

WIRELESS SENSOR NETWORK MONITORING

Pavel Vajsar

Master Degree Programme (2), FEEC BUT

E-mail: xvajsa01@stud.feec.vutbr.cz

Supervised by: Patrik Morávek

E-mail: xmorav08@stud.feec.vutbr.cz

ABSTRACT

The aim of this work is to design and create a web application that allows monitoring of any sensor network. The basis of the project was to design a database that will store data obtained from the sensor network. The main requirement for this database was its universality. It is important to use a database for the maximum number of sensor networks. Application for monitoring is build on J2EE (server side) and Adobe Flex 4 / Air (client side). In addition, other tools are used to support other functions and integration of these two technologies together. They are Spring, Spring Security, Hibernate, BlazeDS. Database itself is available on the MySQL server and data storage type uses InnoDB. Application use Apache Tomcat run-time environment.

1 ÚVOD

Senzorová síť je svou strukturou podobná klasické bezdrátové síti s tím, že jednotlivé koncové body tvoří stanice (počítače ovládané člověkem), nýbrž senzory rozmístěné v dané lokalitě. Tyto senzory mají předem definovanou činnost, kterou je v omezené míře možné modifikovat. Senzor ve svém okolí sleduje určené veličiny (fyzikální veličiny - teplota, tlak, osvětlení, chemické látky v ovzduší), které průběžně odesílá ke zpracování.

Cílem celého projektu je navrhnout a vytvořit aplikaci pro monitoring maximálního počtu realizací senzorových sítí. To znamená, že celé řešení by mělo být maximálně univerzální.

Základním stavebním kamenem celé aplikace je databázové úložiště. Návrh byl postaven na modelu klasické relační databáze s podporou transakcí a za podpory MySQL serveru. Zde byla univerzálnost navrženého úložiště jedna z nejdůležitějších vlastností.

Další neméně podstatnou částí je samotné uživatelské rozhraní, které pracuje s daty v databázovém úložišti. Skrze toto rozhraní jsou data získaná ze senzorové sítě interpretována do vhodné podoby. Například zobrazení jednotlivých senzorů na mapě (použito Google Maps SDK), zobrazení měřených hodnot formou grafů atd. Klientská část aplikace je od počátku stavěná tak, aby byla maximálně modulární, a byla dostupná jako webová aplikace, tak i jako desktopová.

2 ROZBOR

Nejprve bude uveden stručný popis databázového úložiště s výpisem informací, které je toto úložiště schopné uchovávat.

2.1 DATABÁZOVÉ ÚLOŽIŠTĚ

Je tvořeno 39 entitami (tabulkami), z nich každá uchovává určitý typ a počet informací ze sensorové sítě. Jednotlivé tabulky jsou vzájemně propojeny vazbami (1 : n, m : n). Databáze je schopna uchovávat níže uvedené informace :

- Senzor (stav senzoru, rozšířená pravomoc, poloha, typ a výrobce, fyzická adresa, stav baterie, typ sensorové desky, přítomnost v klusteru, přítomnost ve skupině, zprávy od sensorů, příkazy pro senzor, kolik informací může poslat v jednom paketu, jaké fyzikální veličiny může měřit, sousední senzory, změřené hodnoty a jejich kategorie).
- Linka (připojené senzory k lince, stav linky, šířka pásma, chybovost, útlum).
- Prostředí (typ prostředí, popis prostředí).
- Poloha (oblast, budova, místnost).
- Skupiny sensorů.

Dále to jsou informace, které se přímo netýkají sensorové sítě, ale spíše aplikace pro její monitoring:

- Uživatelé (uživatelské účty).
- Role (dostupné role v aplikaci, které mohou být přiřazeny uživatelům).
- Záloha (kdy a čeho byla provedena záloha).
- Informace o tom, kdy a odkud se daný uživatel do aplikace přihlásil.

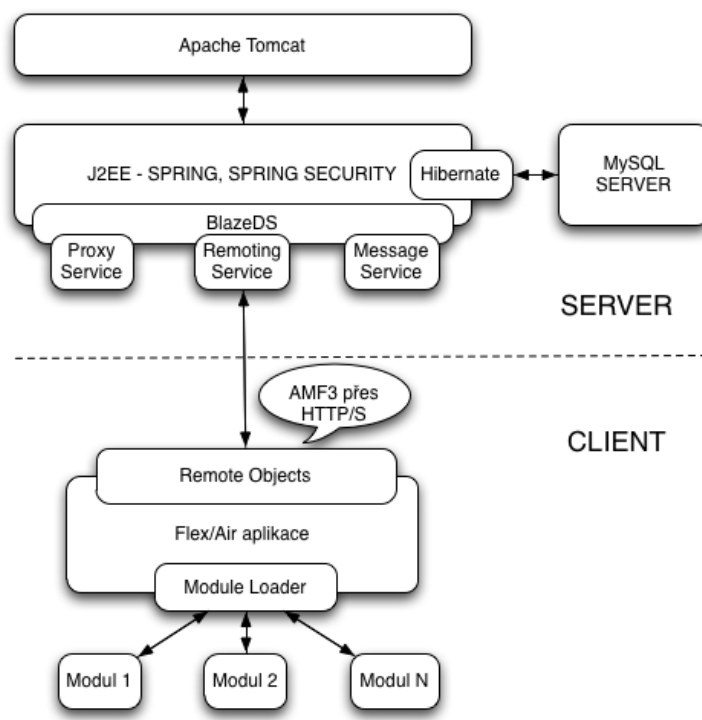
2.2 APLIKACE PRO MONITORING

Aplikace byla od začátku tvořena tak, aby kladla co nejmenší požadavky na programové a technické vybavení uživatele, který bude chtít s touto aplikací pracovat. Proto jediné, co tento uživatel potřebuje, je internetový prohlížeč, Adobe Flash plugin a připojení k síti internetu. Dále bylo snahou pro vývoj použít nástroje, které jsou dostupné zdarma, aby byly minimalizovány náklady z pohledu financí.

Jako běhové prostředí pro aplikaci byl zvolen webový server Apache Tomcat s podporou J2EE kontejneru. Vlastní jádro aplikace je tedy postavené na technologii Java a odpovídajících rámcích, které práci s touto technologií usnadňují a zpřehledňují.

Rámec Hibernate je použit pro objektově relační mapování. To umožňuje pracovat v prostředí platformy Java s entitami z relační databáze jako s objekty přímo v aplikaci. Rámec Spring do značné míry usnadní práci s těmito objekty zejména v oblasti výměny dat mezi aplikací a databází. Rozšíření rámce Spring Security je následně využito pro zabezpečení aplikace.

Jelikož jsou části klient-server z pohledu aplikace maximálně odděleny, tak bylo zapotřebí vytvořit univerzální komunikační rozhraní mezi klientem a serverem. K tomuto účelu byl použit nástroj BlazeDS, který toto rozhraní poskytuje. Tento nástroj, zjednodušeně řečeno, umožňuje volání služeb na serveru z klientské aplikace a výměnu dat mezi nimi.



Obrázek 1: Struktura aplikace

Samotná klientská část aplikace byla vytvořena, tak aby byla co nejvíce modulární a přizpůsobitelná. Hlavní částí je tedy "jádro", které provádí operace načítání funkčních modulů. Až tyto moduly poskytují služby pro monitoring bezdrátové senzorové sítě. Moduly mohou být následně upravovány pro potřeby konkrétní sítě, nebo mohou být vytvářeny moduly nové. A to moduly, které buď přímo souvisí s monitoringem sítě, nebo nikoliv. Do této klientské části aplikace byla přidána i podpora kaskádových stylů (CSS), což umožní jednoduše změnit vzhled a rozmístění prvků uživatelského rozhraní. Dále se uvažuje s podporou technologie Adobe Air.

3 ZÁVĚR

Cílem projektu je vznik maximálně univerzální aplikace, která bude použitelná ve většině případech a bude sloužit k monitorování bezdrátových senzorových sítí. Samotná aplikace je postavena na moderních technologiích, které ve výsledku poskytují kvalitní uživatelské rozhraní postavené na stabilním základu technologie J2EE.

REFERENCE

- [1] Adobe Systems Incorporated. BlazeDS [online]. 2008, 2009-10-20 [cit. 2010-03-15]. Adobe Open Source. Dostupné z WWW: <<http://opensource.adobe.com/wiki/display/blazeds/BlazeDS/>>.