

# PLANTLORE



## Vývojová dokumentace

Univerzita Karlova  
Matematicko-fyzikální fakulta  
Praha, Česká republika

Vedoucí: RNDr. Antonín Říha, CSc.

Lada Oberreiterová ladaob@seznam.cz

Jakub Kotowski fraktalek@gmx.net

Tomáš Kovařík tkovarik@gmail.com

Erik Kratochvíl krater@seznam.cz

## **Rozdělení úkolů**

### **Lada Oberreiterová**

- Webová část aplikace využívající BioCASE provider software.
- Přizpůsobení aplikace pro naše potřeby, komunikace s vývojáři BioCASE.
- Testování a opravy aplikace pro využití s databázemi PostgreSQL a Firebird.
- Namapování Plantlore databáze na standardy ABCD Schema 1.2, ABCD Schema 2.06, Darwin Core.
- Klient Plantlore: Správa metadat, Správa uživatelů, Celková historie, Historie záznamu, Převody mezi souřadnými soustavami. Implementace různých formátů pro export. Uživatelská a programátorská dokumentace těchto částí.
- Návrh grafického vzhledu webové části.
- Příprava a úpravy databázového schématu, udržování databáze.

### **Erik Kratochvíl**

- Implementace RMI mechanismu pro komunikaci klienta a serveru.
- Plantlore klient: Kostra pro Import a Export nálezů, načítání a zpracování XML dokumentů.
- Serverová a komunikační část aplikace.
- Spolupráce na implementaci vrstvy DBLayer.
- Práce na společných částech Swing GUI – ProgressBary, inteligentní ComboBoxy.
- Proces přihlašování k databázi a odhlašování.

### **Tomáš Kovařík**

- Vrstva DBLayer pro přístup k databázi, rozhraní pro zadávání dotazů, použití ORM systému Hibernate.
- Plantlore klient: Správa autorů, Správa publikací, Vytváření nové databáze, Systém pro zobrazování interaktivní nápovědy.
- Instalátor (lokalizace, instalace PostgreSQL, vytváření odkazů).
- Stránky projektu na [plantlore.berlios.de](http://plantlore.berlios.de) resp. [plantlore.sourceforge.net](http://plantlore.sourceforge.net)
- Příprava databázového schématu.

### **Jakub kotowski**

- Plantlore klient: Hlavní okno aplikace (tzv. overview se seznam nálezů, zobrazování stromu nálezů), hlavní MVC komponenta, tzv. AppCore která vytváří jednotlivé dialogy, Dialog pro přidávání a editaci nálezů, vyhledávání nálezů, zobrazování a tisk schedy a seznamu nálezů. Nastavení, načítání konfiguračních souborů.
- Instalátor (základní kostra – výběr balíčků, jednotlivé informační panely).
- Práce na společných částech Swing GUI – ProgressBary.
- Stránky projektu na [plantlore.berlios.de](http://plantlore.berlios.de) resp. [plantlore.sourceforge.net](http://plantlore.sourceforge.net)
- Správa systému pro správu verzí (SVN).

## **Společné úkoly**

- Komunikace se zadavateli – Botaniky Jihočeské pobočky ČBS
- Hledání informací o jiných podobných projektech, botanických programech používaných v České republice.

## **Důvody zpoždění projektu**

Komunikace s budoucími uživateli byla složitá a neefektivní, zejména zpočátku. Požadavky byly špatně interpretovány jednou nebo druhou stranou. Přispěla k tomu i neznalost botanické problematiky a aplikací běžných v této oblasti – nebylo nám vždy jasné zamýšlené použití. Zásadní problém byl i nedostatek zkušeností s používanými technologiemi – přetrvávající nebo dokonce opakující se problémy s RMI, Hibernate, databázovými systémy, Swingem.

## **Instalátor**

Pro Plantlore je použit instalátor IzPack 3.8. Jedná se o multiplatformní instalátor napsaný v Javě, což vyhovuje zejména proto, že se zjednodušila příprava instalátoru pro různé platformy. Instalátor byl otestován na platformě Win32, Linux a Apple Macintosh.

Součástí instalace pro platformu Windows je instalace databázového systému PostgreSQL, pro který jsme využili nativní instalátor dodaný s PostgreSQL pro Windows, který je možné spustit v tichém režimu a zadat mu parametry na příkazové řádce. Nevýhodou je nutnost používat instalační program distribuovaný pro Microsoft Windows msiexec pro spouštění msi instalátorů.

Instalátor je lokalizovaný do dvou jazyků, češtiny a angličtiny

## **Hibernate**

Hibernate je systém pro objektově-relační mapování. Umožňuje při práci s relační databází použít objektově orientovaný přístup a pracovat se záznamy v databázi jako s objekty. Jeho nasazení je ideální v kombinaci s jazykem Java, který je postavený na myšlence objektově orientovaného vývoje. Další výhodou, kterou Hibernate přináší, je přenositelnost mezi databázovými systémy díky tomu, že úplně odděluje SQL příkazy (které se v různých databázových systémech více či méně liší) a vlastní kód aplikace. Při jeho použití nemusí programátor používat přímo jazyk SQL, ale může využít pestrou nabídku API, které Hibernate poskytuje včetně vlastního dotazovacího jazyka podobného SQL (který se nazývá HQL – Hibernate Query Language).

Tyto výhody se sebou samozřejmě nesou i řadu nevýhod. Programátor je omezený na to, co poskytuje Hibernate, a nemůže naplno využít potenciál použitého Databázového systému (bez ztráty přenositelnosti).

## **Naše zkušenosti s Hibernate**

Před prací na projektu jsme neměli žádné zkušenosti se systémem Hibernate. Jeho použití nám v některých ohledech pomohlo, ale také jsme narazili na závažné problémy v kombinaci Hibernate a jiných technologií. Nastíníme některé z těchto problémů.

### **1. Hibernate a RMI**

Použití Hibernate a Remote Method Invocation se ukázalo problematické, neboť jsme byli nuceni psát vrstvu, která byla mezi Hibernate a samotnou aplikací, aby bylo možné zabezpečit posílání objektů mezi klientem a serverem (pomocí RMI nebylo rozumně možné posílat objekty Hibernate). Další problém se ukázal při přenosu tzv. holder objektů, které

zapouzdřují data z databáze do objektů jazyku Java. Hibernate tyto objekty používá pro ukládání dat do perzistentního úložiště a vadí mu, pokud jsou serializovány a posílány pomocí mechanismu RMI ze serveru na klienta a zpátky.

2. Hibernate a DB systém FirebirdSQL – Databázový systém Firebird (původní systém Interbase společnosti Borland který byl uvolněn jako open source) v poslední stabilní verzi 1.5.x neimplementuje některé funkce které bylo potřeba pro úspěšné použití v projektu v kombinaci s Hibernate. V současné době se dokončuje verze 2.0, která by měla takovéto problémy odstranit, ale rozhodli jsme se ji nakonec nepoužít (i když Plantlore by měla s verzí 2.0 fungovat), neboť se jednalo ještě stále o betaverzi. Problémy, na které jsme narazili, jsou popsány v této dokumentaci v části Databázový systém.

## **Databázový systém**

Náš původní cíl byl, aby Plantlore fungovala nad různými databázovými systémy. Nakonec jsme se ale počas vývoje rozhodli, že použijeme systém PostgreSQL díky jeho kompatibilitě s jinými použitými technologiemi, dobré dokumentaci a jednoduché administraci. Důvody, kvůli kterým jsme se rozhodli nepoužít jiné databáze jsou většinou způsobeny nedostatky jiných databázových systémů, příp. jejich nevhodností pro náš projekt. O kterých systémech jsme uvažovali:

1. FirebirdSQL – Jak je poznamenáno v části o systému Hibernate, je Firebird v současné stabilní verzi nekompatibilní s tím, jak používáme v Plantlore systém Hibernate. Mezi problémy, na které jsme narazili patří:
  1. Nemožnost vykonat velké SQL SELECT dotazy, jejichž plán vykonání je větší než určitá mez. Problém vyniká v Hibernate, který SQL dotazy generuje sám a často používá velké dotazy s mnoha tabulkami. Toto chování je možné do určité míry upravit, ale dochází při tom k dalším problémům. Firebird 2.0 toto již řeší.
  2. Podle našeho názoru diskutabilní interpretace SQL standardu, kdy není možné použít přejmenování (SELECT sloupek AS popis\_sloupku...) ve vnořených SELECT dotazech. Jiné databázové systémy toto implementují správně. Bohužel Hibernate toto chování využívá a není možné ho vypnout nebo změnit.
2. MySQL – Tento systém byl testován v úvodních fázích vývoje, ale narazili jsme na několik tradičních problémů – od vnořených dotazů až po správu rolí a uživatelů. Systém se vyvíjí a v posledních major verzích implementuje mnoho požadovaných vlastností, rozhodli jsme se ale, že existují lepší možnosti
3. Oracle – databázový systém Oracle je sice dostupný pro osobní použití, jedná se ale stále o proprietární produkt který je sice vyspělý, ale pro naše potřeby asi málo použitelný, neboť typický botanik si ho nebude instalovat na svůj domácí počítač.
4. PostgreSQL – Tento systém nás zaujal jednoduchostí administrace, dostupností dokumentace, kompatibilitou, podporovanými platformami a vcelku dobrou podporou pro kódování UTF8 s kterým jsme měli také problémy u jiných systémů.

## **Seznam použitých knihoven**

Všechny knihovny a použité zdroje jsou volně šiřitelný software distribuovaný pod různými licencemi pro vývoj open-source software. Použití těchto knihoven a zdrojů nám umožnilo věnovat maximum úsilí na vlastní vývoj Plantlore a dosáhnout co nejvyšší kvality aplikace z uživatelského hlediska.

V plantlore jsme použili následující knihovny (u každé knihovny následuje krátký popis)

- log4j-1.2.13.jar – Log4j - Nástroj pro zaznamenávání logovacích výpisů.
- jcalendar-1.3.1.jar – jCalendar - Knihovna pro výběr data z kalendáře.
- looks-2.0.2.jar - JGoodies look & feel – Knihovna pro vzhled uživatelského rozhraní v Javě.
- dom4j-1.6.jar - Dom4j – Knihovna pro parsování a práci s XML daty.
- swing-layout-1.0.jar -Matisse - Knihovna použitá pro návrh uživatelského prostředí.
- spin.jar - Knihovna pro vytváření nezamrzajících uživatelských rozhraní ve Swingu.
- jasperreports-1.2.1.jar - Jasper reports – Nástroj pro tvorbu, export a tisk reportů.
- antlr-2.7.5H3.jar – Apache Ant – Nástroj pro sestavování a spouštění build-skriptů.
- hibernate3.jar – Systém pro objektově relační mapování.
- jh.jar – Java Help 2.0 – Systém pro tvorbu interaktivní nápovědy.
- jxl.jar – Nástroj pro spracování XLS dokumentů.
- mysql-connector-java-3.1.12-bin.jar – JDBC ovladač pro MySQL.
- postgresql-8.0-311.jdbc2.jar – JDBC ovladač pro PostgreSQL.
- jaybird-full-2.1.0.jar – JDBC ovladač pro Firebird.