

ASAM ODS Study Project Report

Regional Meeting Japan 2023

MEIDENSHA CORPORATION
ONO SOKKI Co.,Ltd
HONDA MotorCo.,Ltd

Yukimasa Maneki
Hiroyuki Sato
Masahiro Sato

28.Jun 2023
Japan



NISSAN MOTOR CORPORATION



Association for Standardization of
Automation and Measuring Systems

Agenda

1	Study Projectに至った背景
2	Member List
3	Group2活動報告
4	Group3活動報告

1.Study Projectに至った背景

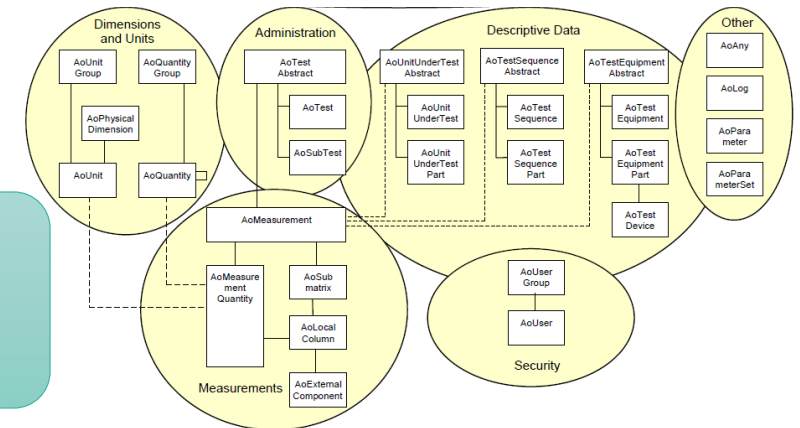
目的

- ODSはV6.1.1から6.2.0が標準化され、最新仕様について学び、活用に向けての理解が必要になってきた。
- ODSの活用を容易にし、利用者を増やすために、実ODSサーバを元に学習と検証を行う事が重要になってきた。
- ODSを上位システムと連携する場合のユースケースについて共通の課題としての検討が必要になってきた。

ゴール

- ODS最新リリースの内容を共有し理解する Group1活動
- ODSサーバーを立ち上げて簡単ODSライブラリを作成する Group2活動
- ODSと上位サーバーの連携を協議し共通の課題とユースケースをまとめる Group3活動

OEM,ODSサーバ,ツールベンダー各社の困りごとをStudy Projectを通して共有し、ODSを有効に活用する活動を行ってきた。



2.Member list

Company	Name	Company	Name	Company	Name
本田技研工業株式会社	佐藤 正浩	株式会社 東陽テクニカ	中山 仁志	Viviota	南 太門
本田技研工業株式会社	小笠原 清太郎	株式会社 東陽テクニカ	江川 正利	株式会社 小野測器	佐藤 広幸
本田技研工業株式会社	堤 純	株式会社 東陽テクニカ	黒田 正好	株式会社 小野測器	三瓶 祐一郎
本田技研工業株式会社	山本 一美	シーメンス株式会社	西田 徹	株式会社 小野測器	國府 裕毅
日野自動車株式会社	田上 佳弘	iASYS Technology Solutions株式会社	桑田 武	株式会社 A&D	山中 健太
日産自動車株式会社	榎谷 啓一	iASYS Technology Solutions株式会社	高橋 知己	AVLジャパン株式会社	田崎 裕児
日産自動車株式会社	大隣 弘幸	日本NI株式会社	米田 実紀	株式会社スカイテクノロジー	池田 練造
株式会社 SUBARU	信平 栄喜	株式会社堀場製作所	三十木 努	株式会社スカイテクノロジー	大野 努
株式会社 SUBARU	齋藤 慶大	株式会社堀場製作所	山本 光延	株式会社スカイテクノロジー	戸沼 渉
株式会社アイシン	森下 龍弘	株式会社 明電舎	招 行正		岡田 真澄
株式会社アイシン	安立 将英	株式会社 明電舎	清水 毅	ASAM Office	庄井 美章

Group 2 activity report

Activity to increase ASAM-ODS supported Measurement and Analysis tools

ODS対応の計測・解析ツールの開発を促進する取り組み

Activity to increase ASAM-ODS supported Measurement and Analysis tools

背景

日本のOEM、Tier1はODSサーバの導入を進めているが、ODSデータに対応した解析ツールがほとんど増えていない。そのため、ODSデータは限られたツールで利用するか、またはCSVファイルに変換してから利用しなければならず、業務の自動化・効率化が遅れている。

課題

以下2つの理由により計測・解析ツールベンダのODS対応が進んでいない。

- ODS仕様が複雑なため理解に時間が掛かる
- 実行環境が無く、動作検証を行うことができない

ゴール

Study Project 用としてサーバベンダに立ち上げていただいたODSサーバを利用し、ツールベンダがプログラムの試作と検証を行う。

成果として簡単ODSライブラリを作成し、公開する。

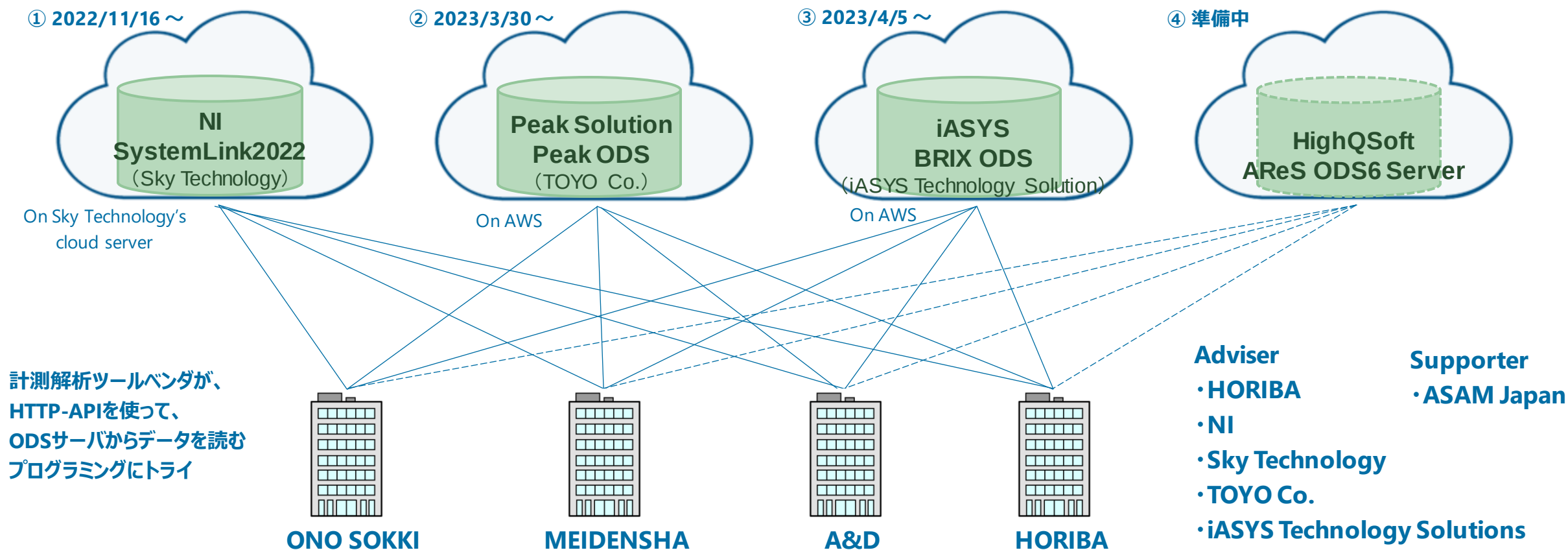
内容

1. 学習・検証用ODSサーバの構築（サーバベンダ）
2. HTTP-APIを使ったデータ取得の実習、及びプログラム試作（計測・解析ツールベンダ）
3. 簡単ODSライブラリの作成と公開（計測・解析ツールベンダ）

実施内容 1 学習用ODSサーバの構築

Content 1 Build up ODS server for learning

ASAM Japan ODS Study ProjectのODSサーバベンダが、学習用にODSサーバを構築・公開し、実行可能なサンプルコードを提供しました。

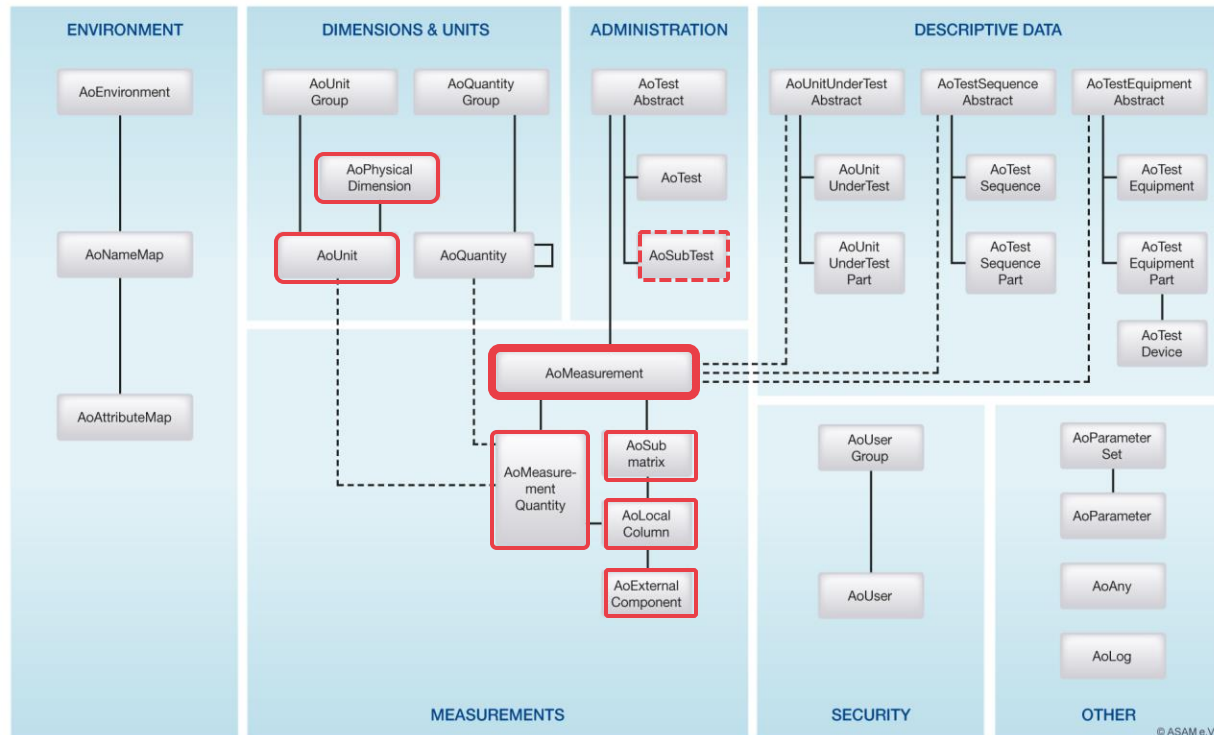


実施内容2 HTTP-APIによるODSサーバの使い方を実習

Content 2 Practicing programming using ODS server with HTTP-API

計測・解析ツールベンダは、ODSサーバベンダから提供してもらったサンプルコードとASAM ODS Base Standardを元に、HTTP-APIを使ってODSサーバから測定データを読み出すプログラミング方法を学習しました。

対象とする要素は、Measurement以下、Unitおよび上位のTest (or SubTest)



Sky Technology's sample code

```
1 import sys
2 import json
3 import pandas as pd
4 import requests
5
6 # Ignore certification verification
7 import ssl
8 ssl._create_default_https_context = ssl._create_unverified_context
9
10 # Disable InsecureRequestWarning
11 import urllib3
12 from urllib3.exceptions import InsecureRequestWarning
13 urllib3.disable_warnings(InsecureRequestWarning)
14
15 # proxy
16 import os
17 #os.environ['HTTP_PROXY'] = ''
18 #os.environ['HTTPS_PROXY'] = ''
19 os.environ['NO_PROXY'] = 'localhost'
20
21 # Scripts Parameters
22 #
23 host = "133.167.114.66"
24 url = "http://" + host + ":8080/ni/asam"
25 apikey = "-BFS8xzywsj-bd0uSLC4-DLFR6t"
26 showRes = False
27
28 # define function
```

```
1 import getpass
2 import base64
3 import json
4 import urllib.request
5 import urllib.error
6
7 baseURI = 'http://localhost:8080/api'
8 username = input('Enter username: ')
9 password = getpass.getpass('Enter password: ')
10
11 authorization = 'Basic ' + base64.b64encode(
12     username.encode() + b':' + password.encode())
13
14 HEADERS = {}
15 HEADERS['Authorization'] = authorization
16 HEADERS['Accept'] = 'application/x-asamods+'
17
18 # Create Connection Identifier
19 url = f'{baseURI}/ods'
20 req = urllib.request.Request(url=url, headers=HEADERS, method='POST')
21 res = urllib.request.urlopen(req)
22 loc = res.info().get('Location')
23 conl = loc[loc.rfind('/') + 1:]
24 print(f'Connection established! conl={conl}')
25
26 HEADERS['Content-type'] = 'application/x-asamods+'
27
28 # Retrieve Application Model
29 url = f'{baseURI}/ods/{conl}/model-read'
30 req = urllib.request.Request(url=url, headers=HEADERS, method='POST')
31 res = urllib.request.urlopen(req)
32 content = res.read().decode('UTF-8')
33 model = json.loads(content)
```

TOYO's sample code

```
1 import sys
2 import base64
3 import json
4 import pandas as pd
5 import requests
6 from requests.auth import HTTPBasicAuth
7 import urllib
8
9 # Ignore certification verification
10 import ssl
11 ssl._create_default_https_context = ssl._create_unverified_context
12
13 # proxy
14 import os
15 #os.environ['HTTP_PROXY'] = ''
16 #os.environ['HTTPS_PROXY'] = ''
17 os.environ['NO_PROXY'] = 'localhost'
18
19 # Scripts Parameters
20 #
21 host = "20.194.251.5"
22 url = "http://" + host + ":8090/brx/ods"
23 user = "ASAMuser"
24 password = "ASAM@2023"
25 showRes = False
26
27 # Open connection
28 #
29 print("\n[Open connection]")
30 try:
31     ...
32     headers = {'Content-type': 'application/x-asamods+json',
```

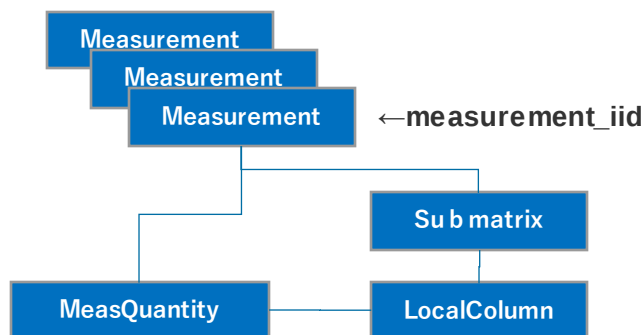
iASYS's sample code

実施内容3 簡単ODSライブラリを作成

Content 3 Create a simple ODS library

計測・解析ツールベンダは、2021年の活動で検討した簡単ODSライブラリの機能要求をPython用のライブラリとして作成することにトライしました。

Functional design at ASAM Japan ODS Study Group 2021



Name[0]	Name[1]	Name[2]	...	Name[n-1]
Unit[0]	Unit[1]	Unit[2]	...	Unit[n-1]
Type[0]	Type[1]	Type[2]	...	Type[n-1]
Data[0][0]	Data[1][0]	Data[2][0]	...	Data[n-1][0]
Data[0][1]	Data[1][1]	Data[2][1]	...	Data[n-1][1]
Data[0][2]	Data[1][2]	Data[2][2]	...	Data[n-1][2]
...	...	...	...	...
Data[0][m-1]	Data[1][m-1]	Data[2][m-1]	...	Data[n-1][m-1]

簡単ODSライブラリの関数（例）

- **Get_channel_list (measurement_iid)**
特定のMeasurementに属する全てのチャンネル情報（チャンネル名、単位名、ID...）を取得する
- **Get_data_by_name (base_element, iid, values_start, limit, namelist)**
チャンネル名リストに従って測定値を取得する

Welcome to SimpleOdsLibrary's documentation!

SimpleOds module

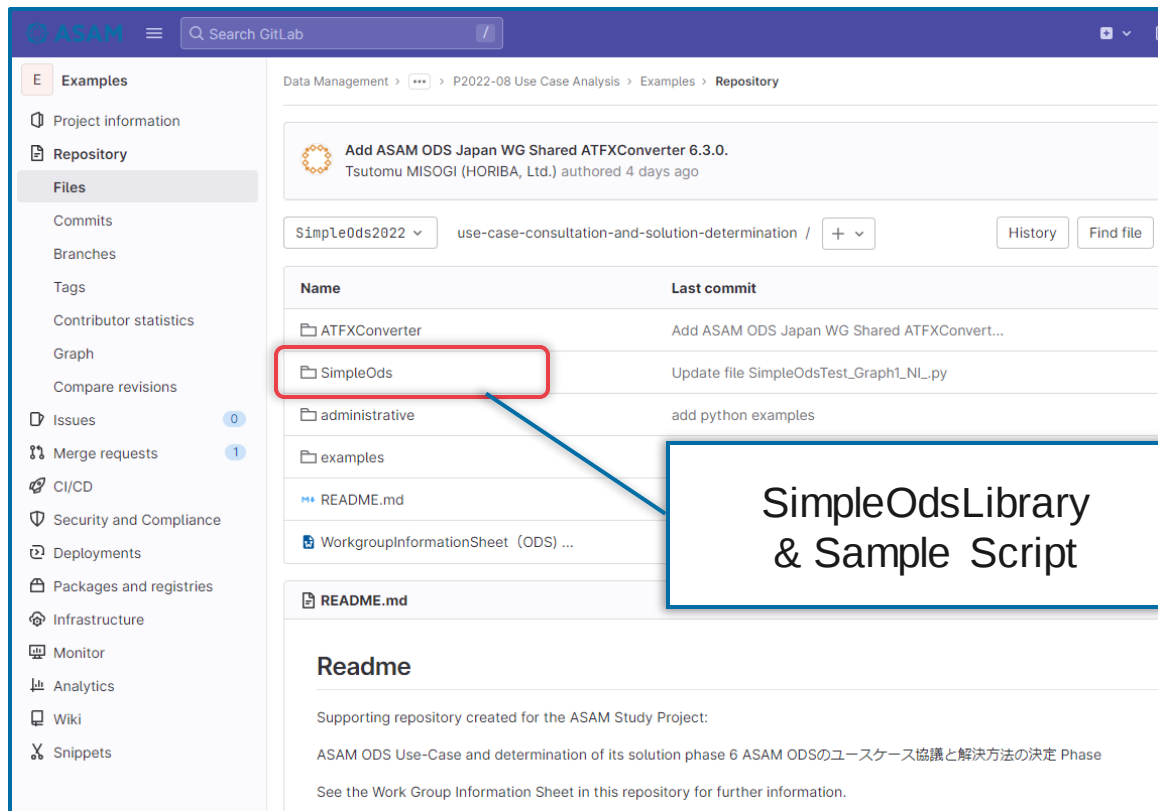
```
class SimpleOds.SimpleOds(url, api_key)
    """
    Simple library using HTTP-API of ASAM ODS 6.0
    Constructor (Initialize this class object)
    Set the ODS server URL and license key (API key)
    Parameters:
        url (string) -- Base URL for connecting to ODS
        api_key (string) -- Authentication key for ODS
    """
    def open():
        """
        Open the ODS server and establish a connection
        """
    def close():
        """
        Disconnect from the ODS server and delete connection ID
        """
    def read_application_model():
        """
        Read ApplicationModel
        Returns:
            Application model (JSON format)
            Returns: string
        """
    def get_aids(base_name):
        """
        Get application model ID list
        Parameters:
            base_name (string) -- Element name of basemodel
            Returns: Application ID list
            Returns: string[]
        """
    def get_attribute_names(oid, base_attribute_name):
        """
        Get the attribute names of the attribute with the base attribute name
        in the element with matching application ID
        Parameters:
            oid (longlong) -- application ID
            base_attribute_name (string) -- base attribute name
            Returns: attribute name list
            Returns: string[]
        """
    def get_relation_names(oid, base_relation_name):
        """
        Get the relation names of the relation with the base relation name
        in the element with matching application ID
        Parameters:
            oid (longlong) -- application ID
            base_relation_name (string) -- base relation name
            Returns: relation name list
            Returns: string[]
        """
    def read_units():
        """
        Read all Unit instances from the ODS server and set the entity in JSON format
        to a member variable.
        """
```

成果 1 簡単ODSライブラリを公開

Result 1 Released a simple ODS library

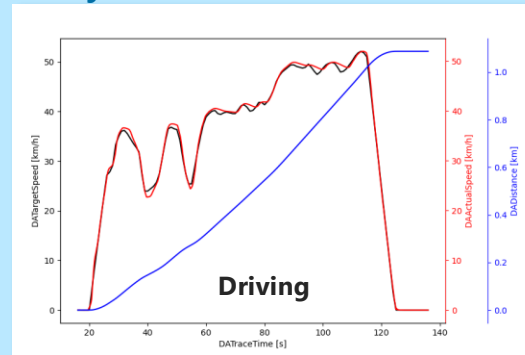
NI SystemLink2022、Peak ODS、iASYS BRIX ODS で共通に使用できるPython用のクラスライブラリ (SimpleODS.py) とサンプルスクリプトを作成し、GitLab上で公開しました。 ※HighQSoft用は7月の予定

ASAM's GitLab : P2022-08 Use Case Analysis/Examples/SimpleOds2022

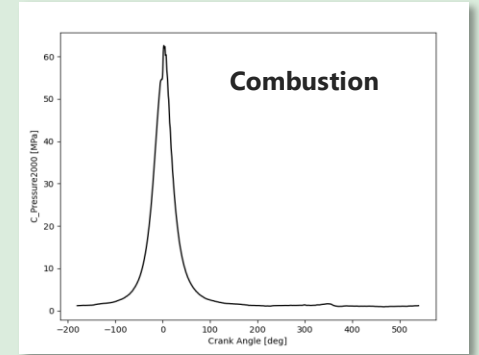


SimpleOdsLibrary
& Sample Script

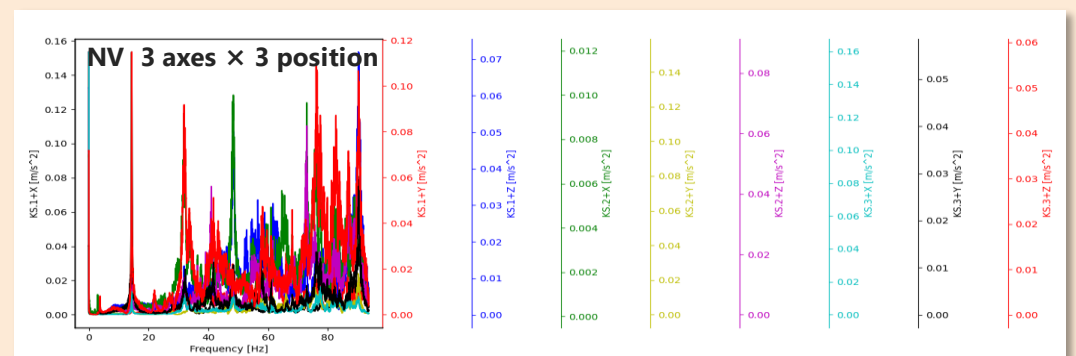
NI SystemLink 2022



iASYS BRIX ODS



Peak ODS



成果 2 計測・解析ツールベンダのスキルアップ

Result 2 Tool vendors got HTTP-API knowledges and improved programming skills

簡単ODSライブラリの作成過程でASAM ODS Standardを参照し、ODSデータを活用するプログラムを作成する方法を理解することができました。

学習の効果について代表的なものは以下の通りです。

HTTP-APIを利用したプログラミング方法が理解できた

3社のサーバからデータを読み出す共通ライブラリを作成でき、標準仕様(ASAM ODS Base Standard)の有効性を実感した

公開ODSサーバを使ったプログラミング実習を通し、Base ModelとApplication Model及びインスタンスの関係を理解した

Base Standardの記述が実際にどのようにインプリメントされているのか理解できた

3社の学習用ODSサーバと、提供いただいたサンプルコードが、何よりも役立つ教科書の役割を果たした

ODSサーバベンダやStudy Projectメンバのサポートにより不明点を解決し、更なる理解を深めることができた

学習を共にするメンバとの交流により、簡単ODSライブラリの作成に役立つ多くのヒントが得られた

今後の課題

Future tasks

今期の活動は計測・解析ツールベンダの学習と、後続く者たちの参入障壁を下げることを狙い、分かりやすさを優先した簡単ODSライブラリを作成しました。

再びODSサーバを利用できる機会をいただき、簡単ODSライブラリに検索などの機能追加とProtobuf使用による速度改善を行い、実際業務で利用できるようにしたいと考えています。

Protobufによる速度性能の向上

対象エレメントを広げ、検索などの実用機能を充実させる

新仕様（ASAM ODS 6.2 EXD-APIによる外部データへのアクセス）への対応

Python以外の開発言語（C#等）に応じたライブラリ作成

様々な用途に合わせたサンプルコードの拡充

謝辞

Thanks

本活動を実施するには多くの方々のご協力が必要でした。

Sky Technology／戸沼さん、NI／米田さん、TOYO Co／江川さん、黒田さん、iASYS Technology Solutions／高橋さんには、貴重なODSサーバ製品の構築・公開とサンプルコードの解説など多大なご尽力に感謝いたします。

活動期間全般でご支援いただいたASAM Office／庄井さん、その他 ODS Study Projectに参加し、活動を応援してくださった皆様ありがとうございました。

