

## OO vývojové nástroje

- OO CASE
  - Rose, XDE, RSA od spoločnosti IBM Rational
  - Enterprise Architect od Sparx systems
  - PowerDesigner od Sybase
  - Visual Paradigm for UML od Visual Paradigm
  - Select Architect od Select business solutions
- OO IDE
  - MS Visual Studio 2010
  - Oracle JDeveloper
  - Eclipse

## Produkty spoločnosti IBM Rational

### História

- Rational Rose
- Rational eXtended Development Environment (XDE)
- Rational Software Modeler
- Rational Modeling Extension for MS.NET

### Súčasnosť

Rational Software Architect (RSM, RME for MS.NET – súčasťou RSA)

### Základné informácie produktov IBM rational

- produkty firmy IBM a Rational Software – výhradný copyright
- Samostatné nástroje, ktoré sú integrovateľné s prostrediami Visual Studio .NET, Eclipse IDE alebo IBM WebSphere Studio ako doplnujúci modul
- plnohodnotné CASE systémy poskytujúce objektovo – orientované návrhárské a vývojárske vybavenie
- Každý z nástrojov je silný nástroj na analýzu, návrh, komunikáciu a dokumentáciu cez životný cyklus projektu, pričom môže integrovaný s vývojárskou platformou používanou na tvorbu aplikácií a systémov (napr. VS.NET, Eclipse)

## Nástroje umožňujú

- znázorniť rôzne typy informácií pomocou UML prvkov a non-UML prvkov
- pracovať efektívnejšie možnosťou definovania vlastných vzorov, ktoré zachytávajú osvedčené architektúry a návrhy a definujú programové a modelové šablóny
- produkovať kód lepšej kvality pri podpore validácie modelov pre UML a špecifických programových jazykov
- pracovať lepšie s inými členmi tímu (analýza, návrh a kódovanie sa robí v jednom prostredí, v rámci projektu sa dajú tvoriť mnohonásobné modely)
- zdieľať informácie projektu jednoduchšie možnosťou distribúcie návrhu a vývoja prostredníctvom tvorby assets (RAS aktív)
- produkovať heterogénne riešenia pre potreby aplikácií možnosťou generovať kód do rôznych jazykov
- rozšíriť o ďalšie vývojárske nástroje pomocou Rational Extensibility API

## Rational XDE

- tvorbu diagramov free-form, activity, class, component, deployment, sequence, statechart, and use-case diagrams
- podporuje modelovacu konvenciu Unified Modeling Language (UML) 1.4 , využíva UML prvky a Non-UML prvky (v taboch Geometric Shapes and UML Deployment v Toolbox)
- plná podpora tvorby vzorov (pattern) a aplikovanie existujúcich vzorov
- tvorba Reusable Asset Specification (RAS) assets (kolekcia príbuzných artefaktov na riešenie opakujúcich sa problémov uložených v repozitory)
- z Rational XDE Extensibility API (RXE API) umožňuje písať programy v rôznych jazykoch a tvorbu súborov XML a CSV
- RUP Configuration for Java Developers
- RUP Configuration for Microsoft Visual Studio .NET Developers predstavuje Rational Unified Process špeciálne upravený pre potreby .NET vývoja
- Možnosť analýzy, definície a návrhu logického a generovania fyzického modelu dát pre databázy IBM DB2, Microsoft SQL Server, Sybase Adaptive Server a Oracle
- Plnú podporu kódových šablón, umožňujúce vytvorenie šablón pre často používané časti kódu
- Podporu na generovanie kódu do Javy, reverzné inžinierstvo modelov z kódu, vrátane J2SE a J2EE
- Podporu modelovania a generovania JSP, tag knižníc a HTML dokumentov

- generovanie kódu do Visual C++, VB a C# a reverzné inžinierstvo modelov z kódu
- modelovanie a generovanie ASP.NET súborov
- generovanie do ANSI C++ a reverzné inžinierstvo
- analýzu run-time vykonávania s PurifyPlus
- automatickú tvorbu tzv. Trace Sequence Diagram vo Visual Trace

## **Rational Software Architect**

- Tvorba UML, free-form, topic (tematických) a browse (prehľadových) diagramov
- Business modelovanie s BPMN 2
- Podpora profilov, stereotypov, obmedzení, návrhových vzorov a taggovaných hodnôt
- Tímová a multi-modelová podpora, porovnávanie a zlučovanie modelov a SCM (Software Configuration Management) integrácia
- Rýchly vývoj vlastných prvkov:
  - UML založený DSL
  - Transformácie model na model pomocou grafických mapovacích nástrojov
  - Transformácie model na text/kód
- Vstavané transformácie – UML do SQL, XSD, Java, JPA, C#, VB.NET, Corba a aj spätne
- Grafické editory pre Javu, C# a VB.NET
- Java – nástroje na analýzu architektúry, recenziu a metriky
- C# - nástroje na recenziu a metriky
- Podpora:
  - RAD (Rapid Application Development) nástrojov a sprievodcov
  - MDA (modely a transformácie)
- Reprotovanie modelu cez BIRT (Business Intelligence and Reporting Tools) - generovanie HTML, PDF a XML reportov
- Generovanie Javadoc s detailnými návrhovými diagramami
- Zabudované Crystal Reports nástroje
- IE a IDEF diagramový editor pre databázové objekty
- Konzola Cloud klient – využívanie Cloud zdrojov
- Dodatočné rozširovacie možnosti cez EMF (Eclipse Modeling Framework)

- Možnosť viacerých rozšírení
  - Architektonické frameworky
  - Plánovanie rozmiestnenia
  - SOA (SoaML, BPEL)
  - C++ vývoj
  - Komunikačné aplikácie
  - Simulačný toolkit

## **Integrácia s inými Rational produktami**

### **XDE-PurifyPlus**

- Poskytuje schopnosť vykonávať run-time analýzy na aplikáciách generovaných v XDE. Ide o vykonávanie kódu a analýzu zozbieraných dát na nájdenie programových chýb a spôsobov na zvýšenie výkonu.

Skladá sa z nasledovných nástrojov:

- Purify (očistiť) slúži na odhalenie pamäťových dier. Ide o pamäť, ktorú aplikácia nepoužíva a ani nemôže byť vrátená systému na znovu použitie.
- Quantifi (kvantifikovať) slúži na odhalenie úzkych miest vo výkone.
- PureCoverage (čisté pokrytie) slúži na monitorovanie pokrytia kódom.

### **XDE-ClearCase**

- Umožňuje:
  - vybrať základný a vstupné modelové súbory
  - porovnať vybrané súbory
  - prezrieť rozdiely a konflikty
  - manuálne alebo automaticky vyriešiť konflikty
  - zlúčiť súbory a vytvoriť jeden zlúčený model

### **XDE-RequisitePro**

- Umožňuje asociovať požiadavky a dokumenty s use case a modelmi.

### **XDE-Rational Unified Process**

- Predstavuje konfigurovateľný vývojový proces pre softvér.

### **Rational XDE Extensibility (RXE)**

- RXE je aplikačné programové rozhranie (API) s nástrojmi, ktoré umožňujú používateľovi podľa požiadaviek upraviť a rozšíriť nástroje.
- RXE obsahuje RXE API (poskytuje programovací prístup k modelovacej sémantike), Code Templates API (programové šablóny napísané v Jave, COM jazyku alebo CLR jazyku) a Patterns API (vkladanie vlastného správania do vhodných častí prostredníctvom vzorov).

## Enterprise Architect

- Produkt spoločnosti Sparx systems
- Vytvorený ako samostatný produkt vo viacerých edíciách (Desktop, Professional, Corporate, Systems engineering, Business and software engineering, Ultimate)
- Možnosť MDG integrácie alebo linku pre Visual Studio a Eclipse
- Možnosť doplnenia frameworkov UPDM (DoDAF, MODAF), SysML, DDS, Zachman, TOGAF, SOMF

## Funkcionality nástroja

- Modelovanie v UML 2.3, BPMN, (DFD, ERD)
- Podpora profilov, stereotypov, obmedzení, návrhových vzorov a taggovaných hodnôt
- Generovanie reportov do HTML a RTF, tvorba šablón pre reporty
- Manažment požiadaviek
- Integrácia systémov kontroly verzií (SCC, CVS, SVN, TFS)
- Import/ Export XMI a CSV
- Podpora metodiky UP
- Round trip inžinierstvo (RTE) kódu vo viacerých jazykoch – Java, C#, VB, C, C++
- Inžinierstvo WSDL, XSD
- Model Driven Architecture - transformácie
- Úložisko:
  - Súborová repository
  - DBMS repository - *MSSQL Server, MySQL, Oracle, PostgreSQL, Progress OpenEdge, MSDE, Adaptive Server Anywhere*
- Modelovanie a FE a RE databázy – DB2, MS SQL, Oracle, MySQL, Informix, PostgreSQL, Ingres, Firebird, InterBase, Sybase
- Skriptovanie v JScript, VBScript, Javascript

- Generovanie kódu z modelov správania
- Generovanie BPEL z BPMN 1.1
- Sledovanie vývoja cez modely v tzv. Trace diagramoch
- Verifikácia, validácia a analýza modelov
- Debugovanie, kompilácia, vizualizácia atestovanie kódu
- Modelovanie a inžinierstvo v SysML
- Aktualizácia šablón transformácie a generovania
- Tímová podpora a bezpečnosť prístupov
- Team review
- Relationship matrix
- Tvorba slovníka
- Audit modelových zmien
- Porovnávanie a zlučovanie modelov

## **Všeobecný popis základných funkcionalít v rôznych OO nástrojoch**

### **Hierarchická štruktúra repozitory v nástrojoch**

- Solution (pre XDE)
- Project
- Model
- Element
- Diagram

#### **Project (Projekt)**

- Kolekcia súborov v rámci partikulárneho adresára v stromovej štruktúre riešenia predstavujúcich aplikáciu alebo systém vo vývoji
- Zahrňuje modely, kód, textové súbory a iné s projektom súvisiace artefakty

#### **Model (Model)**

- Obsahuje hierarchiu objektov nazývaných elementy alebo prvky (napr. diagramy, dátové elementy, služby a relácie)

- Rozlišujeme:
  - PIM – platformovo nezávislý model
  - PSM – platformovo špecifický model
- Alebo
  - Analytický - služby
  - Návrhový - architektúra
  - Implementačný – na konkrétnu platformu

## Diagram

- Prostriedok na vizualizáciu a manipuláciu s modelovými elementmi a reláciami v modeli
- Zobrazuje sa v Diagramovom okne

## Validácia

Môže zahrňovať:

- validovanie modelov a diagramov
- validovanie jazykovo špecifických modelov
- validovanie dátového modelu
- validovanie externých referencií

Typy validácie:

- Sémantiky
- Syntaxe

Dve skupiny špecifických val. pravidiel:

- tzv. živé, ktoré hlásia chybu ihneď po vykonaní akcie
- tzv. dávkové (batch), ktoré zahlásia chybu až po spustení príkazu validovať

Spúšťa sa:

- všeobecne kliknutím pravého tlačidla na element a zvolením **Validate**
- Položkou hlavného menu

Výsledky validácie sa zobrazujú vo výstupovom okne – umožňuje navigáciu na miesto porušenia

Existujú tri úrovne validácie:

- Error – ide o porušenie, ktoré je nedovolené
- Warning – ide o porušenie, ktoré je dovolené ale sémanticky nesprávne (resp. musíme si ho byť vedomý)
- Information message – ide o porušenie konzistencie s UML štandardami alebo s jazykovou špecifikáciou, ktoré nemusí byť opravené

## Import a export XMI

- XMI (XML Metadata Interchange) je typ súboru, ktorý umožňuje výmenu UML informácií medzi rôznymi vizuálnymi modelovacími nástrojmi
- Nie všetky reprezentácie XMI majú korešpondujúce elementy s príslušným nástrojom, preto sú reprezentácie nemajúce jednoznačné mapovanie importované (exportované) ako podobné elementy a všetky nekompatibilné reprezentácie ignorované

## Profil/y

- Umožňujú prispôbiť jazyk pre vlastnú doménu alebo metódu
- Modelu môže pridať vlastnosti, funkcionalitu, podmienky validácie atď.
- Sú mechanizmy UML, ktoré umožňujú rozšíriť sémantiku UML použitím stereotypov a tagovaných hodnôt
- Niektoré sú predefinované (napr. RUP Analysis) ale možno pridávať aj navyše

## Stereotypy (stereotypes)

- Sú rozširovací mechanizmus UML
- Vytvára sa variácia modelového elementu alebo relácie, ktoré sa objavujú v rovnakej forme ale využívajú v rôznych situáciách
- Možno použiť na tvorbu nových typov resp. špecifikáciu elementov založených na existujúcich elementoch
- Štandardne sa definuje vo vlastnostiach elementu UML>Stereotype
- Slovo uzavreté v « » (napr. «library», «table», «call», «create»)

## Kľúčové slová (keywords)

- Kľúčové slovo je textový opis prvků alebo konektoru, ktoré reprezentujú elementy bez vlastnej jedinečnej notácie
- Napr. «enumeration», «interface»



- Nie sú editovateľné na rozdiel od stereotypov
- V niektorých nástrojoch nemusia existovať, lebo sú realizované ako stereotypy

## **Obmedzenia (constrains)**

- Predstavuje rozširovací mechanizmus, ktorý určuje spresnenie štruktúry a správania elementu ku ktorému je pridelené
- Podmienky alebo reštrikcie sú v textovom údaji nazývanom telo obmedzenia
- V diagramovom okne sa zjavujú ako poznámky, ktorých telo je v zložených zátvorkách
- Podporuje Object Constraint Language (OCL) pre výrazy tela
- Vpridanie cez Toolbox, kontextové menu, alebo okno vlastností

## **Taggované hodnoty**

- dodatočné info pre element priamo podporované UML
- Preferovaná metóda rozšírenia možností generovanai kódu modelovacieho nástroja
- Napr. všeobecne pre triedy potrebujeme sledovať vlastnosť, ktorú nemá štandardne, tak vytvoríme tag napr. XYZ, a pre každú triedu môže nadobúdať nejakú taggovanú hodnotu

## **Code templates (kódové šablóny)**

- Predstavujú zdrojový kód (šablónu) pre element v modeli
- Pri generovaní kódu pre daný element sa šablóna stane časťou generovaného zdroja
- Poskytujú rozšírenie model ku kódu pre potrebu generátorov kódov
- Obsahujú telo, ktoré môže byť napísané v ľubovoľnom jazyku aj skriptovacím (.NET, Java, Javascript)
- Môže mať aj parametre
- Je rozvinutá (expanded), keď nastáva generovanie kódu
- Oblasť je označená v generovanom kóde napr. začína z // (Begin Template Expansion) a končí //(End Template Expansion)
- Existujú nastavenia generovania šablón

## **Delenie**

- Interné - napojené len na jeden element a uložené v modeli

- Externé - napojené na viac elementov a uložené napr. v externom XML súbore

## **Návrhové vzory (design pattern)**

- Tvorba je jednoduchá, len treba mať znalosti a skúsenosti izolovať opakovateľnú štruktúru alebo proces a transformovať ju do znovu použiteľného návrhu
- Existujú predefinované vzory, napr. vzory GoF (Interface pattern)
- Obsahuje funkcionalitu na definovanie vlastných vzorov
- Mali by zahrňovať dokumentáciu (wizard na aplikáciu, HTML, XML dokumentáciu)

### Výhody vzorov

- Riešia problém opakovania sa v rámci daného kontextu
- Môžu byť zdieľané
- Riešia problémy dobre definovanými riešeniami (návrh experta, dobre zdokumentovaný)
- Majú individuálne a široké použitie (v rôznych fázach vývoja, vnorené vzory)
- Podporujú znovu použiteľnosť softvéru (vytvorenie, obnovenie a údržba SW, znižuje náklady a šetrí čas znovunávrhu )

## **Reusable Asset Specification (RAS)**

- Vzory možno importovať alebo exportovať z/do RAS.
- Je štandard pre znovu použiteľné artefakty
- Obsahujú elementy, štruktúru a dokumentačné štandardy pre jednoduchú výmenu dát medzi používateľmi Rational nástrojov
- Poskytujú konzistentnú metodológiu na organizáciu, dokumentovanie, zdieľanie, uchovávanie a obnovenie riešenia (solution)

### RAS poskytujú:

- organizáciu rôznych ale príbuzných súborov v jednom package
- prezentáciu konzistentných informácií vo všetkých assets rovnakého typu
- mnohonásobné a flexibilné kľúče na hľadanie assets
- možnosti uchovávať a obnoviť assets z jednej alebo viacerých RAS assets repozitórií
- použitie jednoduchých variácií štandardných funkcií importu a exportu na načítanie package assets
- údržbu aktivít zabezpečenia a správnej obnovy súborov

RAS asset:

- Musí mať pridelený profil: default (predvolený), web services (na tvorbu servisov) alebo component (na tvorbu komponentov).
- Je zipovaný súbor uchovávajúci súbory z ktorých sa asset skladá, má príponu .ras
- Rasset.xml - Manifest udržiava dáta potrebné na obnovenie štruktúry RAS asset , poskytuje informácie pre používateľa vo forme opisu RAS asset (aký problém rieši, kde môže byť aplikovaný, ktoré jazyky a platformy podporuje), kľúčových slov pre vyhľadávanie a dokumentáciu (ako používať asset)
- Môže byť uložený
  - v súbore v adresári na disku
  - uverejnený v repozitory – lokálna, pracovného priestoru.

Obidve možnosti sú prístupné zo sprievodcu exportu. Ale len asset v repzitory možno nájsť cez vyhľadávaciu schopnosť nástrojov

## Modelovanie aplikácií

- Pomocou jazyka UML, jeho diagramov a prvkov a spomenutých rozširovacích mechanizmov
- Round Trip Engineering je proces špirálového vývoja a zahrňuje generovanie kódu, reverzné inžinierstvo a synchronizáciu
- Pri generovaní a reverznom inžinierstve ide o jednostrannú operáciu
- Na riešenie jednosmerných synchronizačných problémov použijeme znovu príkaz generovania kódu alebo príkaz reverzného inžinierstva
- Samotná synchronizácia je obojstranná operácia, ktorá upraví model a kód a ich časti vzájomným porovnaním a znovu nastolením konzistencie medzi nimi – na základe nastavenia synchronizácie
- Synchronizačná politika môže byť nastavená napr. na synchroize (umožnená obojsmerná synchronizácia), disabled (model sa nezúčastňuje synch.), accept code changes (len zmeny kódu sú premietnuté do modelu), accept model changes (len zmeny modelu sú premietnuté do kódu).
- Všetky vzniknuté konflikty sa vypisujú v okne výstupov alebo úloh

## Modelovanie dát

- Modelovanie dát podporuje Round-Trip Engineering (RTE) medzi dátovým modelom a DDL (Data Definition Language) alebo podporovaným systémom riadenia databázy (SRBD).

- Podporované sú rôzne DBS: ANSI SQL 92, IBM DB2, Oracle Database, Microsoft SQL Server, Sybase Adaptive Server atď.

Nástroje umožňujú:

- reverzné inžinierstvo dátového modelu z existujúcej databázy alebo DDL
- transformovať tabuľky na triedy
- transformovať triedy na tabuľky
- porovnať a synchronizovať dátový model s existujúcou databázou alebo DDL
- vytvoriť elementy dátového modelu na alokáciu tabuľkového priestoru
- definovať obmedzenia (constrains) a indexy
- vytvárať trigery (triggers), domény (domains), uložené procedúry (stored procedures) a pohľady (views)
- generovať DDL a databázu z dátového modelu

Na modelovanie dát sa primárne používa diagram tried (class), alebo sa môže použiť diagram komponentov (component) a nezávislých tvarov (free-form) alebo priamo záložku s prvkami na modelovanie dátového modelu (napr. Data Modeler v Toolbox)

Pri tvorbe DM možno použiť:

- Špecifickú SRBD modelovú šablónu – určí sa konkrétny databázový cieľ
- Spoločnú šablónu ANSI SQL 92 – všeobecný štandard

V OO prístupe sa rozlišuje:

- Logický model
  - Tvorí sa pomocou prvkov Diagramu tried, môže slúžiť aj ako konceptuálny model
  - Mal by obsahovať triedy, ktoré mapujú tabuľky dátového modelu v package
  - Transformáciou tried sa generuje fyzický model tabuliek
- Fyzický model (medzimodel)
  - Mal by byť rozdelený na databázový komponent, package pre tabuľky, uložené procedúry, pre domény a pohľady
  - Tvorí sa v pomocou prvkov tabuľka a pod.

## **Prvky používateľského rozhrania**

- Explorer, navigátor - zobrazuje štruktúru projektu, repository triedy
- Diagramové okno - zobrazuje aktuálne diagramy

- Okno vlastností - zobrazuje vlastnosti vybranému prvku
- Príkazové okno - umožňuje pridávať príkazy, debugovať alebo ohodnotiť výrazy
- Dokumentačné okno - umožňuje vytvoriť a editovať dokumentáciu k modelu
- Výstupové okno - zobrazuje správy o statuse modelu, aktivity modelu a príbuzné elementy
- Pomocník - zobrazuje pomoc k vybranému prvku
- Toolbox - poskytuje prvky pre diagramy, voľné pridávanie, drag and drop, klik a klik
- Explorer serverov - poskytuje prehľad serverov
- Toolbar - poskytujú jednoduchý a rýchly prístup k najčastejšie používaným príkazom

#### **Príkazy menu závisia na:**

- Package, ktorý používame
- Funkcionalite sprístupnenej načítanými profilmi
- Type modelu, ktorý je otvorený
- Type prvku, ktorý je vybraný
- Okne v ktorom vyberieme prvok

## **MS Visual Studio 2010 Ultimate a modelovanie**

- Klasické IDE
- Rozšírené o CASE funkcionalitu - Modeling project zahrnutý do Architecture and modeling packu
- Vhodné doinštalovať Visualization and modeling feature pack, ktorý poskytuje ďalšie funkcionality

#### **Funkcionalita**

- Architecture explorer – slúži na prehliadanie prvkov kódu drill down spôsobom (od riešenia až po metódu triedy)
- UML 2.1 diagramy - Use case, Activity, Sekvenčný, Tried, Komponentov
- Model explorer – hierarchická štruktúra modelu a elementov
- Generovanie sekvenčného diagramu z kódu na zobrazenie interakcií medzi inštanciami (diagramy možno editovať)
- Integrácia modelových elementov s Team Foundation Server Work Items

- Diagram vrstiev (layer)
  - Vytvorenie diagramu z kódu alebo priame definovanie
  - Organizovanie tried, menných priestorov, kódových súborov, riešení do logických vrstiev a zobrazenie existujúcich alebo plánovaných závislostí (prezentačná, service, business, dátová)
  - Validácia kódu voči vrstvám, zahrnutie do check-in a built procesu,
- Klasické reverzné inžinierstvo diagramu tried z kódu
- Diagramy závislostí (DGML):
  - na vizualizáciu väzieb a funkčných oblastí kódu
  - generovať z kódu riešenia,
  - manuálne vkladať projekty, triedy, assembly, komponenty a zobrazia sa relácie
- Export diagramov do grafických editorov
- Verziovanie pomocou TFS Source safe

### **Rozšírenia pre diagramy a elementy**

- Prispôbenie pomocou profilov a stereotypov
- Validovanie UML modelov voči vytvoreným obmedzeniam
- Vytvorenie vlastných elementov
- Generovanie kódu a iných artefaktov z modelov
- Pridanie menu príkazov a handlerov do UML diagramov, ktoré umožňujú aktualizáciu modelu a generovanie súborov
- Zbalenie a distribúcia týchto rozšírení do VISX

### **Visualization and modeling feature pack**

- Generovanie kódu z UML diagramu tried (pomocou T4 textových šablón)
- Reverzné inžinierstvo z kódu
- Generovanie diagramov závislostí pre C, C++, a ASP.NET (organizácia a vzťahy, DGML Graphs)
- Import UML modelov z XMI
- Validovanie C, C++ kódu voči diagramom vrstiev