="64 MB" />
            </Policies>
            <DefaultRolloverStrategy max="10">
                <Delete basePath="${logs.dir}" maxDepth="2">
                    <IfFileName glob="*/${app.name}-*.log.gz">
                        <IfLastModified age="30d">
                            <IfAny>
                                <IfAccumulatedFileSize exceeds="10 GB" />
                                <IfAccumulatedFileCount exceeds="10" />
                            </IfAny>
                        </IfLastModified>
                    </IfFileName>
                </Delete>
```

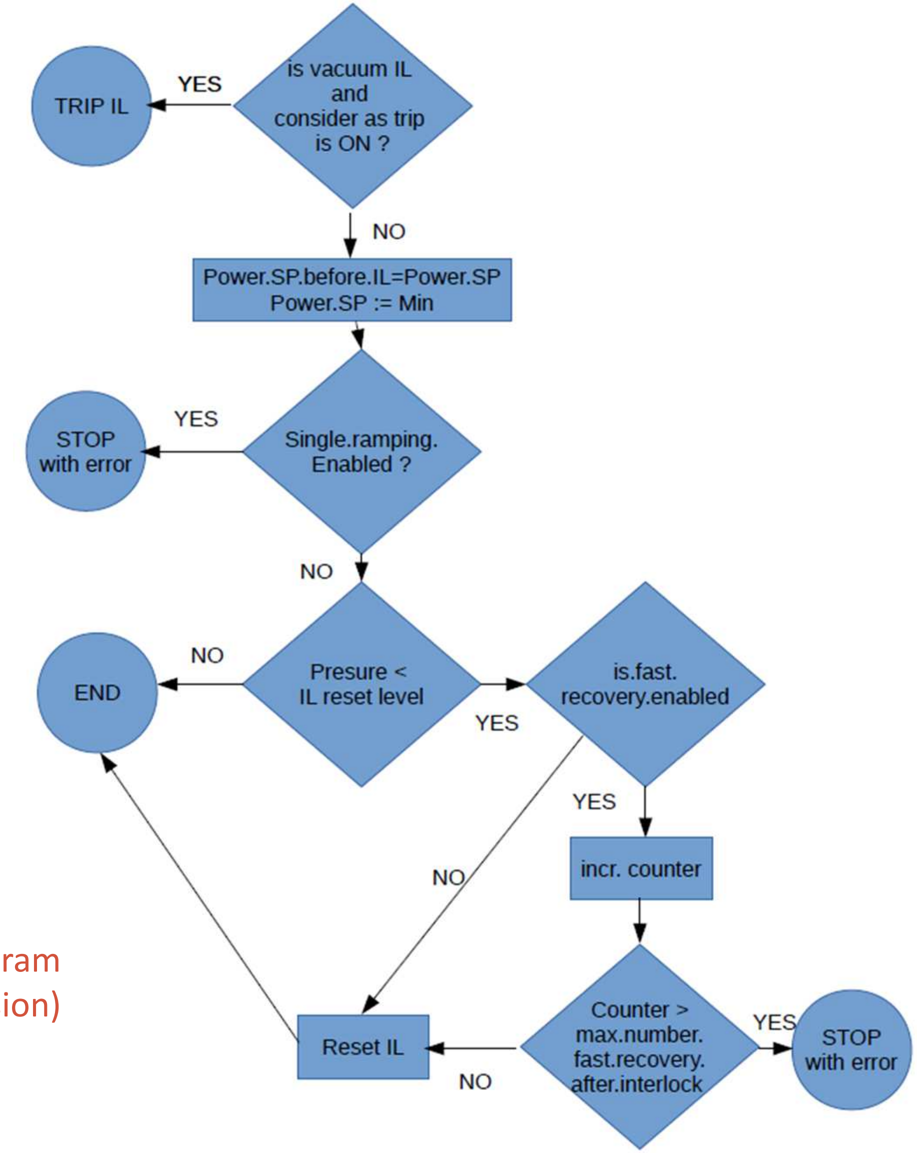
CONCLUSIONS

- The SMAC implementation was intended as a proof of concept, applying a **state-chart approach** for the automatic conditioning.
- The SMAC was brought into operation in 2010 and has been used at PITZ very successfully.
- Since then the application is left to run unattended overnight.
- The SMAC continues to be improved by feedbacks and suggestions from the physicists.

Demo

Backup slides

Backup slides


is vacuum IL and consider as trip is ON ?
YES
TRIP IL
NO
Power.SP.before.IL=Power.SP
Power.SP := Min
Single.ramping. Enabled ?
YES
STOP with error
NO
Presure < IL reset level
NO
END
YES
is.fast. recovery.enabled
YES
NO
incr. counter
Counter > max.number. fast.recovery. after.interlock
YES
STOP with error
NO
Reset IL

Reset IL diagram
(simplified version)