

DATABÁZE A WEBOVÉ SLUŽBY PRO SENZOROVÁ DATA

Michal KEPKA¹, Jan JEŽEK²

^{1,2} Katedra matematiky – oddělení geomatiky, Fakulta aplikovaných věd, ZČU v Plzni, Univerzitní 22, 306 14, Plzeň, Česká republika
{mkepka, jezekjan}@kma.zcu.cz

Abstrakt

V době, kdy je okolo nás množství nejrůznějších senzorů a kdy je snadné jejich měření volně publikovat v prostředí internetu, je vhodné a zároveň žádoucí poskytovat přístup k těmto měřeným datům pomocí určitého standardizovaného rozhraní.

Článek se věnuje správě a možnostem efektivního poskytování dat z databáze senzorových měření. Předmětem popisu je optimální databázové řešení, jež umožňuje rychlé ukládání i získávání velkého objemu dat v reálném čase. Článek se dále zabývá standardem Open Geospatial Consortium (OGC) Sensor Observation Service (SOS), který umožňuje standardizovaný přístup k různorodým datům ze senzorů, a popisuje implementaci základních operací tohoto standardu. V neposlední řadě jsou popsány i možnosti klientského prostředí pro získávání a vizualizaci senzorových dat.

Práce vychází ze zkušenosti z realizace reálných aplikací zaměřených na monitorování pohybujících se objektů a sledování hydrometeorologických jevů pro účely zemědělství. V rámci tohoto článku bude dán větší prostor implementaci služby založené na standardu SOS. Tato služba komunikuje prostřednictvím protokolu HTTP pomocí požadavků a odpovědí v jazyce XML. Pro přístup k datům byla vytvořena webová služba, která zpracovává příchozí požadavky a vytváří na ně odpovídající odezvy. Pro implementaci byla zvolena technologie mapování XML dokumentů na objekty tříd programovacího jazyka Java (JAXB). Článek shrnuje výhody i nevýhody použitého řešení.

Výsledkem implementace je prototyp webové služby, který umožňuje standardizovaný přístup k již existující senzorové databázi. Zmíněn je také alternativní (proprietární) přístup pomocí rozhraní na bázi HTTP GET a formátu JSON.

Abstract

Today there are a number of different sensors around us and it is easy to freely publish sensor measurements in the Internet environment. It is appropriate and desirable to provide access to sensor data by standardized interface.

The article focus at the management and possibilities how to effectively store, query and publish sensor data from the database. The subject of description is the best database solution that allows fast storage and fast retrieval of large amounts of data in real time. The article further deals with the Open Geospatial Consortium (OGC) Sensor Observation Service (SOS) standard that provides standardized access to diverse sensor data, and describes the implementation of core operations of this standard. Finally we described the possibilities of client environment for obtaining and visualization of sensor data.

The work is based on experience from implementation of real world applications focused moving objects tracking and monitoring of hydrometeorological phenomena for purposes of agriculture. In the scope of this article we will focus mainly on the implementation of the service based on the OGC standard - SOS. This service communicates through HTTP protocol with requests and responses in XML. To access the data there has been the web service created that processes incoming requests and creates the corresponding responses. For the implementation we chose technology based of binding XML documents to Java classes (JAXB). This article summarizes the advantages and disadvantages of this solution.

The result of implementation is a prototype of a web service that provides standardized access to existing sensor database. We mentioned also an alternative (proprietary) approach based on HTTP GET and JSON format.

Klíčová slova: Databáze; senzor; observace; webová služba; Sensor Observation Service.

Keywords: Database; sensor; observation; web service; Sensor Observation Service.

ÚVOD

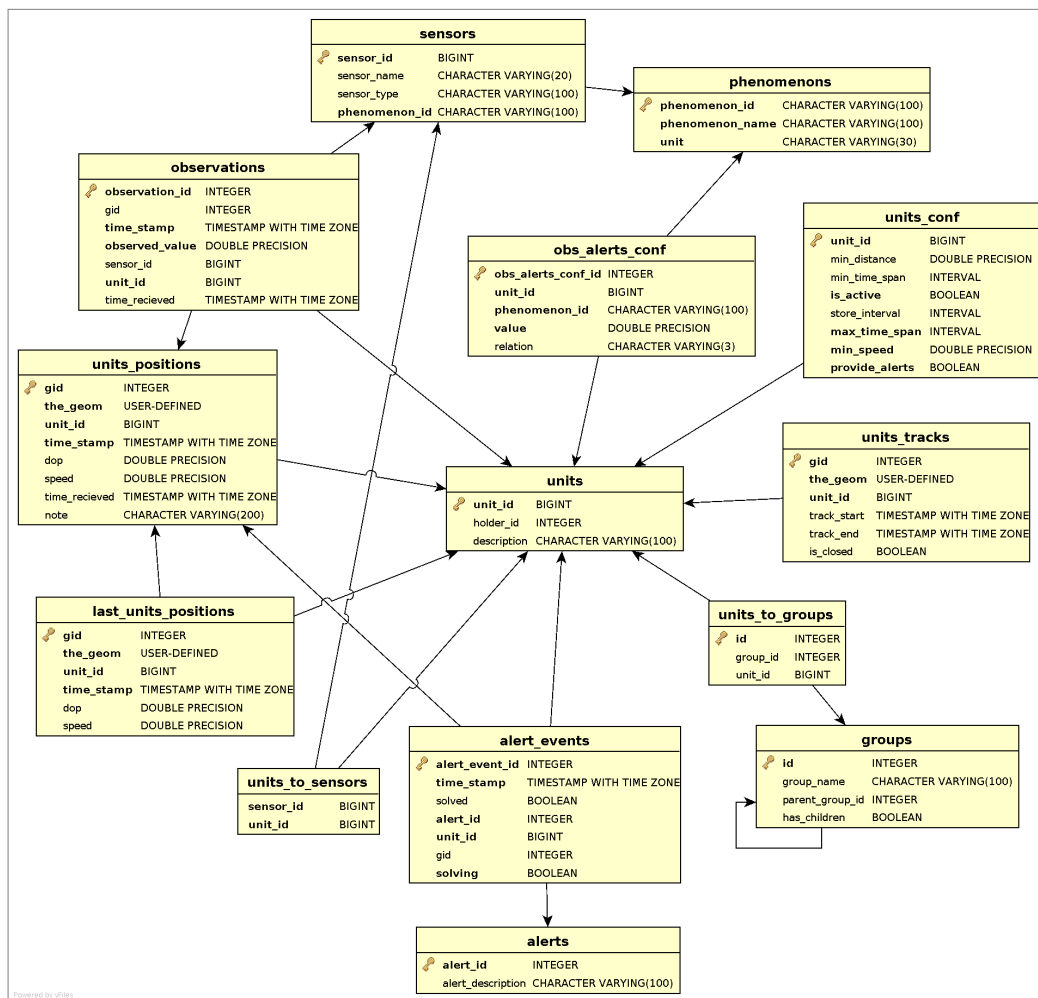
Senzory jsou v poslední době využívány ve stále více odvětvích lidské činnosti. Můžeme se setkat se senzory měřícími povětrnostní podmínky, dopravní situaci na křižovatce, zaznamenávající polohu vozidla nebo i sledující stav pacienta v nemocnici. Proces měření je velmi často rychlý a většinou plně automatizovaný, je tedy možné jej provádět opakovaně s periodou v řádech desítek vteřin nebo minut. Což znamená, že takto častá měření produkují velké množství dat. Zároveň s tímto vyvstává otázka nejen efektivního sběru dat, ale i ukládání, správy a v neposlední řadě snadné publikace například v prostředí internetu. Jak ale efektivně spravovat a následně publikovat data, aby byla dostupná a hlavně srozumitelná pro co největší množství uživatelů? Příspěvek přibližuje možné řešení v podobě sensorové aplikace SensLog se standardizovaným rozhraním konsorcia Open Geospatial Consortium (OGC) Sensor Observation Service (SOS) [1]. Aplikace SensLog byla několik let vyvíjena na ZČU ve spolupráci s firmou HS-RS a sdružením CCSS. V současnosti je aplikace využívána pro meteorologická čidla, případně pro sledování pohybu vozidel. Protože ovšem datový model pro ukládání observací nebyl tvořen pro konkrétní podobu senzorů a rozhraní standardu SOS nově přidané k SensLogu je od počátku zamýšleno jako univerzální rozhraní pro sensorová data, lze do budoucna uvažovat o nasazení i pro jiné typy senzorů.

Nejprve je vhodné přiblížit samotnou sensorovou síť, pro kterou byla aplikace původně vytvořena. Síť se skládá z měřících senzorů, což jsou jednoduchá čidla, která sledují vybrané jevy a posílají pomocí různých druhů připojení (bluetooth, RFID, kabel) změřená data tzv. jednotkám. Jednotky jsou mobilní zařízení, která sbírají data ze senzorů a přeposílají je pomocí internetového připojení (nejčastěji GPRS) dále na server. Server jako poslední součást data ukládá, analyzuje a následně publikuje. Senzory spolu s jednotkami, mohou být umístěny staticky, např. někde v terénu na poli, nebo mohou být vybaveny i GPS modulem, pomocí kterého lze zaznamenávat pohyb čidel, např. pokud jsou namontovány do vozidla.

1. DATABÁZE

Datový model databáze aplikace SensLog je odlišný od standardizovaného modelu pro observace specifikace OGC Observation&Measurement (O&M) [2], ale poskytuje oproti modelu O&M více funkcí. Zjednodušený diagram datového modelu je zobrazen na obrázku 1.1. Model byl implementován v prostředí relační databáze PostgreSQL a její prostorové nadstavby PostGIS. Protože databáze musí spravovat velké množství dat, která neustále narůstají, jsou data před vložením předzpracována triggerem pro následné rychlejší dotazování.

SŘBD sensorové databáze pracuje, stejně jako ostatní software, ve výpočetním prostředí, ve kterém je možný výskyt chybových stavů nebo výpadků. K minimalizování následků poruch v síti bylo nutné zavést speciální opatření. Pro zajištění škálovatelnosti a vysoké míry dostupnosti bylo využito mechanismu replikace a clusterování pomocí middleware Pgpool-II, který pracuje mezi PostgreSQL serverem a databázovým klientem. Pgpool-II optimalizuje připojení k databázi a poskytuje především poolování databázových připojení, replikaci, dále také Load Balance a paralelní dotazování.



Obr. 1.1 Diagram datového modelu

Nejdůležitější tabulky modelu zobrazeného na obrázku 1.1 jsou uvedeny v následujícím seznamu:

- units – seznam měřících jednotek umístěných v síti.
- sensors – seznam typů senzorů použitých v síti. Typy senzorů jsou připojeny k jednotkám relací m:n, rozkladová tabulka je units_to_sensors.
- phenomenons – seznam sledovaných jevů.
- observations – seznam měřených hodnot pro daný senzor a jednotku v určitém časovém okamžiku.
- units_positions – seznam všech pozic jednotek v čase.
- alerts – nabídka možných výjimečných stavů, které mohou v síti nastat.
- alert_events – seznam nastalých výjimečných událostí a stav jejich řešení v určitý časový okamžik.
- groups – seznam skupin uživatelů, kteří mohou přistupovat do jim povolené části senzorové sítě. Skupiny a jednotky jsou opět v relaci m:n, rozkladová tabulka je units_to_groups.

Z obrázku 1.1 a seznamu tabulek je vidět rozšířená funkcionální vytvořeného modelu oproti standardizovanému modelu O&M, která spočívá v zavedení skupin uživatelů (groups) a výjimečných stavů v síti (alerts).

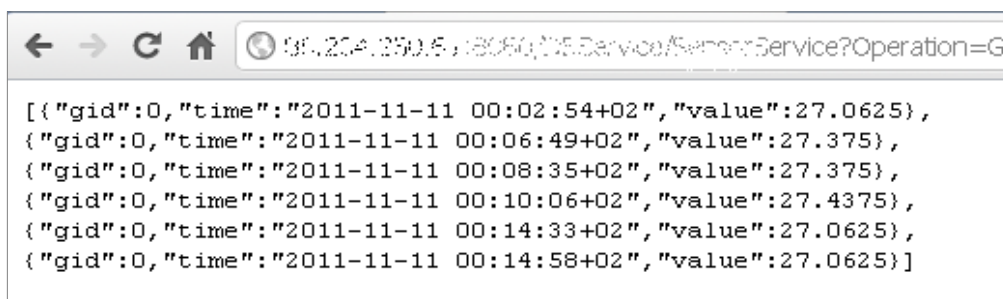
2. UKLÁDÁNÍ A PUBLIKOVÁNÍ DAT PROPRIETÁRNÍM ROZHRAŇÍM

Pro ukládání příchozích dat bylo vytvořeno aplikační programové rozhraní (API) v jazyce Java využívající protokol HTTP. Dále byly vytvořeny HTTP GET metody pro vkládání observací, pozic jednotek a výjimečných stavů. Vkládání je v současnosti řešeno přes jednoduchý proprietární protokol, ovšem v budoucnu jej bude možné zavést pomocí složitějších transakčních metod služby SOS. V již instalovaných sítích je vkládání nových observací řešeno přes zprostředkující službu, která přijímá měření ve zhuštěném proudu bitů, dekoduje jej a následně vytváří odpovídající GET požadavky protokolu HTTP pro aplikaci SensLog.

Publikování dat je řešeno přes skupinu webových služeb vytvořených v jazyce Java. Tyto služby poskytují rozhraní přes GET metody protokolu HTTP. Služby využívají přihlašování, uživatel má tedy přístup jen k jemu povoleným datům, přístupová práva jsou řešena přiřazením uživatele do určitých skupin. Výstupem služeb jsou požadovaná data ve formátu JSON, který sice není pro sensorová data standardizován, ovšem díky jednoduchosti jej lze snadno zpracovávat jak v Javě tak i v uživatelském rozhraní založeném na technologii AJAX. Ukázkový požadavek pro získání observací z typu senzoru 3040 na jednotce 1010 v období od 11. 11. 2011 0.00 do 11. 11. 2011 0.15 vypadá následovně:

```
Service?Operation=GetObservations
&unit_id=1010
&sensor_id=3040
&from=2011-11-11+00:00
&to=2011-11-11+00:15
```

Výsledkem požadavku je odpověď zobrazená na obrázku 2.1.



Obr. 2.1 Podoba odpovědi požadavku GetObservations

Z obrázku 2.1 je patrné, že výstup je relativně dobře dešifrovatelný, ovšem není standardizován. Standardizovaný výstup z databáze je v současnosti v omezené míře zajištěn prototypem služby SOS, která bude podrobněji popsána ve 3. kapitole.

3. PUBLIKOVÁNÍ DAT STANDARDIZOVANÝM ROZHRAŇÍM

V současnosti je velice důležitá interoperabilita, proto je vhodné publikovat data v takovém formátu, aby byl čitelný pro co nejvíce klientů. Konsorcium OGC v rámci Programu interoperability vytvořilo koncept sensorových webů, které by zpřístupnily senzory pro jiné aplikace po sítích jako je internet a přes stejné protokoly, kterými je tvořen web. Koncept sensorových webů je sdružen pod hlavičku specifikací označovaných jako Sensor Web Enablement (SWE) [2]. V rámci aktivit SWE bylo definováno a testováno několik základních komponent potřebných pro sensorové weby, pro účely tohoto článku jsou nejdůležitější následující čtyři specifikace:

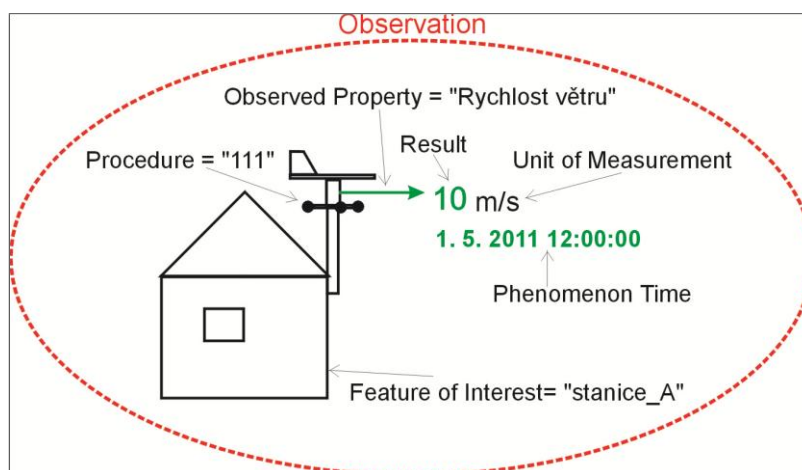
- Observations&Measurements (O&M) – definuje obecné modely a podobu kódování XML pro observace a měření.

- Sensor Model Language (SensorML) – definuje obecné modely a XML schémata pro popis senzorů a procesů spojených s měřením.
- Transducer Model Language (TML) – obecně charakterizuje snímače, jejich data, postup generování, popis převodníků, měřených nebo produkovaných jevů a přenos metadat potřebných pro pozdější zpracování dat.
- Sensor Observation Service (SOS) – služba poskytuje API pro správu senzorů a získávání měřených dat (observací).

Specifikace SOS [1] uvádí, že službu lze použít jak pro statické senzory (např. meteorologická měření na určité lokalitě), tak pro dynamické senzory (např. satelitní snímkování). Koncept samotné SOS byl již od počátku tvořen pro veškeré možné druhy senzorů, tedy i pro různorodé skupiny uživatelů. Služba SOS využívá specifikaci O&M pro popis observací a specifikace SensorML nebo TML pro popis měřících senzorů. SOS používá pro ukládání observací datový model definovaný ve specifikaci O&M, ovšem vytvořený prototyp pracuje nad vlastním datovým modelem (viz. Obr. 1.1).

3.1. Základní prvky SOS

Služba SOS je primárně určena pro přístup k observacím, proto lze observaci považovat za základní prvek. Zároveň je vhodné vysvětlit, jak je pojem observace chápán v kontextu SOS, což nejlépe vystihuje následující obrázek 3.1, který zároveň slouží jako názorný překlad termínů použitých ve specifikaci SOS.



Obr. 3.1 Význam pojmu observace [3]

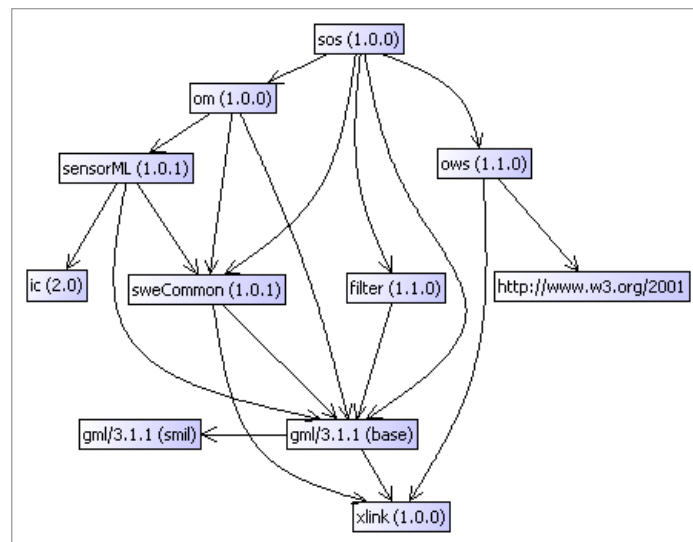
Z obrázku 3.1 plyne, že observace (observation) je určitá událost, která poskytuje výsledek (result) pozorování určitého jevu (observed property) v daný časový okamžik (phenomenon time). Observace je vztažena ke konkrétnímu bodu zájmu (feature of interest) a byla získána pomocí určitého senzoru (procedure), který daný jev měří. Pod observací si lze představit cokoliv od číselné hodnoty rychlosti větru v $\text{m}\cdot\text{s}^{-1}$, přes obrazový snímek povrchu Země z družice až po slovní hodnocení aktuálního stavu pacienta v nemocnici. Soubory spolu souvisejících observací jsou podle [1] sestavovány do kolekcí označovaných jako Observation Offerings. Typicky by se mělo jednat o soubor nepřekrývajících observací, které tvoří analogii k vrstvám jiných webových služeb.

3.2. Koncept operací SOS

Operace SOS jsou definovány ve dvou rovinách [1]. Začneme-li tvorbou observací, mluvíme o pohledu poskytovatele dat, který bude využívat metody pro vkládání informací o senzorech a následně vkládat samotné observace, tedy mluvíme o tzv. transakčním profilu operací, kam spadají operace RegisterSensor a InsertObservation. Tento profil není povinný pro každou implementaci SOS. Na druhé straně se nacházejí odběratelé dat, kteří mají zájem určitým způsobem uložená data získávat. V tomto případě mluvíme o tzv.

pohledu odběratele, kam jsou zařazeny operace tzv. základního profilu GetCapabilities, DescribeSensor a GetObservation. Tyto tři operace jsou povinné pro každou implementaci SOS. Operace GetCapabilities slouží k získání podrobných informací o samotné službě, jako je popis implementovaných operací a jejich parametrů, způsob komunikace, nebo poskytované nabídky observací. Pro detailní popis konkrétního senzoru je určena operace DescribeSensor. Samotné observace lze získat pomocí metody GetObservation, která zároveň nabízí možnost filtrovat množinu všech observací podle času, prostoru nebo naměřené hodnoty. Ve specifikaci SOS [1] je definováno ještě několik dalších metod, které jsou nepovinné a slouží například pro popis použitých datových typů (DescribeObservationType, DescribeFeatureOfInterest, atp.).

Služba SOS komunikuje s klienty pomocí požadavků a odpovědí v jazyce XML, které jsou posílány pomocí POST metody protokolu HTTP. Podoba těchto XML dokumentů je zajištěna prostřednictvím schémového jazyka W3C XML Schema (XSD), schémata jsou součástí implementačního standardu SOS [1], případně lze v aktuální verzi stáhnout ze stránek konsorcia OGC [2]. Protože SOS je součástí SWE, lze přepokládat využívání již definovaných komponent z jiných specifikací SWE, případně i jiných v rámci OGC. Propojení standardu SOS na jiné standardy OGC je zobrazeno na obrázku 3.2.



Obr. 3.2 Závislost SOS na jiných standardech [3]

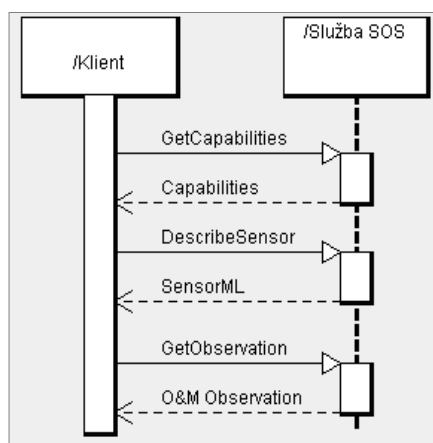
Obrázek 3.2 byl vygenerován pomocí programu OGC Schemas Browser ver. 1.0 [4], který dokáže projít schémata OGC včetně všech závislostí a vykreslit jejich strukturu do diagramu, zároveň dokáže najít a přesně lokalizovat případnou chybu ve schématech. Z obrázku 3.2 je patrné, že SOS využívá komponenty několika dalších standardů nejen ze SWE.

3.3. Možnosti implementace SOS

Služba SOS je aplikace na straně serveru, která přijímá klientské požadavky, vyzvedává data z databáze a vytváří odpovědi, které následně posílá zpět klientovi. Odpovědí je buď standardní dokument s požadovanými daty, nebo výjimka, pokud se při zpracování požadavku vyskytne nějaká chyba. Vzhledem k tomu, že podoba požadavků a odpovědí je předem známa a daná XSD schématy, existuje několik cest pro možnou tvorbu implementace. Jednou z možností je využití transformace za pomoci Extensible Stylesheet Language Transformations (XSLT), kdy jsou proprietární XML výstupy z databáze, které dokáže vytvořit většina dnes používaných databázových systémů, transformovány do standardizované odpovědi. Druhou možností je mapování (tzv. binding) schémat na třídy programovacího jazyka pomocí generovacího nástroje (tzv. schema compiler). V tomto případě dojde k odstínění od samotného XML a s dokumenty se zachází jako s běžnými objekty programovacího jazyka.

4. VLASTNÍ IMPLEMENTACE SOS SLUŽBY

V případě realizované implementace SOS, která je součástí aplikace SensLog, bylo použito postupu mapování XML schémat na třídy programovacího jazyka Java. V aplikaci bylo využito hotových tříd vygenerovaných nástrojem Java Architecture for XML Binding (JAXB), které byly vytvořeny v rámci projektu OGC Schemas and Tools vývojáře Alekseie Valikova [5] a jsou dostupné pod BSD-licencí v hlavním úložišti nástroje Maven. Činnost webové služby tedy spočívá ve čtení příchozích požadavků a jejich převodu na strom objektů, čtení XML dokumentu je v terminologii JAXB označováno jako unmarshalling. Dále ve vyzvednutí dat z databáze, sestavení odpovědi opět v podobě stromu objektů, následném převodu na XML (označováno jako marshalling) a odeslání klientovi. Diagram komunikace mezi službou a klientem je zobrazen na obrázku 4.1, který zároveň ukazuje druh odpovědi vrácené na konkrétní požadavek.



Obr. 4.1 Diagram komunikace klient – služba [3]

Vytvořená implementace v současnosti obsahuje povinné operace se základními parametry uvedenými v následujícím seznamu:

- GetCapabilities
 - ⤴ Požadavek – dokument GetCapabilities s povinnými parametry request a service
 - ⤴ Odpověď – dokument Capabilities s oddíly ServiceIdentification, ServiceProvider, Operations-Metadata, FilterCapabilities a Contents.
- DescribeSensor
 - ⤴ Požadavek – dokument DescribeSensor s povinnými parametry service, procedure, output-Format, version
 - ⤴ Odpověď – dokument SensorML s elementy identification, capabilities, outputs a postions
- GetObservation
 - ⤴ Požadavek – dokument GetObservation s následujícími podporovanými hodnotami parametrů:
 - service: SOS
 - version: 1.0.0
 - srsName: urn:ogc:def:crs:EPSG:4326
 - offering: podle nabídky hodnot z Capabilities dokumentu
 - eventTime: TM_During
 - procedure: podle nabídky hodnot z Capabilities dokumentu

- observedProperty: podle nabídky hodnot z Capabilities dokumentu
 - responseFormat: text/xml; subtype="om/1.0.0"
 - resultModel: om:Observation
 - responseMode: inline
- ▲ Odpověď – dokument ObservationCollection s jedním elementem Observation s elementy samplingTime, procedure, observedProperty, result.

V současnosti není plně podporováno filtrování observací, implementováno je prozatím pouze časové, kdy je zadán počátek a konec požadovaného časového úseku v podobě elementu TM_During. Protože při implementaci nebyl použit standardní model pro observace ale model vlastní (viz Obr. 1.1), neshoduje se terminologie použitá ve vlastním modelu s terminologií používanou v SOS. Vzájemné přiřazení jednotlivých termínů ukazuje následující tabulka 4.1.

Tab. 4.1 Ekvivalence pojmů

SOS	SensLog model
Sensor	Unit
Procedure	Unit_id
ObservedProperty	Phenomenon
ObservationOffering	Group

K tomu, aby bylo možné získat observace z konkrétního senzoru (ve smyslu SensLogu), je potřeba do požadavku GetObservation vyplnit parametry procedure a observedProperty. Příslušnost konkrétních senzorů k jednotkám lze získat z odpovědi požadavku DescribeSensor v sekci Outputs.

Na následujících obrázcích bude zobrazen ukázkový požadavek GetObservation za použití jednoduchého HTML klienta a odpověď jak v podobě surového XML výstupu tak i v podobě po připojení k externímu klientovi v HSLayers.

Obrázek 4.2 představuje požadavek na měření teploty z jednotky číslo 3001 zařazené do skupiny admin ze dne 10. 11. 2011 v době od 10 do 12 hodin.

Testovací SOS klient

Požadavek:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetObservation xmlns="http://www.opengis.net/sos/1.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sos/1.0
  http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
  service="SOS" version="1.0.0" srsName="urn:ogc:def:crs:EPSG:4326">

  <offering>admin</offering>

  <eventTime>
    <ogc:TM_During>
      <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
      <gml:TimePeriod>
        <gml:beginPosition>2011-11-10T10:00:00+01:00</gml:beginPosition>
        <gml:endPosition>2011-11-10T12:00:00+01:00</gml:endPosition>
      </gml:TimePeriod>
    </ogc:TM_During>
  </eventTime>

  <procedure>3001</procedure>

  <observedProperty>Temperature</observedProperty>

  <responseFormat>text/xml;subtype="om/1.0.0"</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>inline</responseMode>

</GetObservation>
```

Obr. 4.2 Ukázkový požadavek GetObservation

Odpověď, která je vrácena klientovi při zaslání požadavku přes HTML formulář, je zobrazena na obrázku 4.3. Konkrétní měřená data jsou umístěna v podobě textového bloku v elementu `<swe:values>` a na obrázku 4.3 jsou zvýrazněna červeným rámečkem, způsob dekódování textového bloku je taktéž součástí odpovědi, v tomto případě se jedná o vzor „časováZnámka;měřenáHodnota@“.

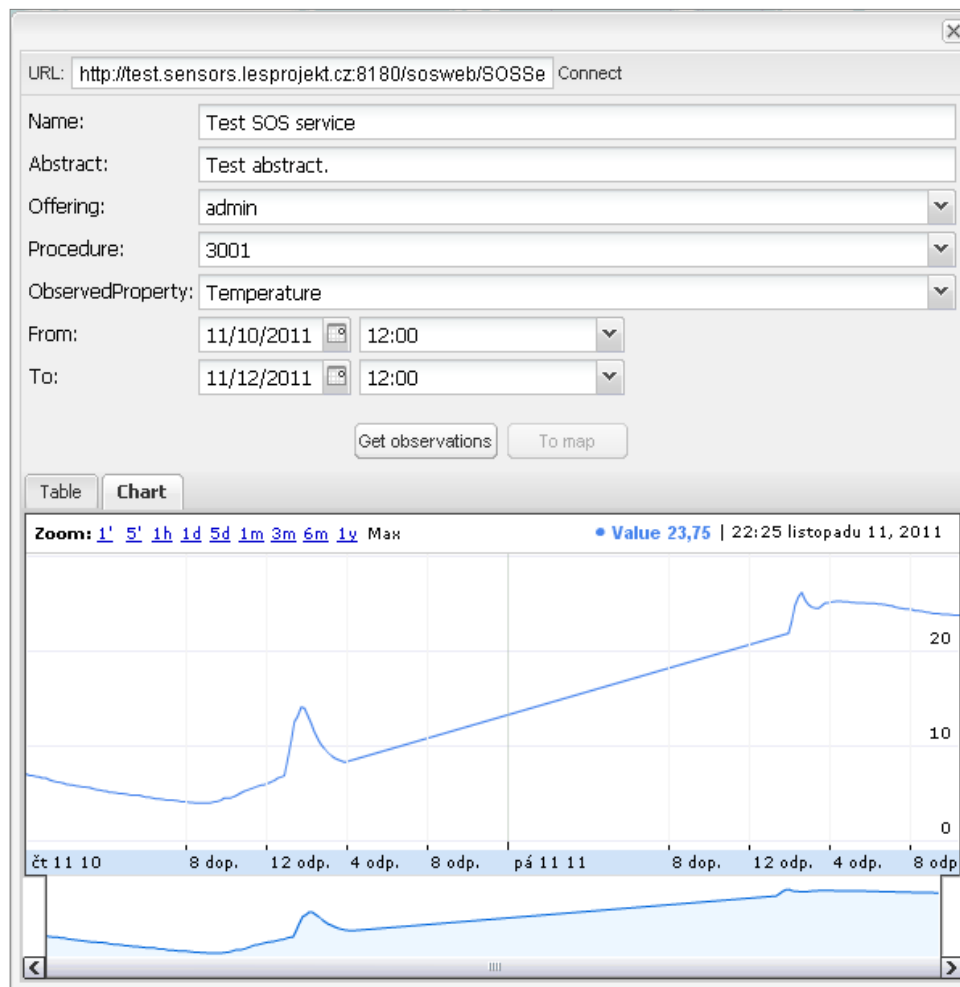
```

▼<ns6:ObservationCollection xmlns:ns1="http://www.opengis.net/ows/1.1"
  xmlns:ns2="http://www.opengis.net/sos/1.0" xmlns:ns3="http://www.opengis.net/ogc"
  xmlns:ns4="http://www.opengis.net/gml" xmlns:ns5="http://www.w3.org/1999/xlink"
  xmlns:ns6="http://www.opengis.net/om/1.0" xmlns:ns7="http://www.opengis.net/swe/1.0.1"
  xmlns:ns8="http://www.opengis.net/sensorML/1.0.1" xmlns:ns9="urn:us:gov:ic:ism:v2"
  xmlns:ns10="http://www.w3.org/2001/SMIL20/" xmlns:ns11="http://www.w3.org/2001/SMIL20/Language"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/om/1.0
  http://schemas.opengis.net/om/1.0.0/om.xsd">
  ▼<ns6:member ns5:type="simple">
    ▼<ns6:Observation>
      ▼<ns6:samplingTime ns5:type="simple">
        ▼<ns4:TimePeriod>
          <ns4:beginPosition>2011-11-10T10:04:06+01:00</ns4:beginPosition>
          <ns4:endPosition>2011-11-10T11:54:06+01:00</ns4:endPosition>
          </ns4:TimePeriod>
        </ns6:samplingTime>
        <ns6:procedure ns5:href="3001" ns5:type="simple"/>
        <ns6:observedProperty ns5:type="simple"/>
        <ns6:featureOfInterest ns5:href="urn:ogc:def:feature:OGC-SWE:3:transient"/>
      ▼<ns6:result>
        ▼<swe:elementCount xmlns:swe="http://www.opengis.net/swe/1.0.1">
          ▼<swe:Count>
            <swe:value>12</swe:value>
          </swe:Count>
        </swe:elementCount>
        ▼<swe:elementType xmlns:swe="http://www.opengis.net/swe/1.0.1" name="Components">
          ▼<swe:SimpleDataRecord>
            ▼<swe:field name="Time">
              <swe:Time definition="urn:ogc:data:time:iso8601"/>
            </swe:field>
            ▼<swe:field name="Mort int temp">
              ▼<swe:Quantity definition="Temperature">
                <swe:uom code="C"/>
              </swe:Quantity>
            </swe:field>
          </swe:SimpleDataRecord>
        </swe:elementType>
        ▼<swe:encoding xmlns:swe="http://www.opengis.net/swe/1.0.1">
          <swe:TextBlock blockSeparator="@" decimalSeparator="." tokenSeparator=";"/>
        </swe:encoding>
        ▼<swe:values xmlns:swe="http://www.opengis.net/swe/1.0.1">
          2011-11-10T10:04:06+01:00;4.5@2011-11-10T10:14:06+01:00;4.5625@2011-11-
          10T10:24:06+01:00;4.6875@2011-11-10T10:34:06+01:00;4.875@2011-11-10T10:44:06+01:00;5.0625@2011-
          11-10T10:54:06+01:00;5.25@2011-11-10T11:04:06+01:00;5.375@2011-11-10T11:14:06+01:00;5.5@2011-
          11-10T11:24:06+01:00;5.625@2011-11-10T11:34:06+01:00;5.75@2011-11-10T11:44:06+01:00;5.875@2011-
          11-10T11:54:06+01:00;5.9375
        </swe:values>
      </ns6:result>
    </ns6:Observation>
  </ns6:member>
</ns6:ObservationCollection>

```

Obr. 4.3 Ukázková odpověď na požadavek GetObservation

Z obrázku 4.3 je patrné, že výstup v podobě XML dokumentu není příliš určen pro přímé čtení člověkem. Pokud ovšem dojde k připojení služby do externího grafického klienta, lze již data získávat v čitelnější formě, jak dokládá následující obrázek 4.4.



Obr. 4.4 Ukázka připojení grafického klienta

Na obrázku 4.4 je zobrazen delší časový úsek, než jaký byl požadován v původním GetObservation požadavku na obrázku 4.2, ovšem jedná se o stejná data.

5. ZÁVĚR

Příspěvek měl za cíl představit aplikaci pro správu a publikaci sensorových dat SensLog včetně vytvořeného datového modelu pro jejich ukládání. Aplikace SensLog umožňuje publikaci dat jak pomocí proprietárního rozhraní založeného na formátu JSON (při využití plné funkcionality datového modelu) tak přes standardizované rozhraní služby SOS (v základní funkčnosti pro přístup k observacím založené na jazyce XML). Výhodou proprietárního rozhraní je možnost tvorby jednodušších klientů tvořených na míru specifickým účelům, výhodou rozhraní pomocí služby SOS je právě její standardizovanost. Konkrétní ukázkou implementace služby SOS lze nalézt na URL adrese projektu [6].

Prezentovaný datový model aplikace SensLog se odlišuje od standardizovaného modelu pro observace specifikace O&M, poskytuje oproti němu větší funkcionalitu v podobě např. alertů nebo seskupování uživatelů a jednotek.

Dalším cílem příspěvku bylo představení samotné služby SOS, ačkoliv byla standardizována již v roce 2007, její používání není do dnešního dne příliš rozšířeno. Důvod lze spatřit ve standardu jako takovém právě proto, že byla služba od počátku zamýšlena jako univerzální pro publikaci dat různého charakteru z různorodých senzorů. Univerzálnost služby s sebou nese velkou variabilitu v podobě předávaných požadavků a odpovědí a tím velkou komplexnost a komplikovanost schémat těchto dokumentů. Vznikl tedy

univerzální standard, který ovšem v konečném důsledku pro velkou složitost není snadné jednoduše implementovat.

LITERATURA

[1] OGC 06-009r6. Sensor Observation Service: OpenGIS® Implementation Standard. Wayland (Mass.): Open Geospatial Consortium, Inc., 26. 10. 2007. xiv, 90 s. Dostupné z WWW: <<http://www.opengeospatial.org/standards/sos>>.

[2] Open Geospatial Consortium. OGC® Standards and Specifications [online]. 2011 [cit. 2011-11-15]. Dostupné z WWW: <<http://www.opengeospatial.org/standards>>.

[3] KEPKA, Michal. Implementace standardu Sensor Observation Service. Plzeň: Západočeská univerzita v Plzni. 2011. 77 s. Diplomová práce (Ing.).

[4] TAMAYO, Alain; GRANELL, Carlos; HUERTA, Joaquín. OGC Schemas Browser: Visualizing OWS' XML Schemas. In Geospatial Thinking: Proceedings of AGILE 2010 [online]. [Guimarães]: [s.n.], 2010 [cit. 2011-05-01]. Dostupné z WWW: http://plone.itc.nl/agile_old/Conference/2010-guimaraes/ShortPapers_PDF/82_DOC.pdf ISBN 978-989-20-1953-6.

[5] VALIKOV, Aleksei. Confluence of OGC Schemas and Tools Project [online]. 2011 [cit. 2011-11-15]. Dostupné z WWW: <<http://confluence.highsource.org/display/OGCS/Home>>.

[6] ZČU v Plzni. Server application for Sensor Observation Service [online]. 2011 [cit. 2011-11-15]. Dostupné z WWW: <<http://git.zcu.cz/index.php/SOS>>.