

SEMESTRÁLNÍ PRÁCE PRO PŘEDMĚT 4IT450

# CASE nástroje pro jazyk UML

---

**Petr Vomáčka, Jan Skružný, Martin Šetek, Marek Klusák**

**LS 2008/2009**

# Obsah

Úvod .....	4
CASE.....	5
UML (Unified Modeling Language).....	5
Součásti UML.....	5
Rational Rose.....	7
Rational Rose Data Modeler.....	7
Operační systémy a příslušné hardwarové platformy .....	7
Rational Rose Developer for Java.....	7
Rational Rose Developer for UNIX.....	8
Operační systémy a příslušné hardwarové platformy .....	8
Rational Rose Developer for Visual Studio.....	8
Rational Rose Modeler .....	8
Rational Rose Technical Developer .....	9
Operační systémy a příslušné hardwarové platformy .....	9
Rational Software Architect for WebSphere Software .....	10
Ukázka rozhraní .....	10
Přehled funkčnosti.....	10
Violet UML Editor .....	12
Přehled funkčnosti.....	12
Co Violet neumožňuje .....	13
Ukázka rozhraní .....	13
Shrnutí .....	15
Borland Together 2008.....	16
Podpora standardů.....	16
Klíčové vlastnosti produktu .....	16
Domain-specific language (DSL) .....	16
Business Process modelování.....	16
UML modelování .....	16
Data modelování .....	17
Pokročilé modelování.....	17
Ukázka rozhraní .....	17
Shrnutí .....	18
Umbrello UML Modeller .....	19

Shrnutí .....	20
ArgoUML.....	21
Ukázka rozhraní.....	21
Shrnutí .....	22
Open ModelSphere .....	23
Ukázka rozhraní.....	24
Shrnutí .....	24
Visual Paradigm for UML Enterprise Edition.....	25
Ukázka rozhraní.....	26
Shrnutí .....	26
Kritéria výběru vhodného řešení.....	27
Závěr .....	28
Zdroje .....	29

# Úvod

Cílem této práce je popsat a charakterizovat produkty pro tvorbu UML diagramů na trhu, ať už komerční, open source či jinak licencované. Téma této práce je v tomto semestru nové, proto zajisté nebude plně vyčerpané, což dá možnost tuto práci inovovat v dalších semestrech našim následovníkům. Na trhu existuje zajisté mnoho produktů, které jsou určeny přímo jen pro tvorbu UML, ale i rozsáhlých, mnohafunkčních produktů, díky kterým lze i přímo generovat kódy do mnoha různých programovacích jazyků. Vybrali jsme si pro charakteristiku především přední výrobce takovýchto produktů, které jsou na špičce trhu.

Nejprve si uděláme lehký teoretický úvod do problematiky CASE nástrojů, poté se podíváme na problematiku samotné UML logiky a pak přijdou na řadu jednotlivé produkty, které jsme se rozhodli charakterizovat.

## CASE

Jak jsme již výše zmínili, celým názvem zkratky CASE je Computer Aided Systems (Software) Engineering, což v překladu znamená počítačem podporované systémové (softwarové) inženýrství. Především se jedná o software, která umožňuje modelování systémů pomocí různých diagramů a generování zdrojového kódu z těchto modelů. Historii zde nebudu uvádět, protože není předmětem této práce, jen zde zmíním, že sahá někam do 60. Let minulého století.

Základní členění CASE nástrojů vyplývá ze životního cyklu projektu, tedy v jaké fázi cyklu vývoje se CASE nástroj využije:

- Pre CASE - globální strategie vývoje IS
- Upper CASE - podporuje tvorbu informační strategie a fázi analýzy
- Middle CASE - podporuje tvorbu globálního a detailního návrhu IS
- Lower CASE - podporuje fázi implementace
- Post CASE - podporuje fázi uvedení IS do provozu, provoz, údržbu, reengineering

Mezi hlavní výhody využití CASE nástrojů patří především produktivita práce, menší chybovost a další. Jako hlavní představitelé CASE nástrojů můžeme uvést např.: Powerdesigner (Sybase), Oracle Designer (Oracle), Case Studio a mnoho dalších.

K této kapitole byly informace čerpány ze zdroje [1].

## UML (Unified Modeling Language)

UML je grafický jazyk. Využívá se pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. UML podporuje objektově orientovaný přístup k analýze, návrhu a popisu programových systémů. UML neobsahuje způsob, jak se má používat, ani neobsahuje metodiky, jak analyzovat, specifikovat či navrhovat programové systémy. Standard UML definuje standardizační skupina Object Management Group (OMG).

### Součásti UML

V současné době je aktuální standard ve verzi 2.0, který se skládá ze čtyř částí:

- **UML 2.0 SuperStructure** – popis UML z hlediska uživatele (analytik/programátor). Tato část popisuje jednotlivé diagramy.
- **UML 2.0 Infrastructure** – metamodel stojící v pozadí za UML, specifikovaný pomocí Meta-Object Facility (MOF).
- **UML 2.0 Object Constraint Language (OCL)** – jazyk pro specifikaci vstupních a výstupních podmínek, invariantů v jednotlivých diagramech.
- **UML 2.0 Diagram Interchange** – popis XML struktur pro výměnu konkrétních modelů mezi jednotlivými modelovacími nástroji.

Nejvyužívanějšími ze standardu UML jsou diagramy, které se člení na:

- **strukturní diagramy:**

- diagram tříd
- diagram komponent
- composite structure diagram
- diagram nasazení
- diagram balíčků
- diagram objektů, též se nazývá diagram instancí

- **diagramy chování:**

- diagram aktivit
- diagram užití
- stavový diagram
- **diagramy interakce:**
  - sekvenční diagram
  - diagram komunikace
  - interaction overview diagram
  - diagram časování

K této kapitole byly informace čerpány ze zdroje [1].

## Rational Rose

Rational Rose je rodina několika produktů od firmy IBM, která využívá klasického UML modelování. Hned ze začátku bych rád uvedl, že se mi nepodařilo stáhnout ani jeden produkt, ačkoliv jsem se o to pokoušel. Tudíž informace, které zde uvedu u jednotlivých produktů, jsou informace získané z webu firmy IBM. Nicméně dále se zmíním o jednom produktu ze tří, které firma IBM uvádí na svém webu jakožto novější produkty oproti rodině Rational Rose, které by měly tuto rodinu plně nahradit. Jedná se o IBM Rational Software Architect, IBM Rational Software Modeler a IBM Rational Data Architect. Podařilo se mi stáhnout trial verze prvních dvou, uvedu zde IBM Rational Software Architect.

Původně se jednalo o samostatnou firmu, ale jak jsem již zmínil výše, dnes funguje pod záštitou IBM. Jedná se tedy o CASE nástroje, které mají poskytovat vývojářům přehledné, snadné a rychlé grafické modelovací prostředí, ať se jedná o use-case diagramy, scénáře, diagramy spolupráce, diagramy tříd, stavové diagramy, diagramy komponent či další komponenty.

### Rational Rose Data Modeler

IBM Rational Rose Data Modeler urychluje návrh databáze v zavedeném modelovacím prostředí.

- Urychluje návrh databáze tím, že poskytuje:
  - o Propracované prostředí pro modelování.
  - o Flexibilní transformaci mezi logickými a fyzickými modely a pokročilé možnosti vizuálního modelování databází.
- Umožňuje návrhářům databází sledovat, jak daná aplikace řeší přístup do databáze a tím umožňuje odhalit potenciální problémy ještě před implementací.
- Spojuje návrháře databází používající ER modelování se zbytkem týmu vyvíjejícího objektivě orientované aplikace prostřednictvím jednoho společného nástroje a jedné společné notace - jazyka UML.

### Operační systémy a příslušné hardwarové platformy

- Windows 2000
- Windows NT
- Windows XP

### Rational Rose Developer for Java

Produkt Rational Rose Developer for Java je plně vizuální modelovací prostředí založené na jazyku UML. Obsahuje podporu generování kódu pro modely Java a J2EE. Jako všechny produkty Rational Rose, poskytuje i tato verze celému týmu společný modelovací jazyk, čímž umožňuje rychlejší vytváření kvalitního softwaru.

Další vlastnosti:

- Podpora analýzy a návrhových vzorů pro Rose J založených na publikaci "Design Patterns: Elements of Reusable Object-Oriented Software" (český překlad Grada: Návrh programů pomocí vzorů).
- Generování kódu Java a CORBA, zpětná vizualizace do modelu a konfigurovatelná synchronizace model - kód.

- Podpora JDK 1.5 a další možnosti (Enumerations, For-Each Loop, Auto-Boxing/UnBoxing, Generics a VarArgs, statické importy apod.)
- Integrace s ostatními vývojářskými nástroji IBM Rational pomáhající podporovat potřeby členů týmu i v ostatních fázích životního cyklu projektu.
- Integrace s hlavními IDE pro vývoj v jazyku Java.
- Veškeré výstupy lze přehledně publikovat pro zjednodušení komunikace mezi členy týmu.

## Rational Rose Developer for UNIX

- Vizualní modelování a vývoj v jazyce UML.
- Generování kostry kódu z modelu a jeho zpětná vizualizace pro prostředí Java, C++ a Ada.
- Založeno na IBM Rational Rose pro UNIX/Linux.

## Operační systémy a příslušné hardwarové platformy

- HP-UX
- Linux
- SUN Solaris

## Rational Rose Developer for Visual Studio

Rational Rose Developer for Visual Studio, plně vizualní modelovací prostředí založené na jazyku UML, obsahuje podporu generování kódu pro jazyky Visual C++ a Visual Basic 6.

Další vlastnosti:

- Analýza, podpora návrhových vzorů ANSI C++, Visual C++ založená na publikaci "Design Patterns: Elements of Reusable Object-Oriented Software" (český překlad Grada: Návrh programů pomocí vzorů).
- Modelování, generování kostry kódu a jeho zpětná vizualizace, podpora jazyků C++, ANSI C++, CORBA, Visual C++ a Visual Basic s konfigurovatelnou synchronizací model - kód.
- Integrace s Visual Studio 6 (pro Visual Basic 6) a integrace s Visual Studio 2003 a Visual Studio 2005.
- UML Modelování při návrhu databází prostřednictvím Rational Rose C++ se schopností reprezentovat integraci dat a aplikační požadavky prostřednictvím logických a fyzických návrhů.
- Schopnost integrace se systémem pro správu verzí vyhovujícím SCC, IBM Rational ClearCase.
- Integrace s ostatními vývojářskými nástroji IBM Rational pomáhající podporovat potřeby členů týmu i v ostatních fázích životního cyklu projektu.
- Veškeré výstupy lze přehledně publikovat pro zjednodušení komunikace mezi členy týmu.

## Rational Rose Modeler

Rational Rose Modeler je vhodný pro architektonické návrhy; obsahuje plnou podporu jazyka UML pro vytváření softwarových architektur tak, abyste mohli využívat a maximalizovat týmové dovednosti navrhování, efektivně a pružně využívat a maximalizovat týmové dovednosti, zjednodušit a urychlit všechny aktivity v průběhu návrhu a zajistit tak nezbytnou kvalitu produktu co nejdříve například využitím standardních návrhových vzorů.



- Podporu návrhových vzorů např. dle publikace: "Design Patterns: Elements of Reusable Object-Oriented Software" (český překlad Grada: Návrh programů pomocí vzorů).
- Integrace s ostatními vývojářskými nástroji IBM Rational pomáhající podporovat potřeby členů týmu i v ostatních fázích životního cyklu projektu.
- Schopnost integrace se systémem pro správu verzí vyhovujícím SCC, IBM Rational ClearCase.
- Veškeré výstupy lze přehledně publikovat pro zjednodušení komunikace mezi členy týmu.

## Rational Rose Technical Developer

Řešení vývoje řízeného modelem automatizující překlad návrhu do kódu pro jazyky Java, C a C++.

- Nejrobustnější řešení vývoje řízeného modelem.
- Plně automatizovaný překlad návrhu do kódu pro jazyky Java, C a C++.
- Spouštění modelu za běhu, generování plně spustitelného kódu a vizuální ladění.
- Automaticky sestavuje ovladače, stuby, simulační programy a testovací skripty.
- Optimalizováno pro aplikace řízené událostmi, souběžné i distribuované aplikace.
- Pokročilé modelování splňuje přísné požadavky na latenci, propustnost a spolehlivost.
- Navrženo pro technologicky nejnáročnější aplikace.
- Obsahuje IBM Rational Rose RealTime (Windows, UNIX, Linux) pro plnohodnotný vývoj řízený modelem v jazycích Java, C a C++.
- Obsahuje IBM Rational Rose for UNIX/Linux a IBM Rational Rose Enterprise for Windows pro integraci na úrovni návrhu s jazyky Java, C++ a Ada.

## Operační systémy a příslušné hardwarové platformy

- Linux
- SUN Solaris
- Windows 2000
- Windows NT
- Windows XP

Bohužel se mi nepodařilo sehnat žádné obrázky prostředí jednotlivých produktů, takže se musíme spokojit pouze s technickými údaji od firmy IBM a rozvinout svou fantazii. I když věřím, že IBM se snaží vyvíjet své produkty co možná nejvíce user-friendly a proto věřím, že vývojové prostředí bude dosti obdobné jako u následujícího produktu.

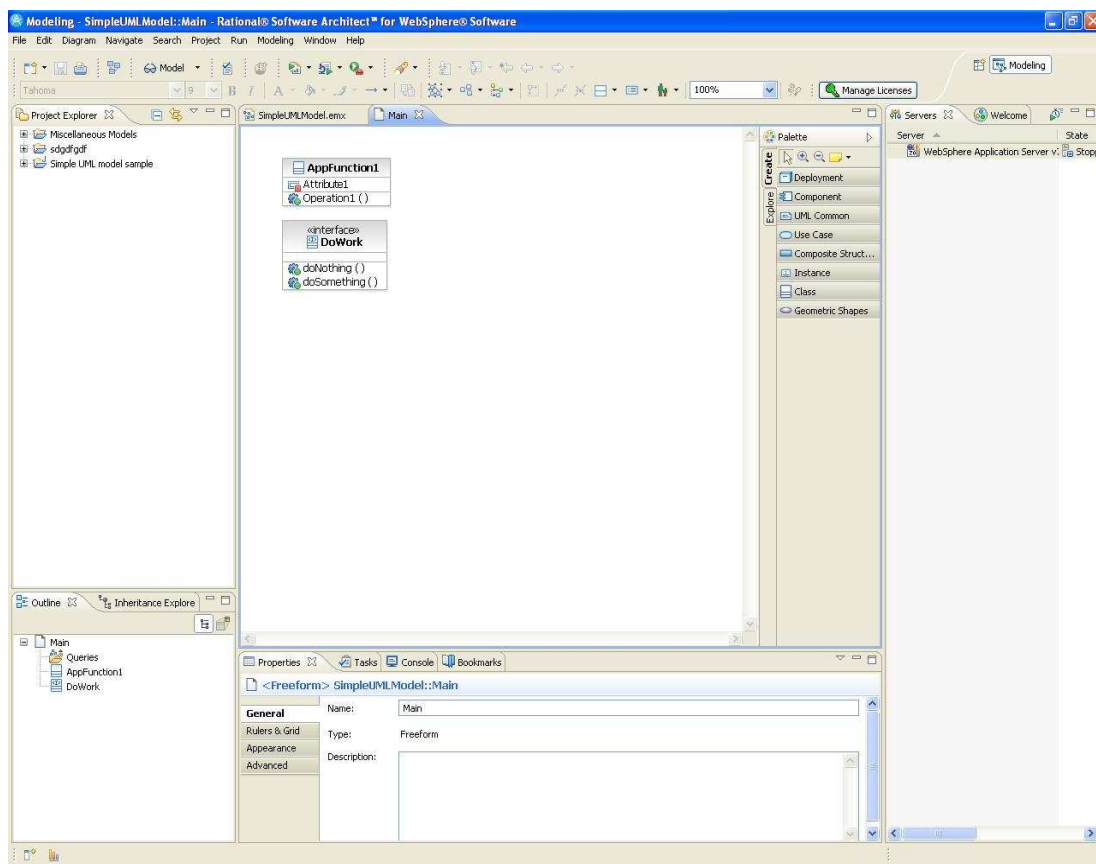
K této kapitole byly informace čerpány ze zdroje [2].

# Rational Software Architect for WebSphere Software

Tento produkt se mi povedlo stáhnout v trial verzi. Jedná se o nástroj, který podporuje komplexně modelování pro vývoj softwarových architektur. Skládá se z mnoha modulů, čehož si můžete všimnout na obrázku níže. Práce s ním je pro nás uživatele relativně rychle pochopitelná a naučitelná, už jen pro podobnost vůči veškerému softwaru, s kterým se běžně setkáváme. Vzhled tohoto produktu je user friendly a přímo nám napovídá, kde co najít. Navíc je jeho prostředí uspořádáno velmi přehledně. Mě osobně nejvíce zaujala paleta, ve které nalezneme veškeré objekty, které bychom mohli využít.

Samozřejmě, že abychom byli schopni navrhovat modely a pracovat s nimi, byla by určitě nutná detailní analýza všech možností, které tento produkt nabízí, ale to není cílem této práce, jejím cílem je seznámit s produkty, které umožňují modelování pomocí jazyka UML. Technické údaje jsou uvedeny pod obrázkem, z webu firmy IBM.

## Ukázka rozhraní



Obrázek 1 - Ukázka rozhraní, zdroj: [www.ibm.com](http://www.ibm.com)

## Přehled funkcí

- modelování a grafická editace v rámci nejrůznějších oblastí včetně jazyka UML 2, Java, XSD, webu, technické infrastruktury atd.
- Komplexní podpora nových i nejnovějších programovacích modelů: Web2.0, J2EE 5, EJB3.0, JPA.

- Zahrnuje produkt IBM Rational Application Developer, a umožňuje tak návrhářům a vývojářům práci v plně integrovaném vývojovém prostředí.
- Nástroje pro analýzu architektury umožňují přezkoumávání a správu struktury vašich aplikací Java a aplikací orientovaných na služby.
- Flexibilní správa modelů podporuje paralelní vývoj a refaktoring architektury, který umožňuje rozdělování, kombinaci, porovnávání a slučování modelů a jejich fragmentů.
- Platforma spolupráce, která nabízí hladkou integraci s platformou poskytování softwaru IBM Rational, včetně řešení IBM Rational Team Concert, IBM Rational Asset Manager, IBM Rational Requirements Management a Change Management.
- Otevřená s širokými možnostmi rozšiřování. Založena na platformě Eclipse 3.4
- Podporované operační systémy: Windows, Linux

K této kapitole byly informace čerpány ze zdroje [2].

## Violet UML Editor

# Violet UML Editor



Aktuální verze: 0.21.1

Výrobce: Cay S. Horstmann a Alexandre de Pellegrin

Licence: zdarma

Violet UML Editor je velmi jednoduchý nástroj pro tvorbu UML diagramů. Neobsahuje prakticky jiné funkce než jejich kreslení, cokoli pokročilejšího (generování kódu,...) v něm není obsaženo. Tento „nedostatek“ je ovšem nedostatek záměrný, neboť hlavním účelem tohoto programu není komplexní dokumentační obsluha složitých projektů, ale naopak záměr posloužit k rychlému kreslení a prezentaci diagramů s důrazem na snadnou použitelnost a pochopitelnost.

Autor sám upozorňuje, že jeho program nejlépe poslouží studentům, učitelům a občasným „uživatelům“ UML diagramů. Naopak komukoli jinému hledajícímu nástroj pro „vážnější“ práci s diagramy doporučuji profesionální nástroje, z nichž mnohé jsou zmíněny v ostatních kapitolách této práce. Pokud nechcete používat placený software, nechcete dlouho čekat na spuštění rozsáhlých programů a nechcete se komplikovaně prokousávat uživatelským rozhraním obrazovek přetékajících funkcemi, tlačítky a ikonami, pak je pro Vás Violet UML Editor vhodným nástrojem.

Je zřejmé, že například pro výuku základů UML či objektových návrhů bude vhodnější studentům na projektoru prezentovat kreslení v jednoduché aplikaci než komplikované rozhraní například PowerDesigneru. Stejně tak je zřejmé, že tato aplikace neposlouží analytikovi, který chce svůj objektový diagram databázové struktury přegenerovat do SQL deklarácí. K tomu Violet UML Editor opravdu určen není.

### Přehled funkcčnosti

- Podpora pro kreslení diagramů:
  - o diagram případů užití
  - o diagram tříd
  - o diagram aktivit
  - o sekvenční diagram
  - o stavový diagram
  - o objektový diagram
- propojení s nástrojem Eclipse
- nezávislost na platformě (vyžaduje pouze přítomnost Java Runtime Environment 6.0)
- rozmanitost možností spouštění
  - o samostatná aplikace
  - o pomocí Java Web Start – aplikace běží na serveru a je k ní přístupováno přes prohlížeč
  - o jako java applet v prohlížeči

- o jako Eclipse plugin
- export diagramu do PNG obrázku či schránky

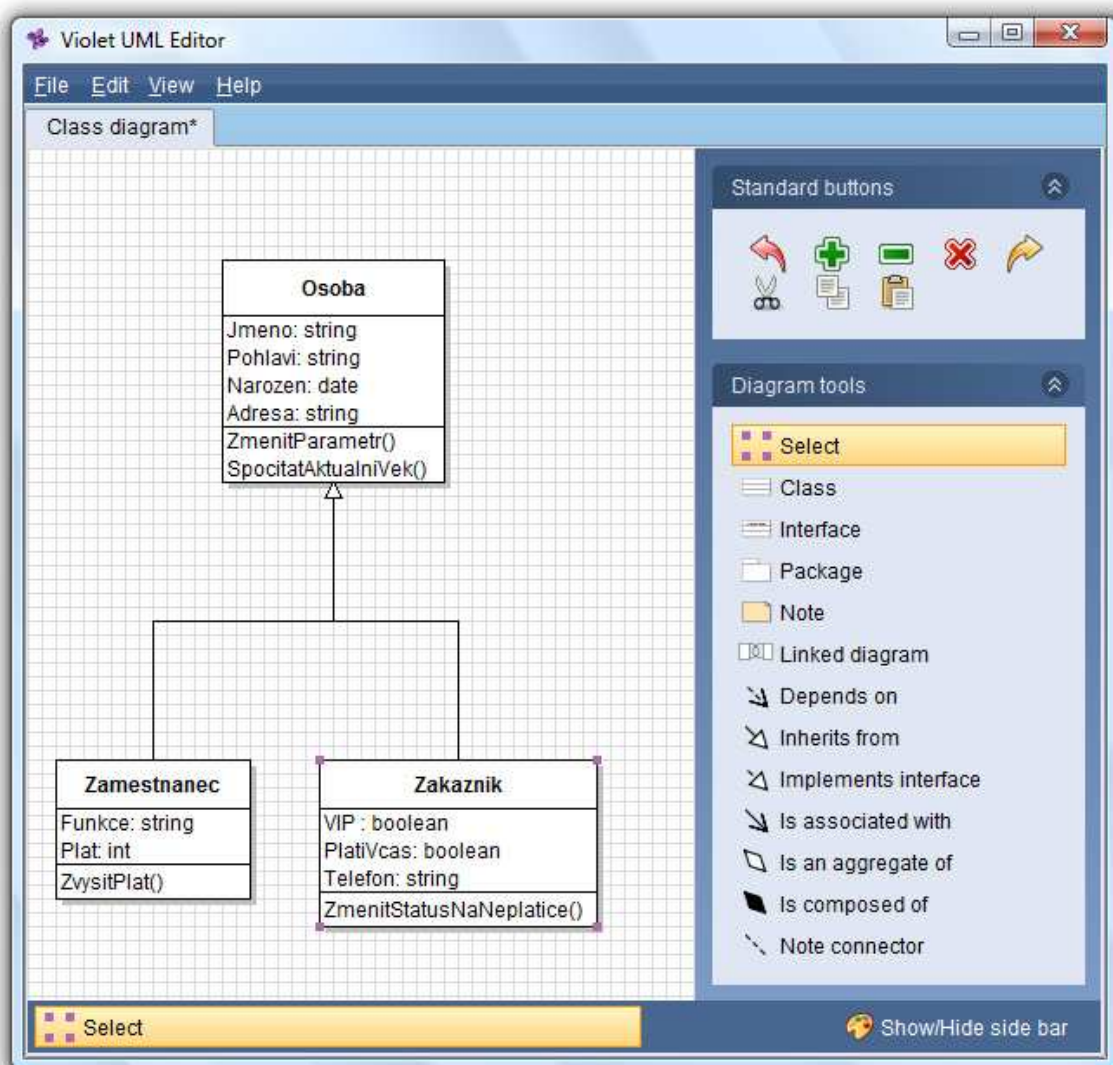
## Co Violet neumožňuje

- generování zdrojového kódu z diagramů
- reverzní inženýrství (generování UML diagramů podle zdrojového kódu)
- kontrolu modelů (můžete tak tvořit záměrně nekonzistentní modely)
- XML export / import (výměnu modelů s jinými UML nástroji)

Těmito funkcemi profesionálnější nástroje samozřejmě disponují, ale Violet UML Editor chce sloužit především k jiným účelům, k již zmíněné výuce a především být nástrojem pro rychlé kreslení diagramů. Tomu je přizpůsobeno jejich tvoření, grafické rozhraní a intuitivní ovládání.

## Ukázka rozhraní

Jak znázorňuje následující obrazovka, rozhraní programu je opravdu jednoduché a maximálně přizpůsobené rychlé práci s diagramem.



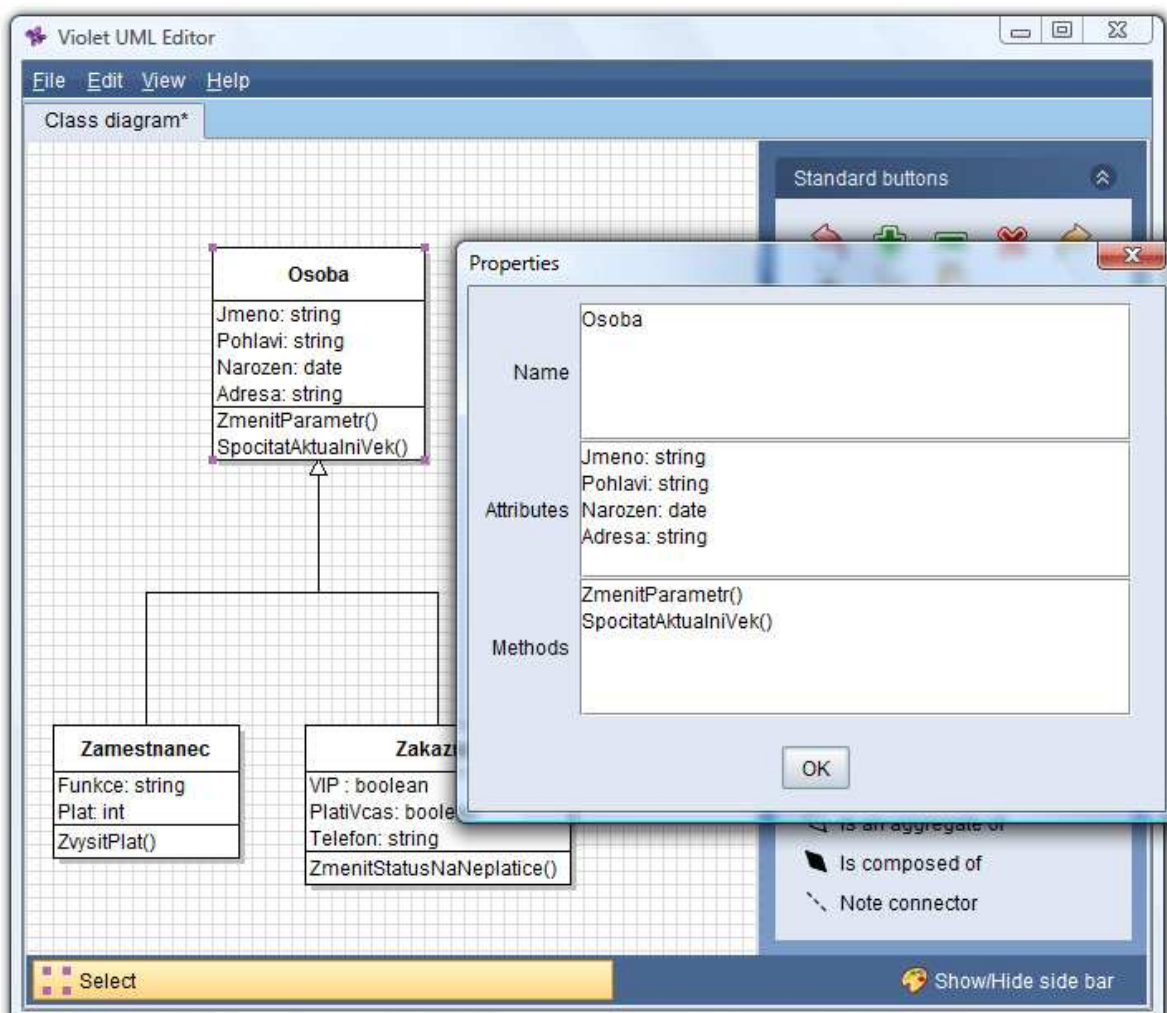
Obrázek 2 - Ukázka tvoření diagramů

Pravá horní nabídka obsahuje nejběžnější „editační funkce“ – undo, redo, copy&paste, vyjmutí, smazání. Ta zůstává neměnná pro všechny diagramy, se kterými může tato aplikace pracovat.

Druhá nabídka je naopak proměnlivá – podle toho, se kterým typem diagramu se pracuje. Jiná bude pro Class diagram a jiná pro Use-case diagram.

Velmi zajímavou vlastností je vkládání objektů do diagramu (ve smyslu jeho „stavebních částí“). Při prvním použití se bude uživatel určitě snažit přetahovat objekty z pravé nabídky na kreslicí plochu, ovšem tyto pokusy budou neúspěšné. Objekty se do diagramu vkládají prostým klepnutím na kreslicí plochu, kdy je vložen objekt označený v pravé nabídce. Neefektivní? Naopak! Výběr v pravé nabídce lze totiž provádět prostým scrollováním kolečka na myši! Mohu tak jednoduše pootočením kolečka vybrat objekt a ihned jej kliknutím vložit na požadované místo v diagramu. Efektivní, rychlé, jednoduché.

Obdobně jednoduchá je i práce s objekty – zadávání názvů, atributů, metod. Více na následujícím obrázku:



Obrázek 3 - Ukázka zadávání atributů a metod objektů

Název, atributy i metody objektu se manuálně vpisují do jednoduchých textových polí. Pokud nepotřebujete vytvořit reálný a konzistentní diagram, ale naopak oceníte možnost vepsat do atributů

nebo metod objektu cokoli potřebujete, pak je tento způsob práce s „obsahem“ objektů ideální. Nemusíte myší naklikávat atributy, vybírat jejich „datové typy“, zadávat návratové hodnoty metod. Pouze si snadno a rychle napíšete, co potřebujete zobrazit v diagramu.

## Shrnutí

Nástroj Violet UML Editor je jednoduchá aplikace vhodná pro výukové účely nebo rychlé vytvoření diagramu, kdy není kladen důraz na jeho konzistenčnost nebo pokročilejší funkce jako generování zdrojových kódů apod. Stačí pár minut a uživatel je schopen velmi efektivně modelovat základní UML diagramy. Pro cokoli jiného je nutné poohlédnout se po propracovanějších nástrojích.

K této kapitole byly informace čerpány ze zdroje [3].

# Borland Together 2008



Aktuální verze: Borland Together 2008 SP1 Worldwide

Výrobce: Borland Software Corporation

Licence: komerční

Borland Together je robustní nástroj pro analýzu, návrh a vývoj podnikových aplikací. V jedné aplikaci jsou tak obsaženy mechanismy potřebné pro všechny role podílející se na vývoji řešení: podnikové analytiky, systémové analytiky, architektky a vývojáře společně s nástroji pro jejich komunikaci, kolaboraci na projektu a jeho řízení.

## Podpora standardů

V produktu Borland Together jsou využity následující průmyslové standardy:

- Unified Modeling Language (UML)
- XML Metadata Interchange (XMI)
- Query / Views / Transformations (QVT)
- Object Constraint Language (OCL)

Značná automatizace různých procesů přispívá k vyšší efektivnosti pracovních týmů. Lze zmínit automatickou tvorbu dokumentace, snadné znovupoužití dříve vytvořených vzorů a komponent, rychlé zanesení prováděných změn do modelů díky refaktoringu a v neposlední řadě unikátní technologii LiveSource hlídající kompatibilitu modelu a zdrojového kódu z něj vytvořeného (a naopak).

## Klíčové vlastnosti produktu

### Domain-specific language (DSL)

Novinkou v tomto nástroji je podpora pro práci s DSL, což je zkratka pro „domain-specific language“. Zjednodušeně řečeno, UML nám poskytuje možnosti jak namodelovat „cokoli“, ale díky tomu jsou návrhy velmi obecné. Pokud se pochybujeme v rovině „podnikání“, pak už můžeme potřebovat určitá, pro tuto oblast specifická, omezení. A k tomu účelu slouží DSL, který vstupuje do práce s diagramy, upravuje transformaci modelů do zdrojových kódů apod. Je to jazyk pro zachycení specifik, která nemohou být zachyceny v UML diagramech.

### Business Process modelování

Podpora Business Process Modeling Notation (BPMN) včetně validačních pravidel. Taktéž umožňuje import a export specifikací pro interakci s webovými službami, tedy BPEL4WS.

### UML modelování

Modelování za pomoci diagramů UML 1.4 a UML 2.0. Velmi přínosnou je technologie LiveSource, která dokáže synchronizovat změny mezi modely a zdrojovými kódy v jazycích Java, C++ a COBRA IDL. Z toho vyplývá podpora transformací modelů do zdrojových kódů více jazyků. Není problémem taktéž modely slučovat či rozdělovat a tyto postupy spolehlivě kontrolovat.



## Data modelování

Borland Together dokáže generovat modely jak ze zdrojových kódů vybraných programovacích jazyků, tak z vybraných databázových rozhraní. Neméně důležité je samozřejmě generování logické a fyzické struktury databáze z navržených modelů a diagramů. Podporované databázové platformy: Oracle, DB2, Sybase, MS SQL Server.

## Pokročilé modelování

Podpora Object Constraint Language pro podrobnější specifikaci objektů, podpora pro transformace mezi jednotlivými modely (QVT), generování zdrojových kódů v jazyce Java, Java2EE, C++ a C#, rozpoznávání návrhových vzorů ve zdrojových kódech, import a export XMI 2.0 modelů,...

## Ukázka rozhraní

Následující ukázka uživatelského rozhraní zachycuje práci s nástrojem a generování zdrojového kódu v jazyce Java pomocí technologie LiveSource.

The screenshot displays the Borland Together IDE interface. The top window, titled "Modeling - course management classes - Together", shows a UML class diagram. The diagram includes classes: `Course` (with attributes `isSuperseded`, `supersedeCourse`, `isWithdrawn`, `withdrawCourse`), `CourseWithdrawal` (with attributes `availableFrom`, `code`, `description`, `minimumStudents`, `duration`, `keywords`, `name`, `prerequisites`, `courseInterests`, `maximumStudents`), `Prerequisite` (with attributes `name`, `id`, `description`, `satisfyingCourses`), and `CourseWithdrawal` (with attributes `name`, `id`, `description`, `satisfyingCourses`). Relationships include a 1-to-0..\* association between `Course` and `Prerequisite`, a 1-to-0..\* association between `Course` and `CourseWithdrawal`, and a 1-to-0..1 association between `CourseWithdrawal` and `CourseWithdrawal`. The bottom window, titled "Course.java", shows the following Java code:

```
/**
 * returns true if course has been withdrawn
 *
 * @return true if withdrawn
 */
public boolean isWithdrawn() {
    return courseWithdrawal != null;
}
```

Obrázek 4 - Ukázka rozhraní produktu Borland Together

## Shrnutí

Borland Together 2008 je mocný nástroj (nejen) pro UML modelování. Obsahuje spousty dalších funkcí a podporuje technologie, bez kterých nelze na modelování či vývoj komplexních systémů ani pomyslet, nemluvě o podpoře týmové práce, automatizaci rutinních činností, důmyslné „spolupráci“ se zdrojovými kódy a mnoha dalšími vymoženostmi, které tento produkt zaslouženě řadí ke špičce komerčních modelovacích nástrojů.

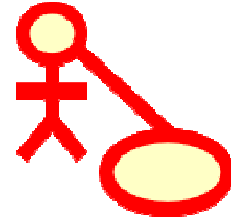
K této kapitole byly informace čerpány ze zdroje [4].

# Umbrello UML Modeller

Aktuální verze: Umbrello 2.0

Výrobce: Open-source komunita

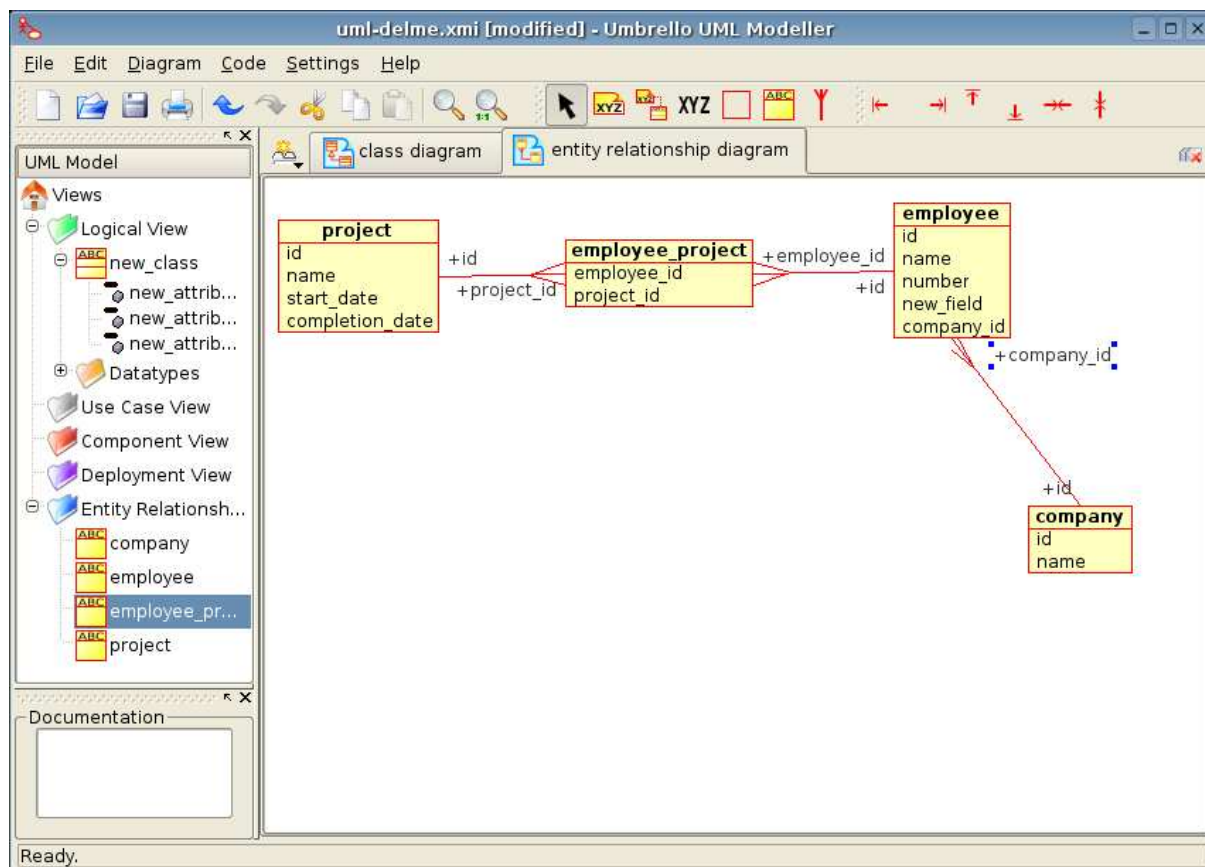
Licence: GPL



Jedná se taktéž o CASE nástroj určený k podpoře modelování UML diagramů, a to s podporou většiny prvků UML 2.0. Je určen primárně pro Unixové operační systémy, existuje však i verze pro MS Windows. Umbrello UML Modeler je vyvíjen jako součást Linuxového prostředí KDE, jehož standardní součástí se stal od verze KDE 2.0 (poslední verze Umbrello UML Modeller 2.0 součástí KDE 4.0). Jeho šíření podporuje fakt, že je šířen pod GPL licenci, což umožňuje jeho libovolné šíření za libovolnou cenu, tedy i zdarma. Je naprogramován v jazyce C++, což představuje jednu jeho výhodu oproti javovským nástrojům. Doba odezvy je totiž o něco kratší. Podstatnějším kladem ale je podpora širokého spektra programovacích jazyků. Jako příklady zmiňme PHP 5, Python, Perl atd. Umožňuje generování kódu z UML (primárně do C++ a Javy) a naopak (pouze z C++).

Podporované typy UML diagramů:

- Use Case Diagramy
- Diagramy tříd
- Sekvenční Diagramy
- Diagramy spolupráce
- Stavové diagramy
- Schémata činností
- Diagramy komponent
- Diagramy nasazení



Obrázek 5 - Rozhraní Umbrello UML Modelleru , zdroj: [www.abclinuxu.cz](http://www.abclinuxu.cz)

Domovská stránka Umbrello UML Modelleru: <http://uml.sourceforge.net/index.php>

Stránka pro stažení Umbrello UML Modelleru: <http://uml.sourceforge.net/download.php>

Manuál pro práci s Umbrello UML Modellerem: [http://docs.kde.org/stable/en\\_GB/kdesdk/umbrello/](http://docs.kde.org/stable/en_GB/kdesdk/umbrello/)

## Shrnutí

Umbrello UML Modeler je jednoduchý nástroj, díky čemuž je práce s ním poměrně snadná. Negativní stránkou však je fakt, že nedisponuje některými pokročilými funkcemi využívanými především zkušenějšími UML modeláři. Současně také nepodporuje týmovou práci ani sdílení dat.

K této kapitole byly informace čerpány ze zdroje [5].

# ArgoUML

Aktuální verze: ArgoUml 0.28

Výrobce: Open-source komunita

Licence: BSD

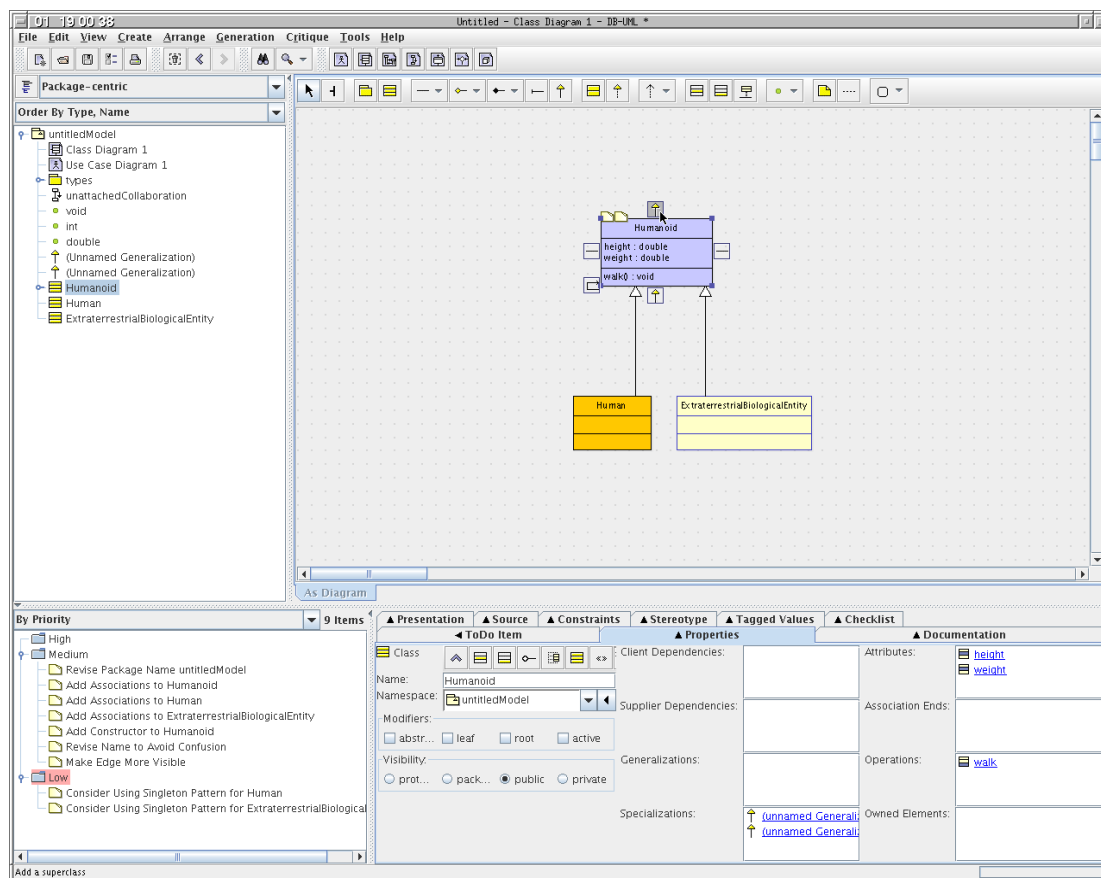


ArgoUML je šířen pod licencí BSD (typ open source). Poslední verze 0.28 byla uvolněna 23.3. 2009. Je napsaný v Javě, což podporuje jeho přenositelnost mezi platformami s JRE. Umožňuje tvorbu řady UML diagramů (viz níže) a také databázových schémat.

Podporované typy UML diagramů:

- Use Case diagramy
- Diagramy tříd
- Stavové diagramy
- Sekvenční diagramy
- Diagramy spolupráce
- Diagramy aktivit
- Diagramy nasazení

## Ukázka rozhraní



Obrázek 6 - Rozhraní Argo UML, zdroj: [www.abclinuxu.cz](http://www.abclinuxu.cz)

Vytvořené diagramy a schémata je možné ukládat ve formátu XMI s možností exportu do GIF, PNG, PostScript, EPS a SVG.

Z hotových class diagramů je možné také vygenerovat zdrojový kód v jazyce Java a po přidání patřičných pluginů i v dalších programovacích jazycích včetně C++ (dále pak C#, PHP4/5, Python a další).

Domovská stránka Argo UML: <http://argouml.tigris.org/>

Stránka pro stažení ArgoUML: <http://argouml-downloads.tigris.org/>

Manuál pro práci s ArgoUml: <http://argouml.tigris.org/documentation/defaulthtml/manual/>

## **Shrnutí**

Poněkud stinnější stránkou tohoto nástroje je poněkud delší doba odezvy ve srovnání s nástroji vytvořenými v jiných programovacích jazycích. Zajímavou schopností je možnost zpuštění přímo z domovské stránky bez nutnosti instalace.

K této kapitole byly informace čerpány ze zdroje [6].

## Open ModelSphere



Aktuální verze: Open ModelSphere 3.0

Výrobce: Open-source komunita

Licence: GPL

Jak již ze jména vyplývá, jedná se o zástupce volně šiřitelných nástrojů. Tento program vytvořili na počátku 90. let minulého století profesori a studenti na Laval University v Quebecu v Kanadě. Ve své době to byl jeden z prvních grafických CASE nástrojů pro business procesy a data modeling. Program byl dále vyvíjen firmou Grandite a prodáván jako komerční produkt a v roce 2008 byl program uvolněn pod GPL licenci. V současnosti je aktuální verze 3.0.

S přihlédnutím k tomu, že je tento program k dispozici zdarma, tak jeho možnosti jsou poměrně široké. V případě UML sice nepodporuje všechny diagramy verze 2.0, ale pouze následující:

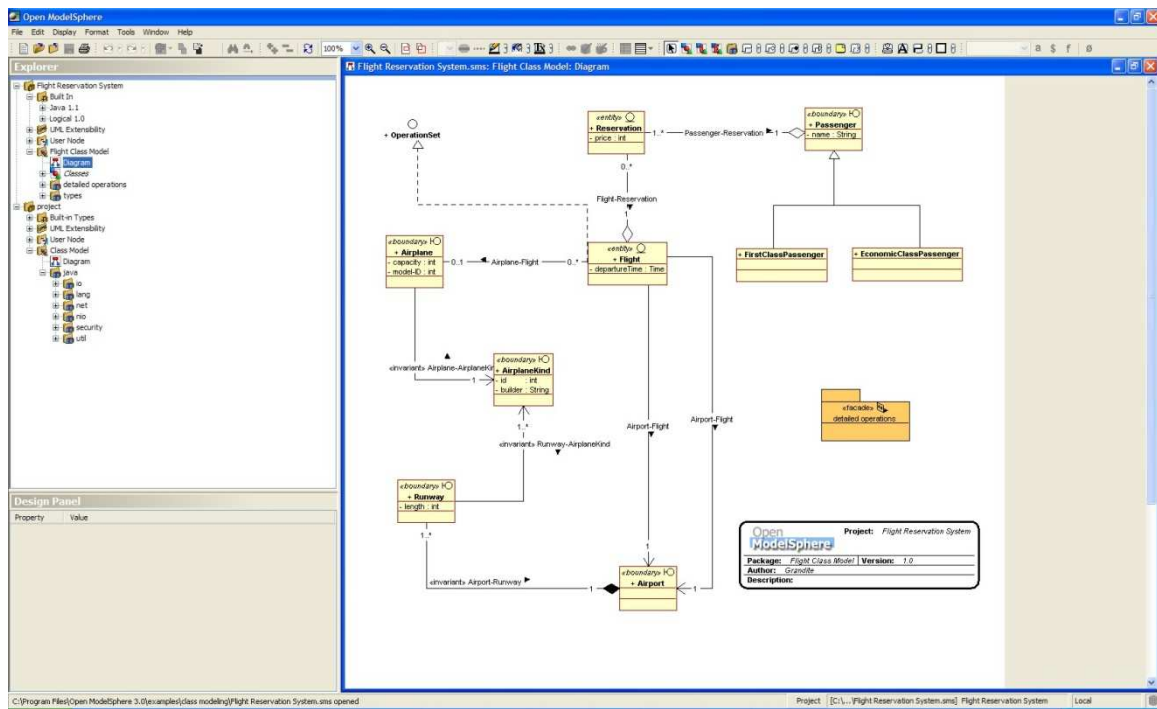
- diagram tříd
- diagram užití
- diagram aktivit
- diagram nasazení
- diagram balíčků
- diagram komponent
- diagram komunikace
- sekvenční diagram
- stavový diagram

Open ModelSphere může být velmi přínosný při navrhování Java aplikací. Umožňuje totiž jak generování zdrojového kódu aplikace z vytvořeného diagramu tříd, tak také umožňuje vytvořit tento diagram z již existujícího kódu. Dále také umí zkontrolovat validitu kódu.

Další oblastí kde lze program využít je data modelování. Zobrazí vizualizaci databázové architektury, kterou pak lze jednoduše upravit. Také samozřejmě generuje SQL skripty z modelu, které umožňují nejen tvorbu nové databáze, ale zároveň i synchronizaci stávající v případě její úpravy.

Open ModelSphere obsahuje nástroje pro tvorbu business process a data flow diagramů, které podporují grafické zobrazení v několika notacích.

## Ukázka rozhraní



Domovská stránka: <http://www.modelsphere.org/>

## Shrnutí

Open ModelSphere je nekomerční program, který umí přesně to, co by od něj analytik očekával, ale o mnoho více ne. Přesto obsahuje i některé funkce, které nejsou u open-source produktů v tomto oboru příliš obvyklé, například generování modelů ze zdrojových kódů Javy a naopak nebo validátor tohoto kódu.

K této kapitole byly informace čerpány ze zdroje [7].



# Visual Paradigm for UML Enterprise Edition



Aktuální verze: Visual Paradigm Suite 7.0

Výrobce: Visual Paradigm International

Licence: komerční

Visual Paradigm for UML vyrábí firma Visual Paradigm International. Lze ho zakoupit samostatně nebo jako součást celého balíčku nástrojů Visual Paradigm Suite. Samotný program v nejnižší verzi pro nekomerční použití stojí 70 \$, v nejvyšší verzi Enterprise Edition, kterou zde popisují, pak téměř 1680 \$. Nejnovější verze z dubna letošního roku je 7.0.

Z názvu by se mohlo zdát, že Visual Paradigm for UML je CASE nástroj opravdu jen pro tvorbu UML diagramů, zvláště když v portfoliu firmy jsou další specializované nástroje například na modelování business procesů, data modelování. Opak však je pravdou, Visual Paradigm for UML obsahuje i tyto nástroje, je to opravdu komplexní program.

Jelikož je tento software přímo určen pro UML tak samozřejmě nabízí plnou podporu UML ve verzi 2.1. To znamená, že lze vytvářet všech třináct UML diagramů.

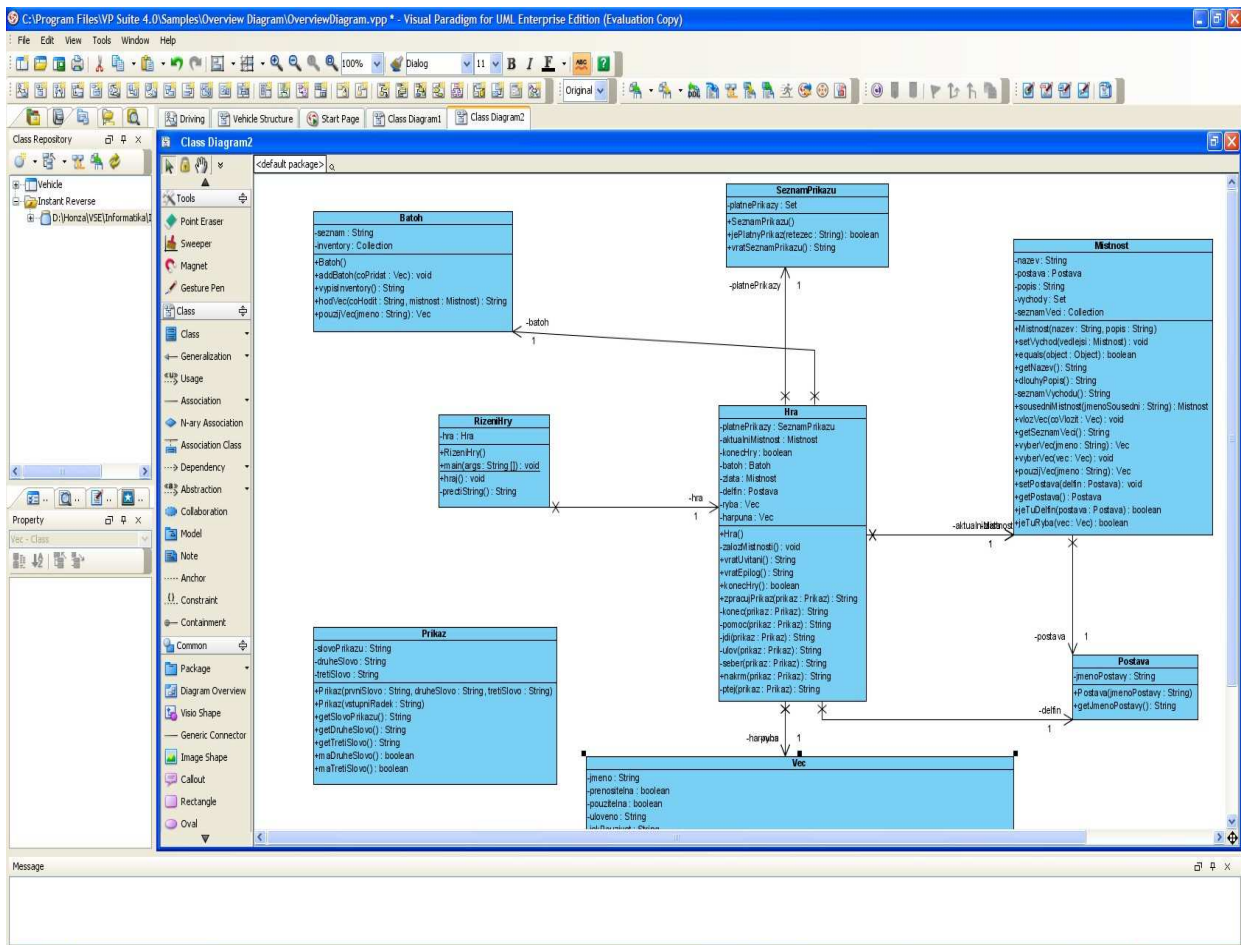
Jak již sem naznačil, program nabízí také spoustu možností pro business process modelování. Kromě samotného business process diagramu (pro který používá BPMN notaci) obsahuje data flow diagram, diagram procesní mapy a organizační struktury.

U takového programu nemůže chybět podpora několika programovacích jazyků, zde jich nalezneme více než deset, například Java, C++, PHP, C# . Samozřejmostí je obousměrný proces, generování kódu z diagramu a naopak převod z kódu do diagramu.

Mezi další funkce patří možnost týmové spolupráce pomocí verzování, modelování databází, dále pak jednoduchá integrace programu do různých vývojových prostředí jako jsou Visio, Eclipse či NetBeans.

Uživatelské rozhraní programu je velice intuitivní a snadno ovladatelné. Pokud by však méně zkušený uživatel měl přece jen problémy s jeho ovládním, tak lze vřele doporučit výborně zpracované internetové stránky programu, kde se nacházejí video tutoriály k velké většině funkcí programu.

## Ukázka rozhraní



Domovská stránka: <http://www.visual-paradigm.com>

## Shrnutí

Visual Paradigm for UML je profesionální program pro práci s UML diagramy a obsahuje všechny funkce potřebné v tomto oboru. Zajímavostí je schopnost spolupráce s více než desítkami programovacími jazyky (respektive zdrojovými kódy v těchto jazycích) a výborně zpracovaná nápověda s tutoriály na webu produktu.

K této kapitole byly informace čerpány ze zdroje [8].

## Kritéria výběru vhodného řešení

Jistě každého napadne, že výběr vhodného řešení je velmi subjektivní záležitost. Nicméně jako základní kritérium bychom mohli považovat rozdělení CASE nástrojů pro jazyk UML dle licencí, ve kterých jsou poskytovány. Tudiž prvním kritériem jsou free a komerční licence. Z předešlého textu, kde jsme zpracovávali několik nástrojů je očividné, že komerční nástroje sice něco stojí, ale na druhou stranu mají mnohem více funkcí, které by se nám mohli hodit.

Z předešlého vyplývá další kritérium a to, funkcionality. Je nutné, abychom zvážili, za jakým účelem chceme CASE nástroj pořídit, jaké funkce vyžadujeme, aby nástroj uměl zpracovat. Pokud budeme chtít nástroj například jen za účelem prezentace UML jazyka, pak by nám mohl stačit free nástroj, ale pokud máme v plánu modelovat komplexní řešení, ať se jedná pouze o jeden či více diagramů, pak je jasné, že musíme přistoupit ke komerční licenci a zaplatit si za funkcionality.

Jako další kritérium pro výběr vhodného řešení, které bych zde zmínil, je tzv. „user friendly hladina“. Jedná se o to, že na první pohled je jasné, kde najdeme jakou funkci, jak ovládat nástroj. V současnosti je nutné, aby byly nástroje jednoduché na pochopení a na práci s nimi, což je obrovskou zbraní. Jistě jste se všichni setkali se softwarem, ať už se jedná o CASE nástroje pro jazyk UML či o jakýkoliv jiný software, který nebyl vůbec snadný k pochopení jeho funkcionality, kde jaké tlačítko stisknout, aby se vykonala Vámi požadovaná funkce a hlavně vyhledání funkcí. Tento fakt je v dnešní době hlavním trendem a proto si výrobci musí dávat pozor, aby jej splnili a neodradili zákazníky složitostí nástrojů.

Jistě bychom našli mnoho dalších kritérií, dle kterých bychom si vybírali tyto nástroje, napadá mě například jazyk, ve kterém jsou nástroje poskytovány, ale z mého pohledu ty hlavní tři jsem zmínil výše, které považuji za hlavní.

## Závěr

Závěrem je nutno uvést fakt, že se nám podařilo některé produkty stáhnout a vyzkoušet, ať opensource(free) či trial verze komerčních produktů, ale některé nikoliv, proto nejsou mnohdy naše informace, které jsme uvedli o jednotlivých produktech podloženy našimi vlastními zkušenostmi s produkty, což by mohla být další příležitost pro následující semestry, aby se pokusili produkty vyzkoušet a ověřit námi uvedené informace.

## Zdroje

[1] Internetová encyklopedie wikipedia na adrese <http://en.wikipedia.org>

Dále internetové adresy jednotlivých výrobců zpracovávaných produktů

[2] Web firmy IBM na adrese <http://www.ibm.com/us/en/>

[3] Web produktu Violet UML editor na adrese <http://www.horstmann.com/>

[4] Web firmy Borland na adrese <http://www.borland.com/>

[5] Domovská stránka produktu Umbrello UML modeller na adrese <http://uml.sourceforge.net/>

[6] Domovská stránka produktu ArgoUML na adrese <http://argouml.tigris.org/>

[7] Domovská stránka produktu ModelSphere na adrese <http://www.modelsphere.org/>

[8] Domovská stránka produktu Visual Paradigm for UML Enterprise Edition na adrese <http://www.visual-paradigm.com>