

センサー情報相互運用配信システム システム設計書

Ver. 1.0.0

2015/9/1

国立研究開発法人 防災科学技術研究所

変更履歴

Version	変更日付	変更内容
1.0	2015/09/1	初版作成

目次

1	概要	1
2	システム構成	1
2.1	システム構成図	1
2.2	システム構成	2
3	機能設計	3
3.1	SOS 格納システム	3
3.2	SOS 配信システム	5
4	画面仕様	6
4.1	GetCapabilities リクエストに対するレスポンス画面	6
4.2	DescribeSensor リクエストに対するレスポンス画面	7
4.3	GetObservation リクエストに対するレスポンス画面	8
5	プロパティファイル詳細	10
5.1	格納配信システム設定ファイル	10
5.2	各センサー設定ファイル	10
6	データベース仕様	13
6.1	データベース ER 図	13
6.2	各テーブルスキーマ	16

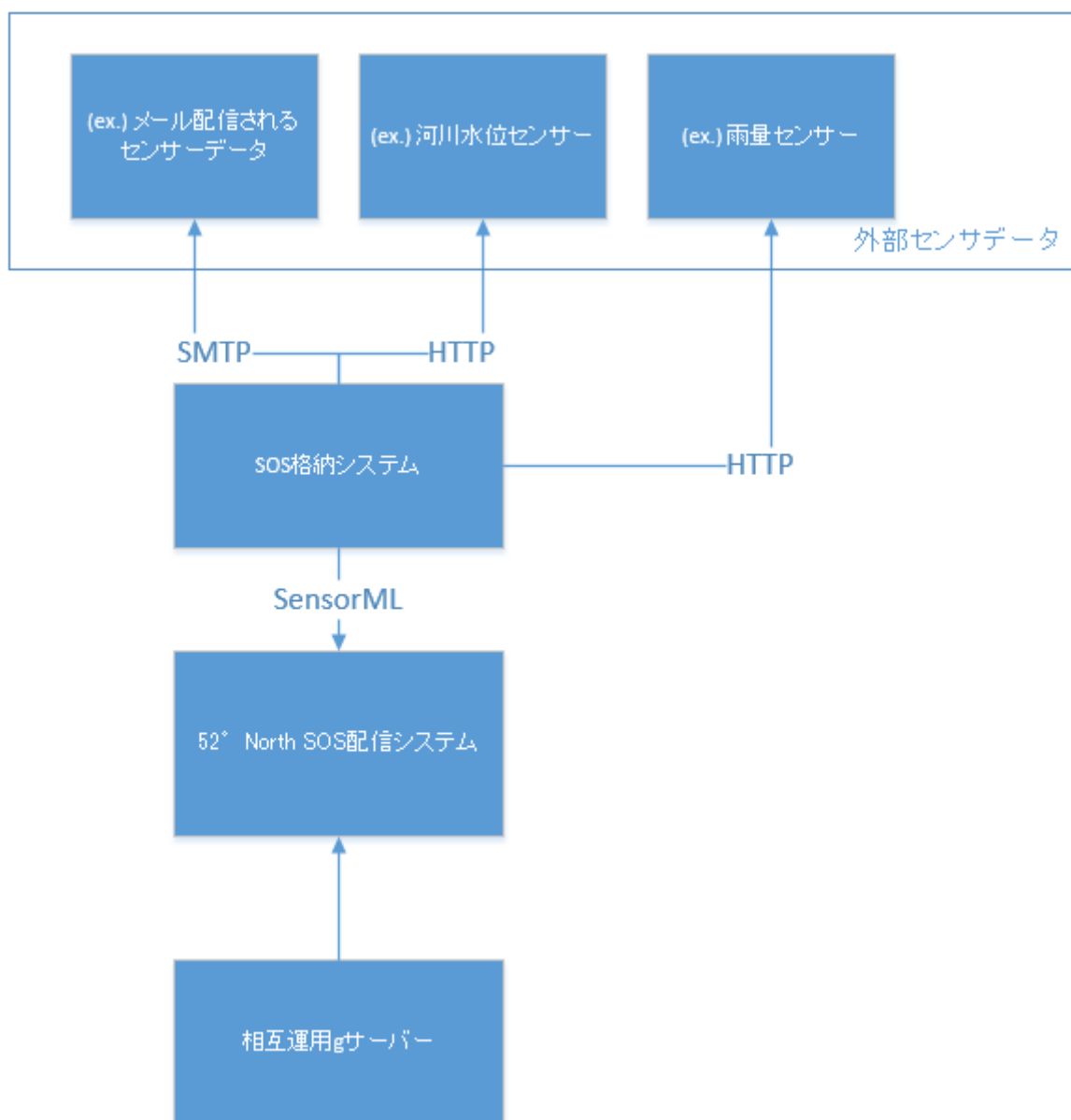
1 概要

本基本設計書は、センサー情報相互運用配信システムをの基本仕様を示すものである。

2 システム構成

2.1 システム構成図

SOS 格納システムは外部センサデータサーバから観測値を取得し 52°North SOS 配信システムに登録できる OGC の SensorML を作成できるシステムである。



2.2 システム構成

本システムは、SOS 格納システムと SOS 配信システムから構成されるシステムである。SOS 格納システムは、Web 上に公開された外部センサーデータ（CSV 配信を想定）を取得あるいはメール配信されたデータを取得してデータベースに格納する処理を行う。また、SOS 配信システムは、データベースに格納したデータを OGC(Open Geospatial Consortium)の標準に準拠した仕様で外部に配信する。

SOS 配信システムは、52°North Sensor Observation Service を利用しており、以下のリクエストに対して OGC SOS 標準に従ったレスポンスを行う。

- GetCapabilities
- DescribeSensor
- GetObservation

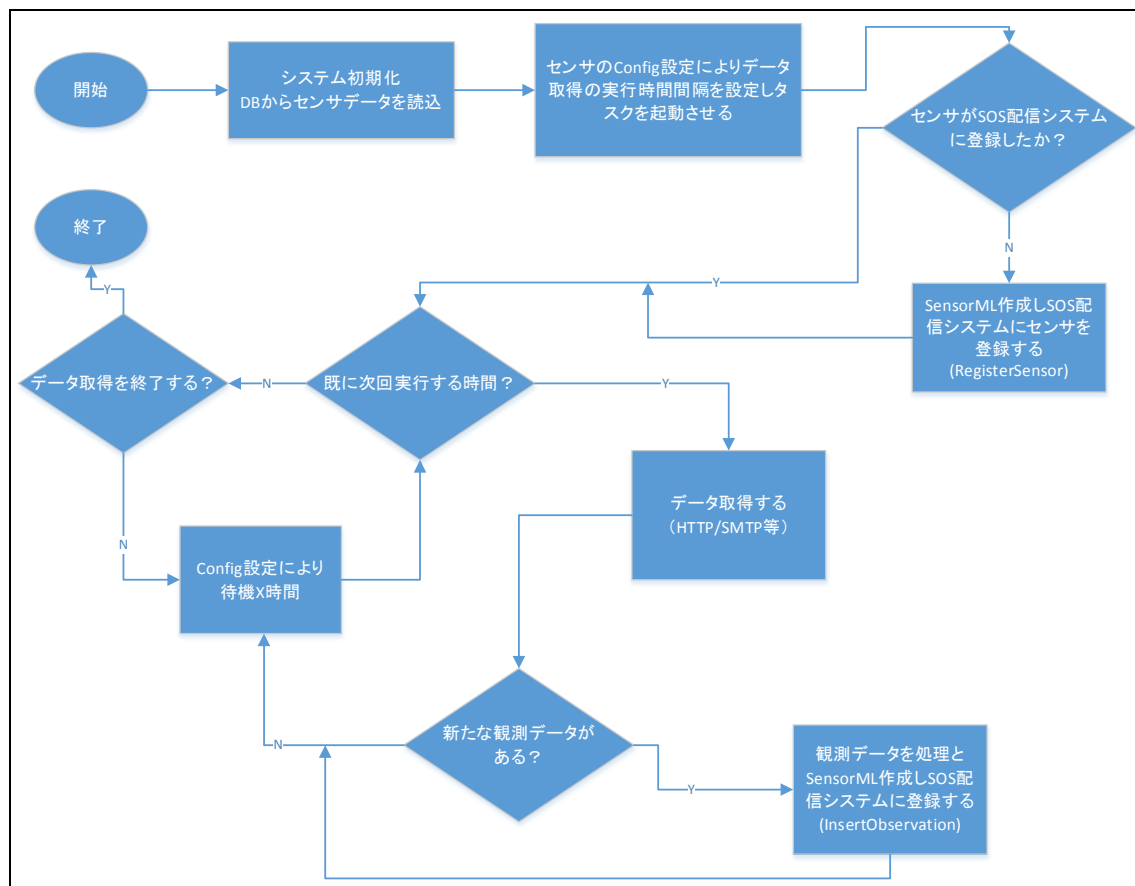
API については、センサーAPI(Sensor Observation Service)仕様書を参照のこと。

3 機能設計

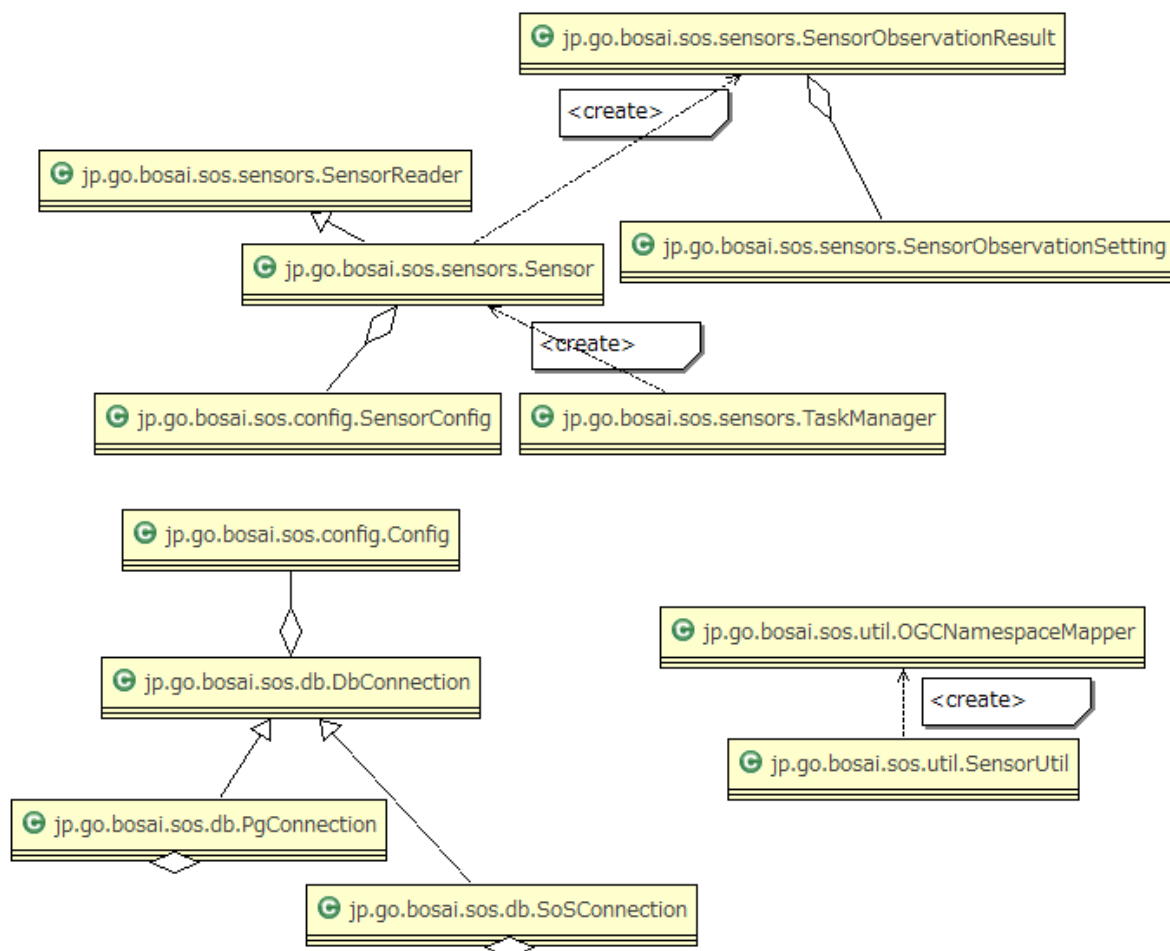
3.1 SOS 格納システム

1) フローチャート

SOS 格納システム起動すると格納システムのデータベースからセンサ設定を読み込んで SOS 配信システムに登録する。センサ設定が SOS 配信システムに登録されていなければ、自動的に OGC に準拠した SensorML を作成し SOS 配信システムに登録する。



2) UML クラス図



3) パッケージとクラス構成

パッケージ名	クラス名	説明
jp.gobosai.sos		
	Main	メインクラス
jp.go.bosai.sos.config		設定に関するパッケージである
	Config	システム全般的な Config 設定クラスである。データベースの接続文字列、ホストとユーザなどの設定を設定ファイルから読み込んでプログラム全体的にアクセスできる機能である
	SensorConfig	センサの設定ファイルから読込クラスである
jp.go.bosai.sos.db		データベースにアクセスとクエリする等のパッケージである
	DbConnection	データベース共通処理クラスである

	PgConnection	SOS 格納システムのデータベース接続クラス
	SosConnection	SOS 配信システムへのデータベース接続クラスである
jp.go.bosai.sos.sensors		センサデータ処理に関するパッケージである
	SensorObservationResult	所得した観測値を保管するクラスである
	SensorObservationSetting	観測データの設定を保管するクラスである
	SensorReader	センサデータ取得と処理の共通処理クラスである
	Sensor	センサデータ取得と取得したデータの処理をするクラスである
	TaskManager	タスクの起動と終了を管理するクラスである
jp.go.bosai.sos.util		ユーティリティ パッケージである
	DateUtil	日付ユーティリティクラスである
	OGCNamespaceMapper	SensorUtil で SensorML 作成する際に使われる名前空間を管理するユーティリティクラスである
	SensorUtil	JAXB でインスタントから SensorML 形式に XML 化をするユーティリティクラスである

3.2 SOS 配信システム

SOS 配信システムは、52°North SOS 配信システムを利用している。

詳細は、52°North Sensor Observation

Service <http://52north.org/communities/sensorweb/sos/index.html> を参照のこと。

4 画面仕様

SOS 格納システムについてのユーザ・インターフェース画面はない。

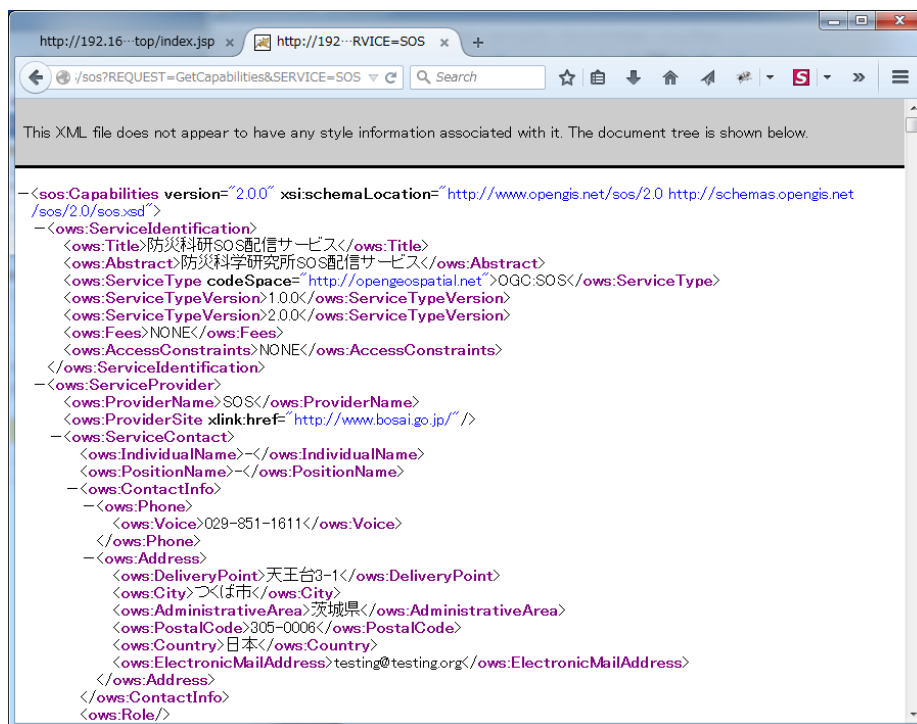
SOS 配信システムについてのユーザ・インターフェース画面は、52°North Sensor Observation Service を利用している。



4.1 GetCapabilities リクエストに対するレスポンス画面

OGC の仕様に則った GetCapabilities リクエストで下図のようなレスポンス画面が得られる。

http://192.168.100.169:8080/NIED_SOS/sos?REQUEST=GetCapabilities&SERVICE=SOS

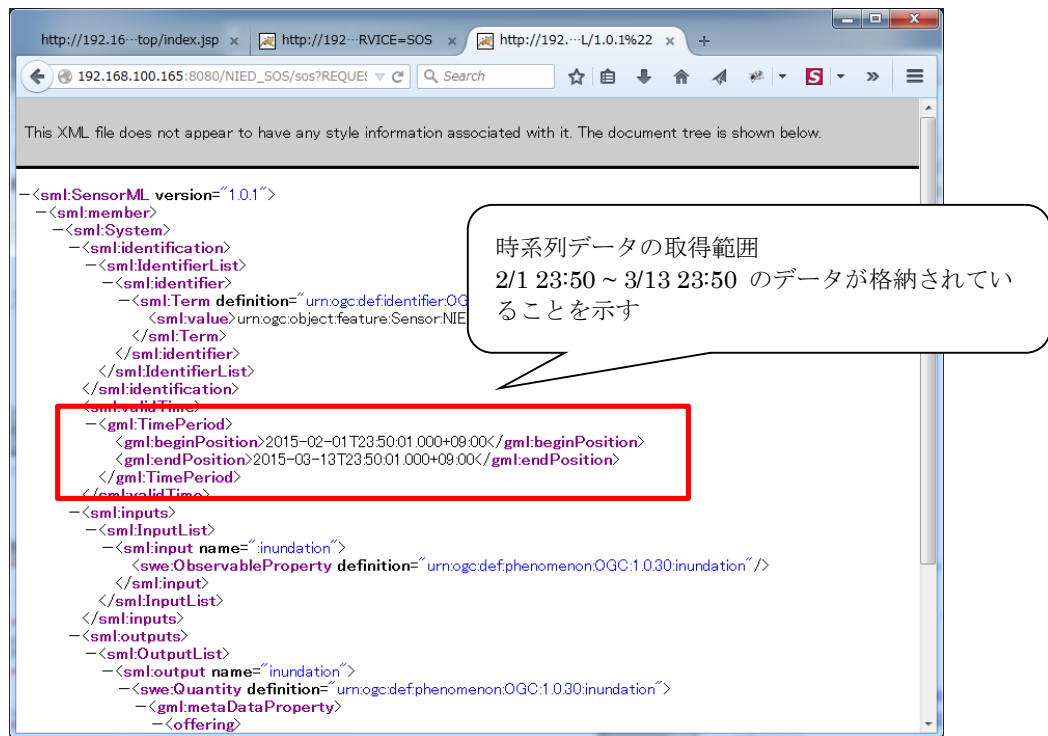


DescribeSensor リクエスト例 (センサ ID procedure_id

urn:ogc:object:feature:Sensor:NIED:flood-10)

OGC の仕様に則った DescribeSensor リクエストで下図のようなレスポンス画面が得られる。

[http://192.168.100.169:8080/NIED_SOS/sos?REQUEST=DescribeSensor&SERVICE=SOS&VERSION=1.0.0&PROCEDURE=urn:ogc:object:feature:Sensor:NIED:flood-10&OUTPUTFORMAT=text/xml:subty
pe="sensorML/1.0.1](http://192.168.100.169:8080/NIED_SOS/sos?REQUEST=DescribeSensor&SERVICE=SOS&VERSION=1.0.0&PROCEDURE=urn:ogc:object:feature:Sensor:NIED:flood-10&OUTPUTFORMAT=text/xml:subtype=sensorML/1.0.1)



4.3 GetObservation リクエストに対するレスポンス画面

GetObservation リクエスト例 (センサ ID

OBSERVEDPROPERTY=urn:ogc:def:phenomenon:OGC:1.0.30:inundation)

OGC の仕様に則った GetObservation リクエストで下図のようなレスポンス画面が得られる。

[http://192.168.100.169:8080/NIED_SOS/sos?REQUEST=GetObservation&SERVICE=SOS&VERSION=1.0.0&OFFERING=INUNDATION&OBSERVEDPROPERTY=urn:ogc:def:phenomenon:OGC:1.0.30:inundation&RESPONSEFORMAT=text/xml;subtype="om/1.0.0"](http://192.168.100.169:8080/NIED_SOS/sos?REQUEST=GetObservation&SERVICE=SOS&VERSION=1.0.0&OFFERING=INUNDATION&OBSERVEDPROPERTY=urn:ogc:def:phenomenon:OGC:1.0.30:inundation&RESPONSEFORMAT=text/xml;subtype=)

http://192.../index.jsp x http://1...ICE=SOS x http://1...0.1%22 x http://1...0.0%22 x

Search

センサー群の緯度、経度範囲

```

<om:ObservationCollection gml:id="oc_1426303481945" xsi:schemaLocation="http://schemas.opengis.net/om/1.0.0/extensions/observationSpecial http://schemas.opengis.net/sampling/1.0.0/sampling.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
      <gml:lowerCorner>35.30978056 139.4322361</gml:lowerCorner>
      <gml:upperCorner>36.124777 140.090877</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <om:member>
    <om:Observation gml:id="o_1426303481960">
      <om:samplingTime>
        <gml:TimePeriod xsi:type="gml:TimePeriodType">
          <gml:beginPosition>2015-02-01T23:50:01.000+09:00</gml:beginPosition>
          <gml:endPosition>2015-03-13T23:50:01.000+09:00</gml:endPosition>
        </gml:TimePeriod>
      </om:samplingTime>
      <om:procedure xlink:href="urn:ogc:object:feature:Sensor:NIED:flood-3"/>
      <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:inundation"/>
      <om:featureOfInterest xlink:href="http://localhost:8080/NIED_SOS/sos?REQUEST=GetFeatureOfInterest&service=SOS&version=2.0.0&featureOfInterest=foi_flood_sensor_3-INUNDATION" xlink:title="foi_flood_sensor_3-INUNDATION"/>
      <om:result>
        <swe:DataArray>
          <swe:elementCount>
            <swe:Count>
              <swe:value>63</swe:value>
            </swe:Count>
          </swe:elementCount>
          <swe:elementType name="Components">
            <swe:DataRecord>
              <swe:field name="SamplingTime">
                <swe:Time definition="http://www.opengis.net/def/property/OGC/0/SamplingTime">
                  <swe:uom xlink:href="http://www.opengis.net/def/uom/ISO-8601/0/Gregorian"/>
                </swe:Time>
              </swe:field>
              <swe:field name="inundation">
                <swe:Quantity definition="urn:ogc:def:phenomenon:OGC:1.0.30:inundation">
                  <swe:uom code="cm"/>
                </swe:Quantity>
              </swe:field>
            </swe:DataRecord>
          </swe:elementType>
          <swe:encoding>
            <swe:TextBlock decimalSeparator="." tokenSeparator="," blockSeparator=";" />
          </swe:encoding>
          <swe:values>
            2015-02-01T23:50:01.000+09:00,0.0137,2015-02-02T23:50:01.000+09:00,0.0137,2015-02-03T23:50:01.000+09:00,0.0
          </swe:values>
        </swe:DataArray>
      </om:result>
    </om:Observation>
  </om:member>
</om:ObservationCollection>

```

センサーデータの測定範囲

1つのセンサーデータの内容
センサー群の緯度、経度範囲

実際のデータ時刻、値

5 プロパティファイル詳細

5.1 格納配信システム設定ファイル

DB への接続情報などを記載する。

パッケージ NIED_SOS¥SOSDataService¥conf¥config.properties

ファイル名	config.properties
ファイル説明	SOS 格納機能全般的な設定ファイル
場所	conf/

項目	説明	値
PG_DRIVER	Postgres の接続のドライバー	org.postgresql.Driver
PG_CONNECTIONSTRING	Postgres の接続文字列	jdbc:postgresql://[ホスト]:5432/[格納機能 DB 名]
PG_USER	Postgres のユーザー名	postgres
PG_PASS	Postgres のパスワード	
PG_MAXCONNECTION	Postgres への最大同時接続数	50
SOS_DRIVER	Postgres の接続のドライバー	org.postgresql.Driver
SOS_CONNECTIONSTRING	Postgres の接続文字列	jdbc:postgresql://ホスト:5432/[52North SOS 配信サービス DB 名]
SOS_USER	Postgres のユーザー名	postgres
SOS_PASS	Postgres のパスワード	
SOS_MAXCONNECTION	Postgres への最大同時接続数	50
SRID	空間参照システム識別子	4326
SOS_SERVICE_ENDPOINT	52North SOS 配信サービスの URL	—

5.2 各センサー設定ファイル

各センサーの設定ファイルでは、データ取得に必要な情報（メール形式であればメールサーバの接続情報等、ウェブアクセスであれば URL パターンなど）や取得したデータの解析に必要な情報を設定する。ここでは、具体例としてふじのくにオープンデータカタログ¹にて公開されている雨量情報（以下雨量センサ）水位情報（以下水位センサー）について記述する。

(1) 雨量センサの設定ファイル

パッケージ NIED_SOS¥SOSDataService¥conf¥shizuoka_rainsensor.properties

¹ <http://open-data.pref.shizuoka.jp/mu0b05bub-139/>

ファイル名	shizuoka_rainsensor.properties
ファイル説明	雨量センサの設定ファイル
場所	conf/
連動するテーブル名	shizuoka_rain_sensor

項目	説明	値
DATA_CRAWL_INTERVAL	データ得る間隔（秒）	60
DATA_CRAWL_TIME_OFFSET	URL で使われる日付との差（秒）	-300
URL_PATTERN	URL パターン \${formatted_datetime} は日付パターン自動作成箇所	http://sipos.shizuoka2.jp/rain-river/Rain/Shizuoka_Rain_\${formatted_datetime}.csv
FORMATTED_DATETIME_EXPR	URL 日付パターン	yyyyMMddHHmm
FORMATTED_DATETIME_STEP	URL 日付のステップ（分）	5
CSV_DATA_ROW_INDEX	CSV 無視する行数	1
CSV_DATETIME_FORMAT	CSV 日付フォーマット	yyyy-MM-dd HH:mm:ssZ
CSV_DATETIME_TIMEZONE	CSV 日付時間帯	Asia/Tokyo
PROCEDURE_ID_PATTERN	Procedure Id のパターン \${rain_sensor_id} はデータベースのテーブル Id により自動作成する	urn:ogc:object:feature:Sensor:NIED:rain-\${rain_sensor_id}
FOI_ID_PATTERN	Feature of Interest Id のパターン \${rain_sensor_id} はデータベースのテーブル Id により自動作成する	foi_rain_sensor_\${rain_sensor_id}

(2) 水位センサの設定ファイル

パッケージ NIED_SOS¥SOSDataService¥conf¥shizuoka_waterlevelsensor.properties

ファイル名	shizuoka_waterlevelsensor.properties
ファイル説明	水位センサの設定ファイル
場所	conf/
連動するテーブル名	shizuoka_waterlevel_sensor

項目	説明	値
DATA_CRAWL_INTERVAL	データ得る間隔 (秒)	60
DATA_CRAWL_TIME_OFFSET	URL で使われる日付との差 (秒)	-300
URL_PATTERN	URL パターン \${formatted_datetime} は日付パターン自動作成箇所	http://sipos.shizuoka2.jp/rain-river/Water_Level/Shizuoka_WaterLevel_\${formatted_datetime}.csv
FORMATTED_DATETIME_EXPR	URL 日付パターン	yyyyMMddHHmm
FORMATTED_DATETIME_STEP	URL 日付のステップ (分)	10
CSV_DATA_ROW_INDEX	CSV 無視する行数	1
CSV_DATETIME_FORMAT	CSV 日付フォーマット	yyyy-MM-dd HH:mm:ssZ
CSV_DATETIME_TIMEZONE	CSV 日付時間帯	Asia/Tokyo
PROCEDURE_ID_PATTERN	Procedure Id のパターン \${waterlevel_sensor_id} はデータベースのテーブル Id により自動作成する	urn:ogc:object:feature:Sensor:NIED:waterlevel-\${waterlevel_sensor_id}
FOI_ID_PATTERN	Feature of Interest Id のパターン \${waterlevel_sensor_id} はデータベースのテーブル Id により自動作成する	foi_waterlevel_sensor_\${waterlevel_sensor_id}

6 データベース仕様

6.1 データベース ER 図

(1) SOS 格納システム

観測データ設定テーブル(sensor_observation_setting)と他のセンサテーブルと直接に外部キーで結ばれていないが、実際には sensor_observation_setting.table_name=<センサテーブル名>と sensor_observation_setting.ref_sensor_id=<センサテーブルのセンサ Id>の条件により動的に結ばれている。

ここでは、具体例としてふじのくにオープンデータカタログ²にて公開されている雨量情報、水位情報のデータベース ER 図について記述する。

sensor_observation_setting	
id	SERIAL
table_name	VARCHAR(2147483647)
ref_sensor_id	INTEGER
csv_column_index	INTEGER
calculate_expr	VARCHAR(2147483647)
phenomenon_id	VARCHAR(2147483647)
offering_id	VARCHAR(2147483647)
uom	VARCHAR(2147483647)

shizuoka_rain_sensor	
rain_sensor_id	INTEGER
name	VARCHAR(50)
office_id	INTEGER
address	VARCHAR(250)
latitude	INTEGER
longitude	INTEGER
feature_of_interest_id	VARCHAR(100)
last_timestamp	TIMESTAMP
procedure_id	VARCHAR(100)
date_column_index	INTEGER

shizuoka_waterlevel_sensor	
waterlevel_sensor_id	INTEGER
name	VARCHAR(50)
office_id	INTEGER
latitude	INTEGER
longitude	INTEGER
drainage_name	VARCHAR(50)
river_name	VARCHAR(50)
address	VARCHAR(250)
rank_1	INTEGER
rank_2	INTEGER
rank_3	INTEGER
rank_4	INTEGER
feature_of_interest_id	VARCHAR(100)
last_timestamp	TIMESTAMP
procedure_id	VARCHAR(100)
date_column_index	INTEGER

² <http://open-data.pref.shizuoka.jp/mu0b05bub-139/>

(2) SOS 配信システム

以下に SOS 配信システムの ER 図³を示す。

観測データは、**observation** テーブルの **numeric_value** カラムに格納され以下のカラムにより一意に決定される。

- **time_stamp** カラム (観測時刻を格納)
- **procedure_id** カラム (センサー)
- **feature_of_interest_id** カラム (センサーが配置されている地物)
- **phenomenon_id** カラム (観測対象、雨量など)

³ The 52North SOS standard data model
<https://wiki.52north.org/bin/view/SensorWeb/SosDataModeling>

6.2 各テーブルスキーマ

具体例としてふじのくにオープンデータカタログ にて公開されている雨量情報(以下雨量センサ) 水位情報(以下水位センサー) を本システムで設定する場合のテーブルスキーマを記述する。

(1) shizuoka_rain_sensor⁴

テーブル名	shizuoka_rain_sensor
説明	雨量センサ設定テーブル

フィールド名	データ型	説明
rain_sensor_id[PK]	int	雨量センサ ID
name	varchar(50)	センサ名
office_id	int	雨量センサ事務所 Id
address	varchar(250)	住所 (参考のために保留)
latitude	double	センサの経度
longitude	double	センサの緯度
feature_of_interest_id	varchar(100)	SOS 配信システムから戻した FOI の ID (RegisterSensor を作成する際、設定ファイルに指定した FOI_ID_PATTERN を基づき生成する)
last_timestamp	timestamp	最後のデータ取得した時間
procedure_id	varchar(100)	SOS 配信システムから戻した Procedure の ID (RegisterSensor を作成する際、設定ファイルに指定した PROCEDURE_ID_PATTERN を基づき生成する)
date_column_index	int	データを取込際に日付データのコラム位置

(2) shizuoka_waterlevel_sensor⁵

テーブル名	shizuoka_waterlevel_sensor
説明	水位センサ設定テーブル

フィールド名	データ型	説明
waterlevel_sensor_id[PK]	int	水位センサ ID

⁴雨量センサデータの参照リンク

<http://sipos.shizuoka2.jp/rain-river/siposDataDownload.pdf>

⁵水位センサデータの照リンク

<http://sipos.shizuoka2.jp/rain-river/siposDataDownload.pdf>

name	varchar(50)	センサ名
office_id	int	水位センサ事務所 Id
latitude	double	センサの経度
longitude	double	センサの緯度
drainage_name	varchar(50)	水系（参考のために保留）
river_name	varchar(50)	河川系（参考のために保留）
address	varchar(250)	住所（参考のために保留）
rank_1	double	水防団待機水位（参考のために保留）
rank_2	double	氾濫注意水位（参考のために保留）
rank_3	double	避難判断水位（参考のために保留）
rank_4	double	氾濫危険水位（参考のために保留）
feature_of_interest_id	varchar(100)	SOS 配信システムから戻した FOI の ID （RegisterSensor を作成する際、設定ファイルに指定した FOI_ID_PATTERN を基づき生成する）
last_timestamp	timestamp	最後のデータ取得した時間
procedure_id	varchar(100)	SOS 配信システムから戻した Procedure の ID（RegisterSensor を作成する際、設定ファイルに指定した PROCEDURE_ID_PATTERN を基づき生成する）
date_column_index	int	データを取込際に日付データのコラム位置

(3) **sensor_observation_setting**

テーブル名	sensor_observation_setting
説明	観測データ設定テーブル

フィールド名	データ型	説明
id[PK]	int	自動インクリメント Id
table_name	varchar	連携するセンサテーブル名
ref_sensor_id	int	連携するセンサ Id
csv_column_index	varchar	CSV データの列位置
calculate_expr	double	計算式（変数「obsVal」は取得した観測値） ⁶
phenomenon_id	double	RegisterSensor 用の PhenomenonId
offering_id	varchar	RegisterSensor 用の OfferingId
uom	timestamp	単位

⁶ フィールド calculate_expr の関数に関して、下記のリンクに載っている演算関数の使用は可能である。例: abs(obsVal)+1234/5

http://www.objecthunter.net/exp4j/#Built-in_functions