

DAQミドルウェアの概要

2008年8月5・6日

DAQ 実習 - DAQ-Middleware

KEKエレクトロニクスシステムグループ

KEK測定器開発室DAQプロジェクト

安 芳次

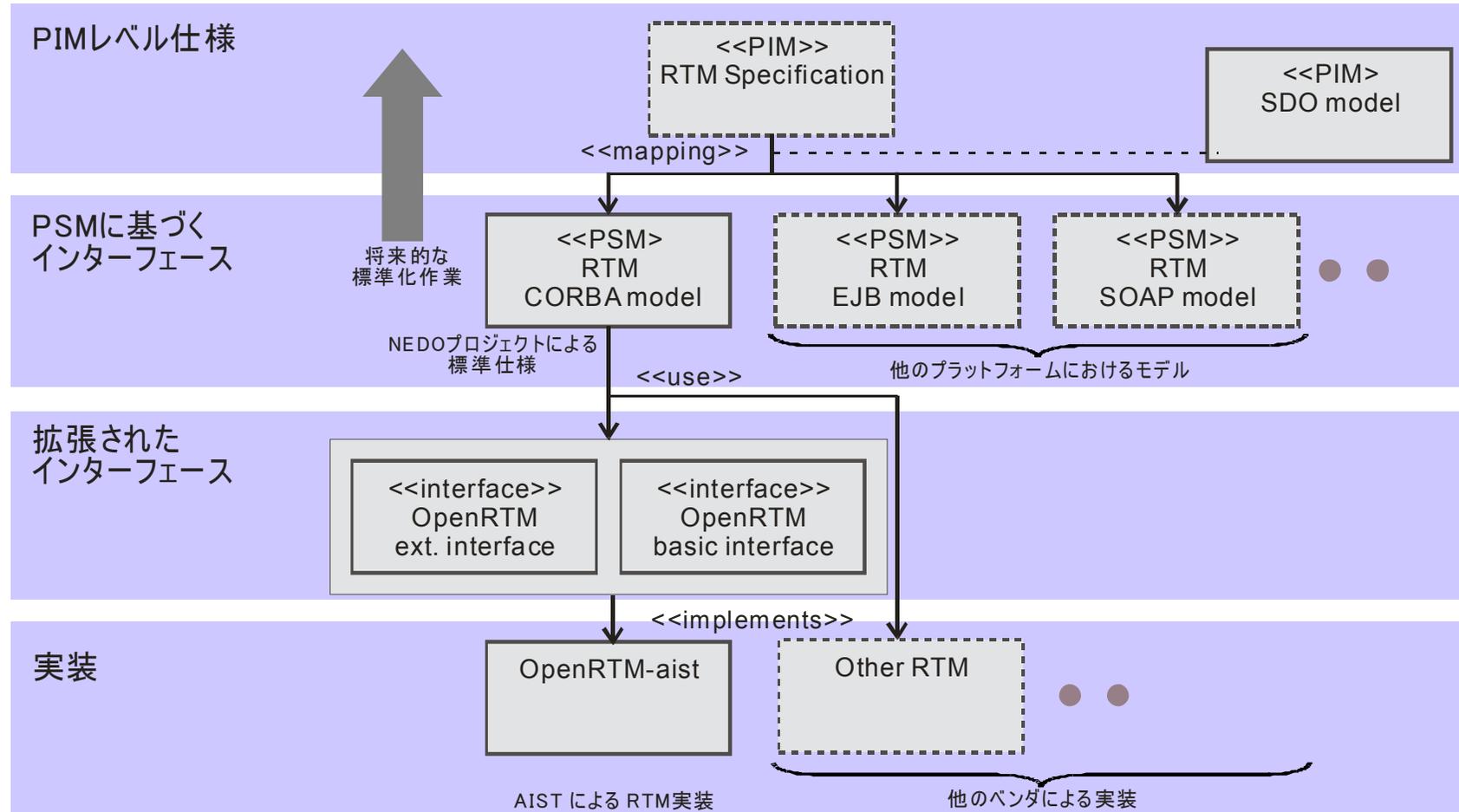
DAQミドルウェアの狙い

- 多数の計算機を使ったネットワーク分散型DAQソフトウェア
- ソフトウェアの汎用化
 - 国際標準ロボット技術(RTミドルウェア)の利用
 - XMLによるデータ収集パラメータの記述
- 複雑さの隠蔽
 - パッケージソフトウェア化(簡単インストール)
 - 通信・同期・マルチスレッドなどの隠蔽
 - 自動化による簡単操作

RTミドルウェアとは？

- Object Management Group(OMG)による国際標準ロボット技術
- ロボット分野に限らず、自律し協調したコンポーネントを必要とする組み込み系を含むソフトウェアのインフラを提供する
- 日本では産総研が標準化を主導し、OpenRTM-aistパッケージを開発している
<http://www.is.aist.go.jp/rt/OpenRTM-aist/html/index.html>

OMG のモデリング

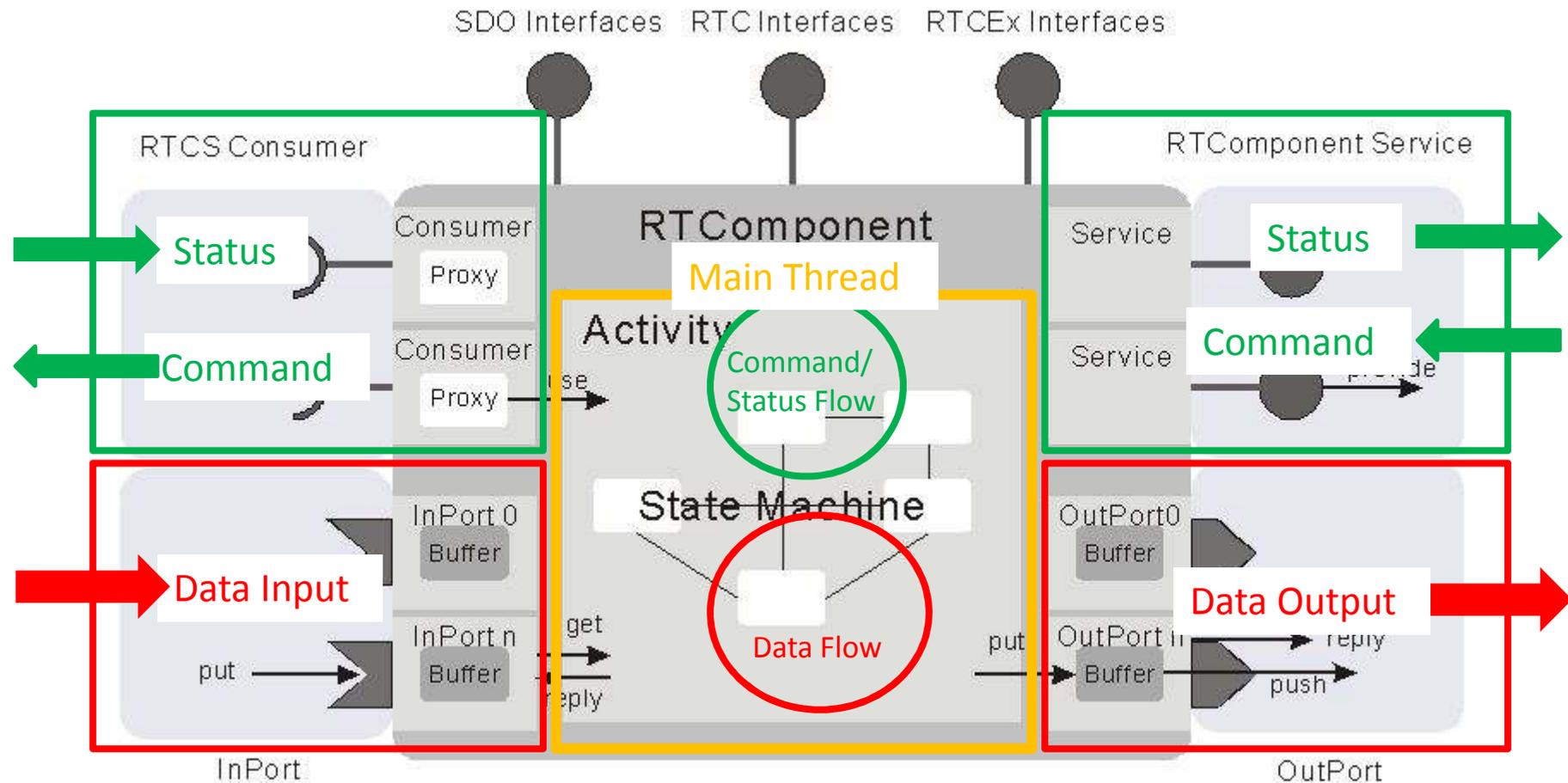


PIM: Platform Independent Model

PSM: Platform Specific Model



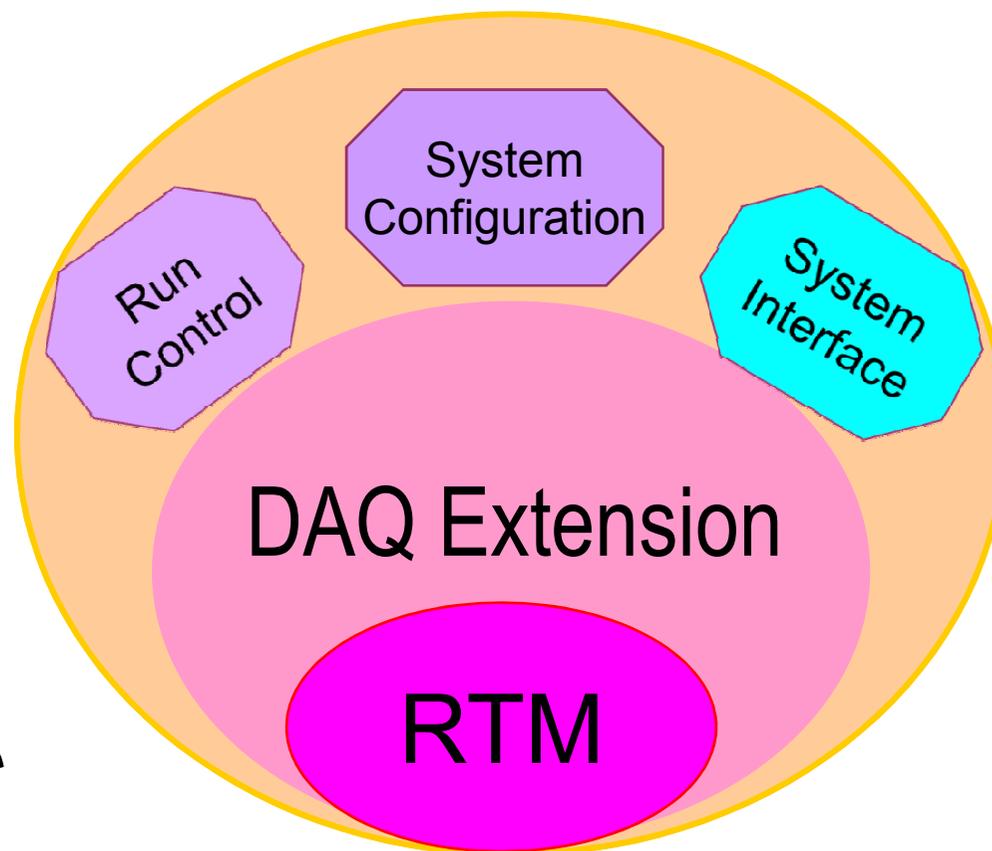
RT-Component Architecture



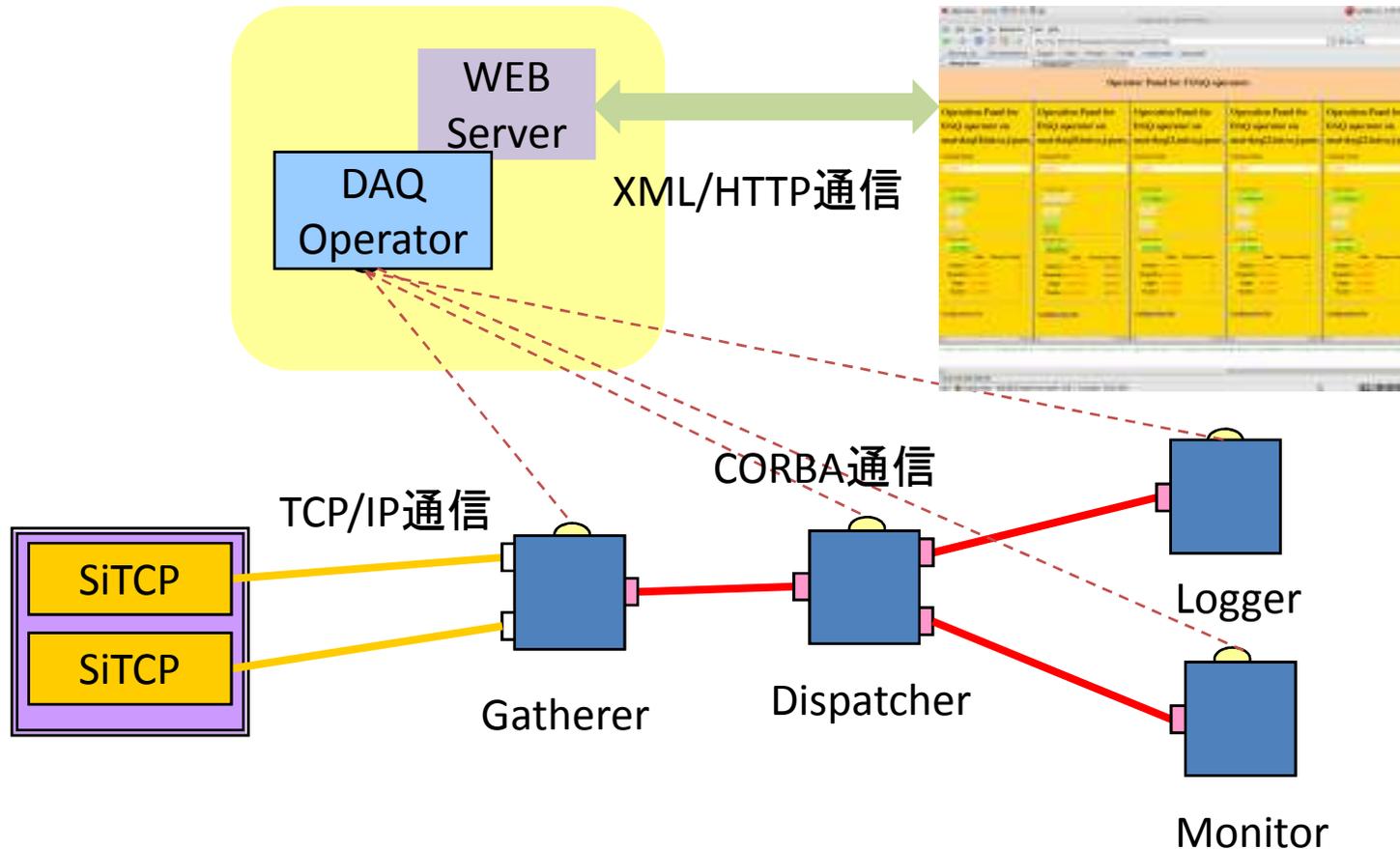
自律型モデル

DAQミドルウェアの特徴

- DAQ Extension
 - RTミドルウェアの上に独自のステートマシンを構築し、RTミドルウェアを基にした新たなDAQ用のベースクラスを設計
- その上に
 - コマンド・ステータスを操作するRun control
 - XMLを用いたSystem configuration
 - XML/HTTPプロトコルを用いたSystem Interface



DAQミドルウェア構成例

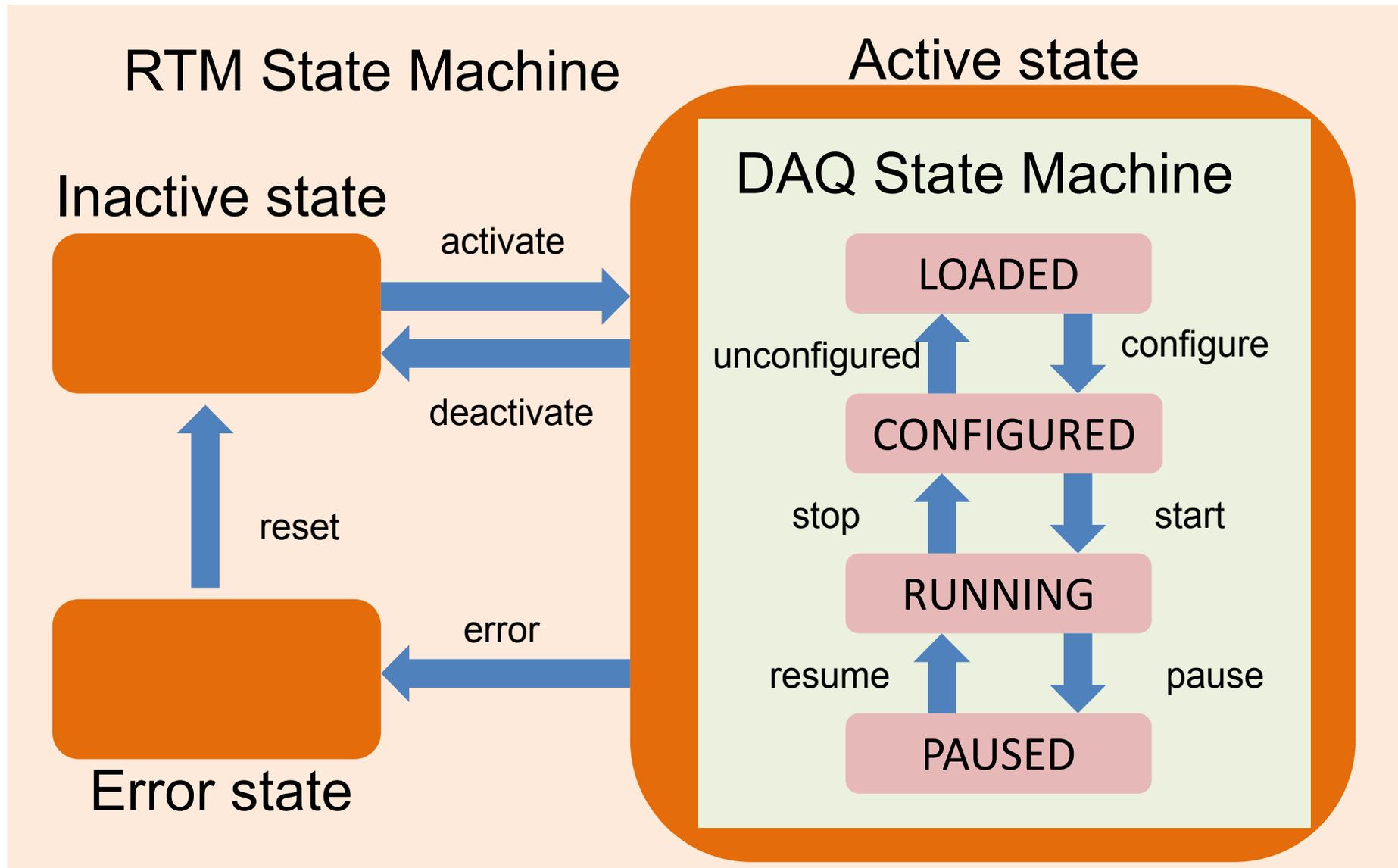


Gatherer、Dispatcher、Logger、Monitor、これらのコンポーネントはDAQOperatorの指示に従い、互いに他のコンポーネントと自律協調して動作している。

Run control

- コマンド・ステータスを操作する
- DAQ独自のステートマシンを持つ
- ステートの遷移はSystem Interfaceからのコマンドを受け取ることで実現する
- DAQOperatorは各種DAQコンポーネントに対応するコマンドを実行し、それらの操作する

State Machine



ユーザが記述すべき 主なメソッド

- `daq_configure()` — `configure`コマンドで構成パラメータを取得
- `daq_start()` — `start`コマンドでRUNの準備
- `daq_run()` — `daq_start()`が呼ばれた後、ひたすら実行される(このメソッドが何度も呼ばれる)
- `daq_pause()` — `pause`コマンドを受けて一時データ収集を中止
- `daq_resume()` — `resume`コマンドを受けて一時中止していたデータ収集を再開
- `daq_stop()` — `stop`コマンドを受けてRUNを終了する
- `daq_unconfigure()` — `unconfigure`コマンドを受けて再び`configure`コマンドを受け取れるようにする
- これらのメソッドを自分に合わせて変更するだけでよい。ただし、XMLのスキーマを変えるような変更の場合は、別途変更が必要。

System Configuration

- XML言語で記述される
- DAQOperatorはその記述をDocument Object Model(DOM)解析する
- DAQOperatorは解析されたパラメータを、DAQコンポーネントには、名前と値の対(NVList)にして渡す(コンポーネントの負担を軽減)
- これにより、汎用DAQコンポーネントを実現する

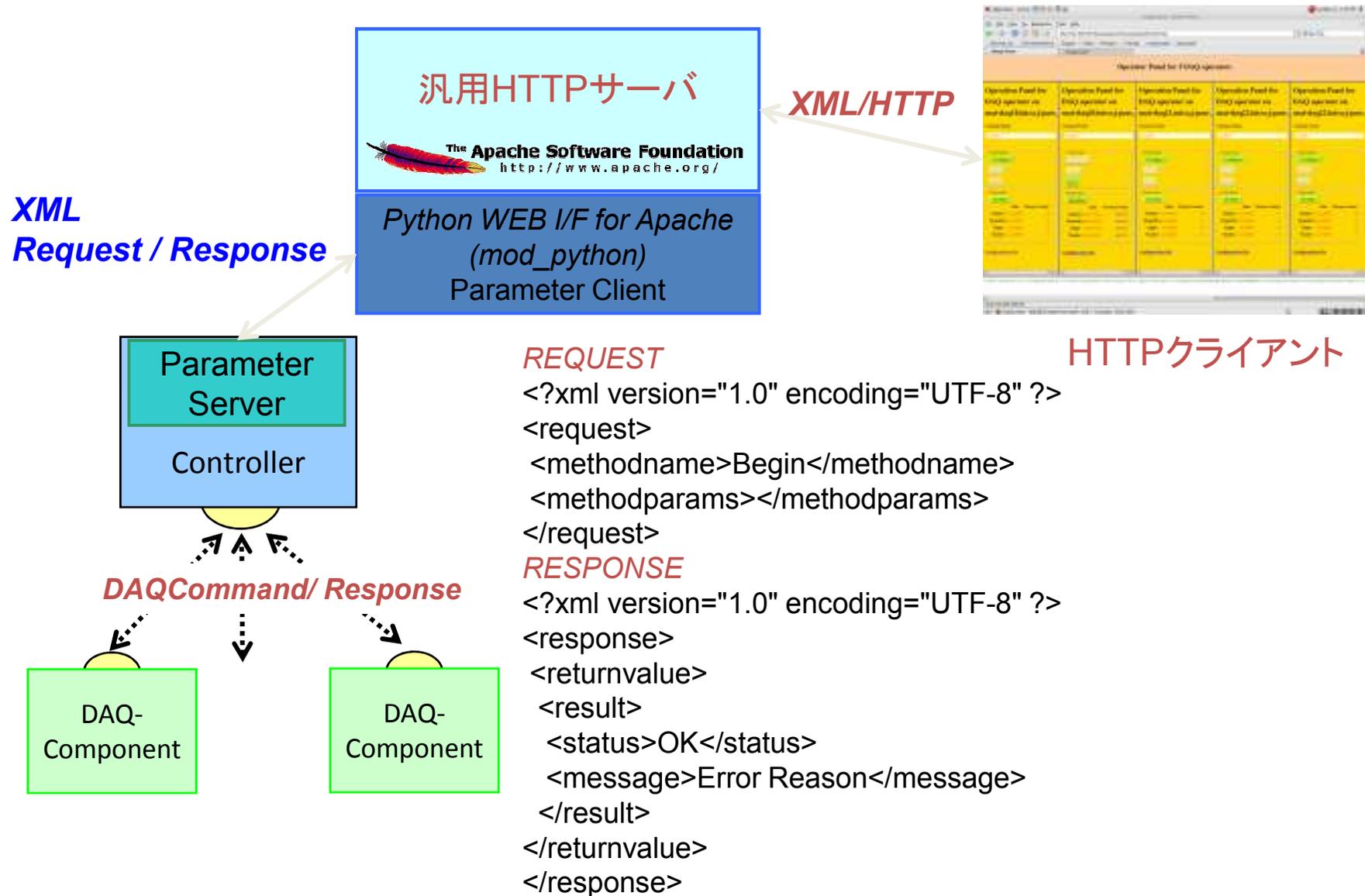
DAQミドルウェアの構成XMLファイル

```
<?xml version="1.0"?>
<configInfo>
  <daqGroups>
    <daqGroup gid="group1">
      <components>
        <component cid="Reader0">
          <hostAddr>kensdaqpc5.kek.jp</hostAddr>
          <hostPort>50000</hostPort>
          <instName>Reader0.rtc</instName>
          <execPath>/home/daq/bin/ReaderComp</execPath>
          <confFile>/home/daq/rtc.conf</confFile>
          <startOrd>4</startOrd>
          <inPorts>
          </inPorts>
          <outPorts>
            <outPort>reader_out</outPort>
          </outPorts>
          <params>
            <param pid="monRate">100</param>
            <param pid="portNo">1111</param>
            <param pid="srcAddr">127.0.0.1</param>
          </params>
        </component>
      </components>
    </daqGroup>
  </daqGroups>
</configInfo>
```

System interface

- メッセージはXML言語で、通信プロトコルはHTTPで実現される
- Apache httpd + mod_pythonモジュールを使用
- PythonスクリプトとDAQOperatorを結びつけるため、ParameterClient/ParameterServerクラスを開発(Python interface + C++クラス)

システムインターフェイスについて



ParameterClient programming

HTML programming

```
<form name="myform" action="/DAQ/daq.py/command"
      method="post">
  <input type="text" name="cmd" size=40 maxlength=40 />
  <input type="submit" value="Submit" />
</form>
```

mod_python
programming

Python interface for
ParameterClient

ParameterClient

```
def command(req, cmd):
    if not cmd:
        return "A required parameter is missing..."
    client = ParameterClient.ParameterClient("localhost",
                                             30000)

    client.put("command", cmd)
    return "OK"
```

ParameterServer programming

Pseudo DaqOperator +
ParameterServer

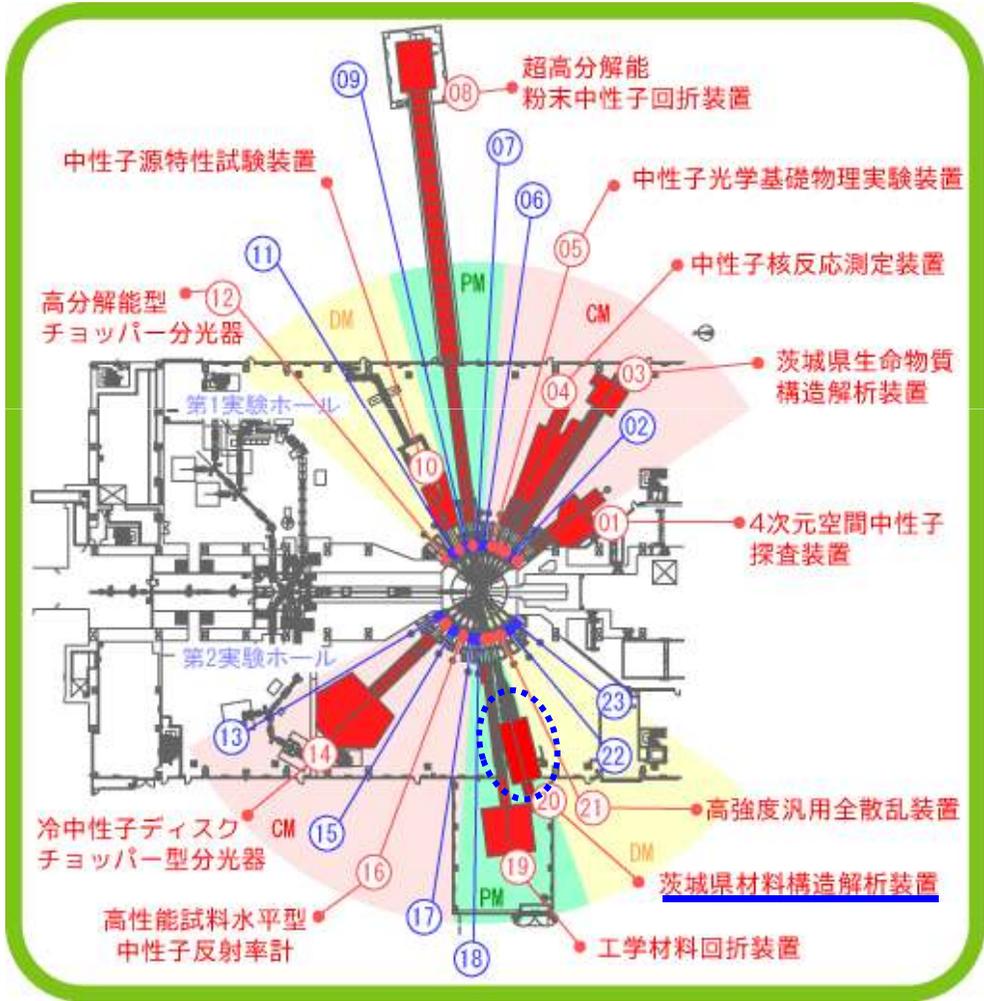
```
int start() {
    cout << "this is start method" << endl;
    cout << "gotValue = " << gotValue << endl;
}
RTC::ReturnCode_t DaqOperator::run2() {
    if(server==NULL) {
        server= new ParameterServer(30000);
        server.bind("command", &command, start);
        server.bind("status", &status);
        server.Run();
        command ="NOP";
    }
    // check
    if(command=="START") {
        start处理
    } else if (command=="...") {

    }
}
```

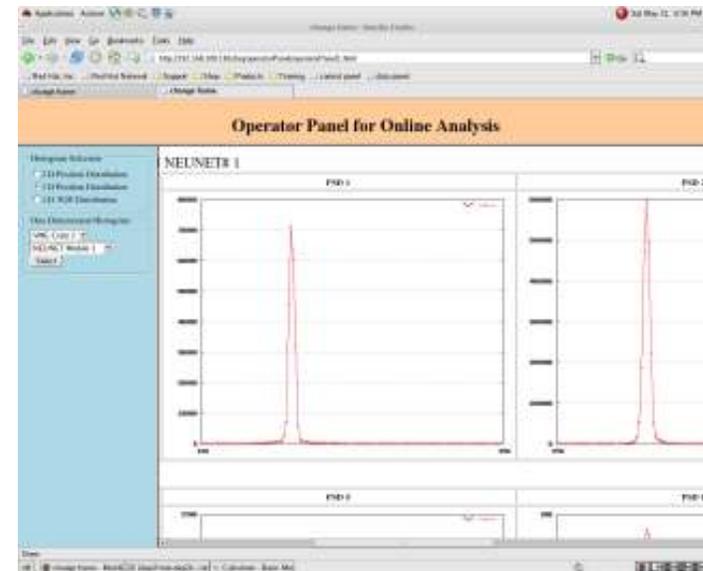
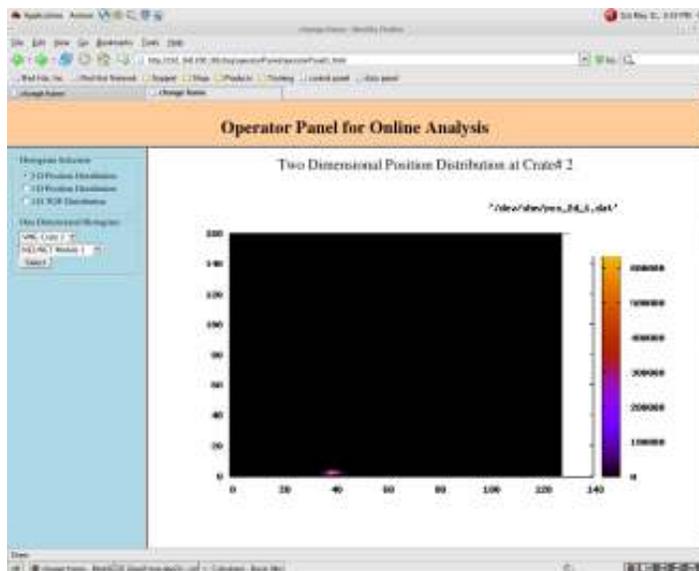
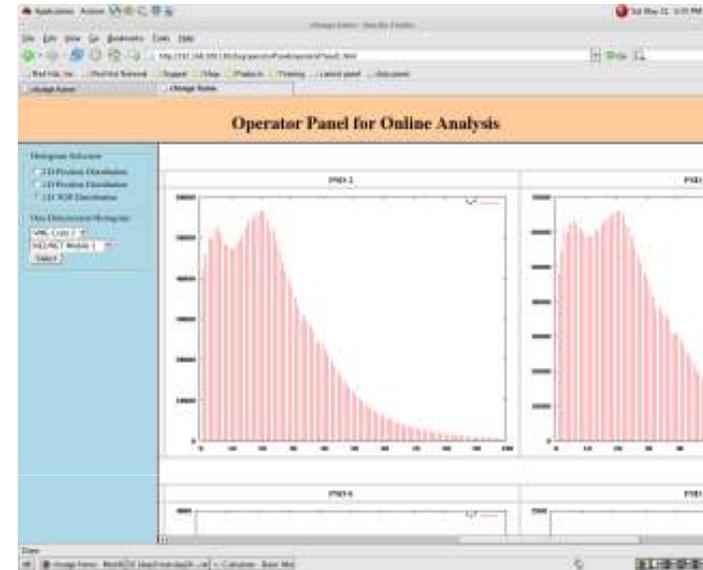
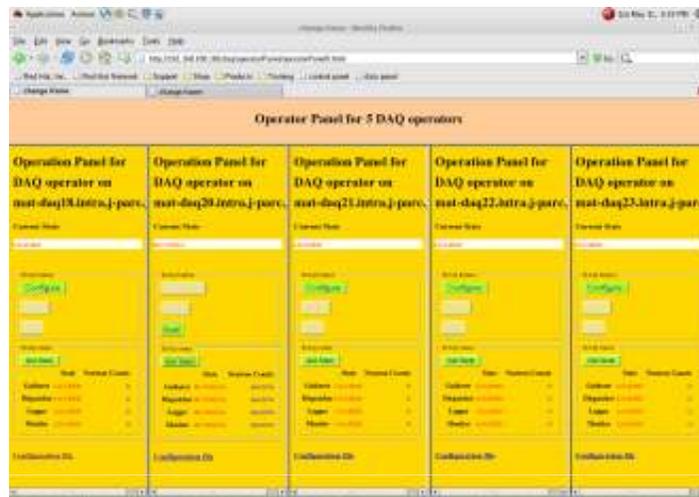
開発の現状

- RTミドルウェア
 - 産総研と共同研究を続けている
- DAQミドルウェア
 - 全国大学・研究所向けDAQ-Middleware-1.0.0-RCのリリース(DAQ講習会で利用しているもの)
 - DAQ-MiddlewareがJ-PARC物質・生命科学実験施設(MLF)の5月末の中性子ファーストビームで、初めて本格的な実験に利用された
 - 秋のMLF実験に向けてDAQ-Middlewareを改良中

J-PARC MLF



中性子ファーストビーム実験 スクリーンショット



まとめ

- DAQミドルウェアを紹介した
 - 国際標準ロボット技術、RTミドルウェアを拡張したもの
 - Run Controlを開発
 - XML言語を使ったSystem configurationを実現
 - Webサーバと連携してXML/HTTP通信で疎結合可能な仕組みを作った
- 全国大学・研究所向けDAQ-Middleware-1.0.0-RCをリリース
- DAQミドルウェアは本年5月末のJ-PARC中性子ファーストビーム実験で初めて本格的な実験に使用