InterSystems Caché

Post-Relational Database

Martin Holoubek
xholoub@fi.muni.cz

Úvod

- InterSystems byla založena shodou okolností ve stejném roce jako její největší konkurent Oracle – 1978
- Caché je velmi vyspělý nástroj pro tvorbu komplexních aplikací založených na práci s persistentními daty
- Díky své architektuře uložení dat je schopna velmi efektivně pracovat jak z pohledu relačního tak objektového

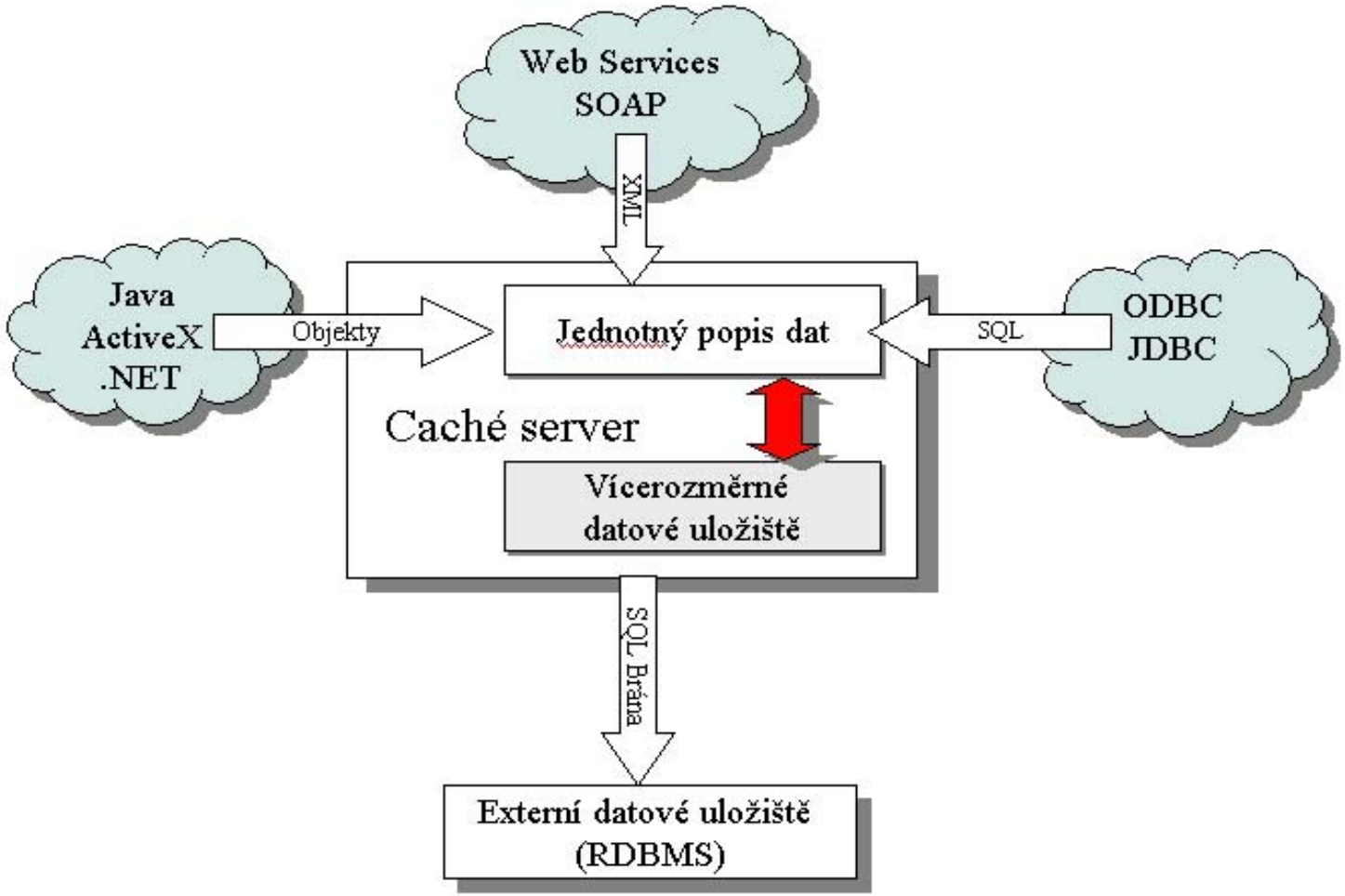
Architektura databáze 1/3

- Základem modelování komponent aplikací jsou v Caché objekty
- Objekty jsou organizovány do tříd s vlastnostmi a metodami
- Definice tříd spolu s ostatními daty jsou uloženy v jednotném uložišti zvaném Caché Class Dictionary
- Tento „slovník tříd“ tvoří databázi, ke které můžeme objektově přistupovat

Architektura databáze 2/3

- Zkompilováním definice třídy dojde k vytvoření dvou různých, navzájem synchronizovaných sad kódu, které zajistí optimální přístup k datům jak pomocí objektového, tak relačního přístupu

Architektura databáze 3/3



Definice třídy

- Třídy v Caché můžeme tvořit několika způsoby:
  - „ručně“ pomocí Caché studia
  - „relačně“ pomocí DDL jazyka SQL
  - „programově“ pomocí objektů. Caché obsahuje systémové třídy pro práci se slovníky tříd
  - pomocí XML.
  - pomocí UML. Caché ovládá import i export ze systémů Rational Rose a MS Visual Modeller

Typy tříd 1/2

- Abstraktní třídy – nelze od nich vytvářet objekty. Odvozují se od nich třídy představující konkrétní modelované objekty.
- Registrované třídy – implementují plnohodnotné objekty ve smyslu OOP. Nepodporují persistenci, proto se zpravidla využívají pro tvorbu kódu aplikační logiky.
- Persistentní třídy – jsou odvozené od registrovaných tříd. Jsou trvale uloženy. Při kompilaci vytváří kód, zajišťující komunikaci mezi aplikační a fyz. vrstvou.

Typy tříd 2/2

- Vnořené třídy – podobné persistentním, ale nemají vlastní uložiště. Data v nich obsažená jsou uložena spolu s daty persistentní třídy jež na vnořenou třídu odkazuje.
- Odvozené třídy – odvozené od jedné či více uživatelských tříd (dědičnost, polymorfismus).
- Datové typy – jsou třídy pro popis dat. Mohou být libovolně složité. Zajišťují správnou konverzi mezi zobrazovanou (externí) podobou a jejich uloženou (vnitřní) podobou.

Vztah objektů a relačních tabulek

- Při kompilaci definice třídy se vytváří dvě synchronizované sady kódu
- Nezávislost objektové a relační projekce datového modelu je vidět i na možnosti přiřadit třídě alternativní název jako tabulky (důvod – např. klíčová slova v SQL)

Vztah objektů a relačních tabulek

- Vlastnost třídy, mající jednu hodnotu je transformována na sloupec relační tabulky
- Vlastnost typu pole, nabývající libovolného počtu hodnot je převedena na samostatnou tabulku obsahující odkaz na původní tabulku
- Metoda třídy odpovídá uložené proceduře v SQL

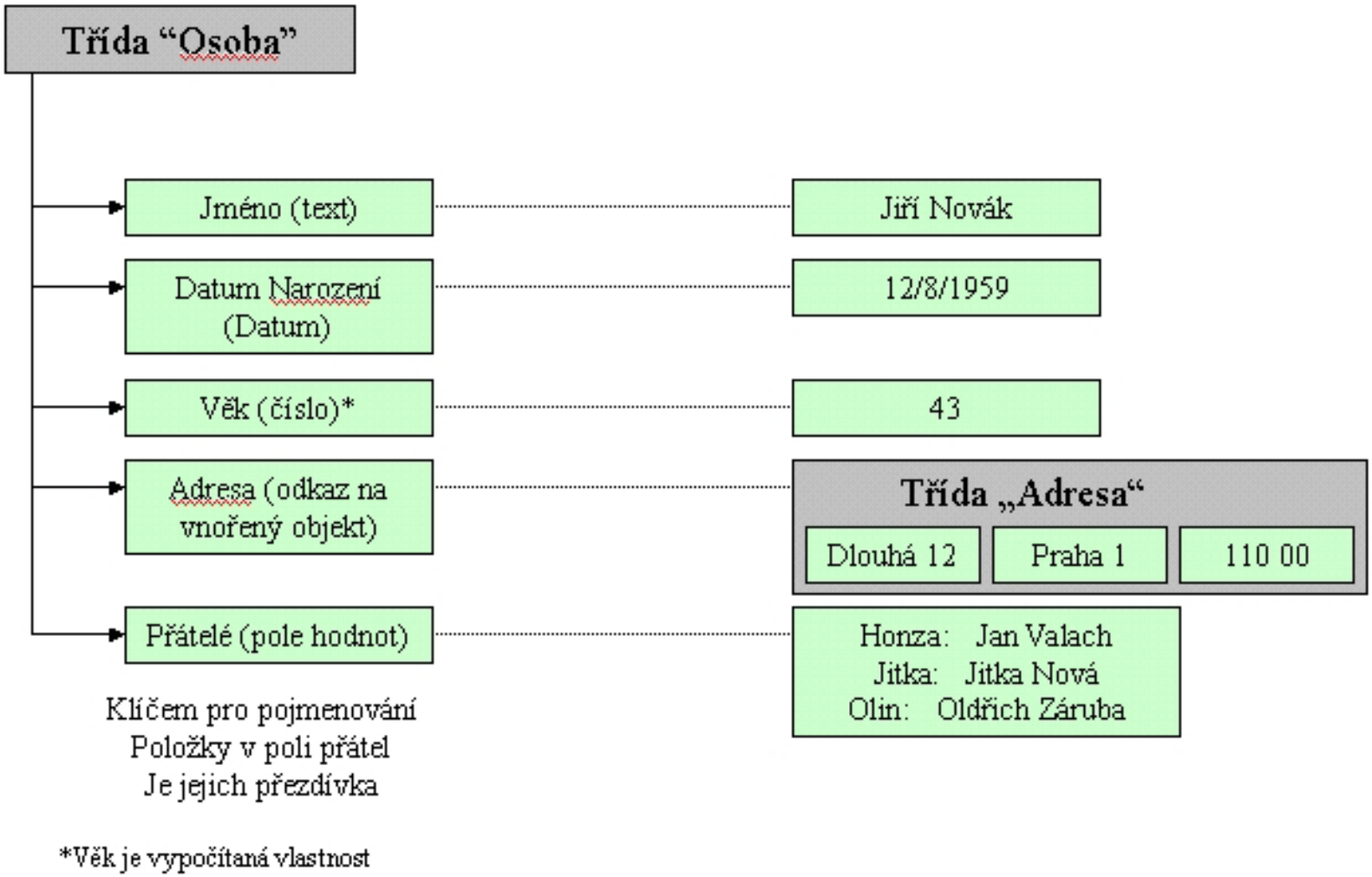
Příklad - definice třídy objektově

```
Class User.Osoba Extends %Persistent [ ClassType = persistent,
   ProcedureBlock ]
{
   Property DatumNarozeni As %Date;
   Property Jmeno As %String;
   Property Vek As %Integer [ Calculated ];
   Property Pratele As %String [ Collection = array ];
   Property Adresa As User.Adresa;

   Method VekGet() As %Integer
   {
   set vek=+$h-..DatumNarozeni\365
   Quit vek
   }

   Method Pozdrav()
   {
   write "Dobrý den!",!
   quit
   }
}
```

Příklad - objektový model



Příklad - relační model

Tabulka "Osoba"

ID	Jméno	Datum Nar.	Věk	Adresa_Ulice	Adresa_Sídlo	Adresa_PSČ
1	Jiří Novák	12/8/1959	43	Dlouhá 12	Praha 1	110 00

Tabulka "Osoba_Přátelé"

Osoba	ID	Přátelé	Klíč položky
1	1||Honza	Jan Valach	Honza
1	1||Jitka	Jitka Nová	Jitka
1	1||Olin	Oldřich Záruba	Olin

Implementace SQL

- Caché implementuje plně základní úroveň standardu SQL-92 s několika málo rozšířeními jako např.
  - Podpora uživatelsky definovaných dat. typů
  - Podpora objektové syntaxe
  - Podpora dědičnosti a sub-classingu
  - Možnost definovat strukturu uložení dat za účelem maximalizace výkonnosti

Příklad – objektové syntaxe v SQL

- ```
  SELECT A.Jmeno, B.Jmeno FROM
  Osoba AS A, Osoba AS B WHERE
  A.ID=1 AND Partner=B.ID
  ```

- ```
  SELECT Jmeno, Partner->Jmeno FROM
  SQLUser.Osoba WHERE ID=1
  ```

Skriptovací jazyky Caché

- V Caché lze napsat celé aplikace. A to i bez toho, aby kód jakkoliv přistupoval k datům uloženým v databázi.
- Caché obsahuje dva skriptovací jazyky. Starší Caché ObjectScript a novější Caché Basic. Z obou vzniká stejný zkompilovaný kód.

Caché ObjectScript 1/4

- Lze pomocí něho psát i těla metod Caché tříd
- Značně se liší od klasických programovacích jazyků. Například tím, že nepracuje s datovými typy, nemá rezervovaná slova apod.
- Příkazy lze zapisovat na řádky za sebe. Musí se ovšem oddělit mezerou. Pokud příkaz nemá argumenty (např. ELSE), pak musí být odděleny dvěma mezerami.

Caché ObjectScript 2/4

- Caché ObjectScript je case sensitive
- Rozlišuje dva typy proměnných
  - Persistentní (globály) např. ^x
  - Dočasné (pouze v paměti vyhrazené procesu) např. x
- Funkce jsou též dvojího typu
  - Vnitřní (systémové) před název umístíme $
  - Vnější (uživatelské) před názvem je $$

Caché ObjectScript 3/4

- V Caché ObjectScriptu existují tři úrovně rutin:
  - Nejvyšší je úroveň makro (MAC) – v ní se píše aplikační kód
  - O úroveň níž se nachází částečně přeložený kód (INT), který se je vlastně čistý ObjectScript v textové formě, kde jsou všechna makra rozvinuta
  - Nejníže se nachází binární kód (OBJ), přeložený a připravený k interpretaci databázovým strojem

Caché ObjectScript 4/4

- Při psaní rutin v Caché ObjectScriptu je možno kombinovat různé jazyky vložením úseků kódu napsaným v:
  - SQL
  - HTML
  - JavaScript
  - XML
  - Java (dokonce umožňuje i napsání celé metody třídy v čisté Javě)

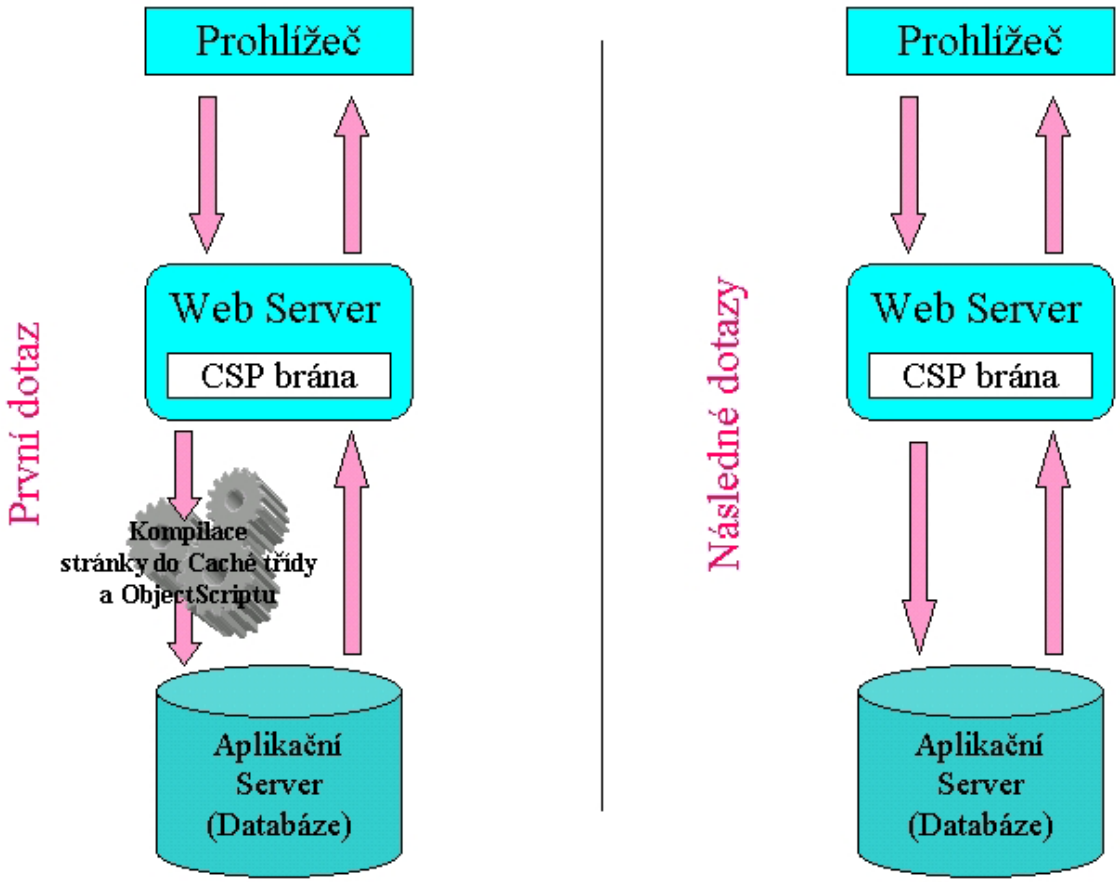
Caché Basic

- Byl vytvořen za účelem oslovit komunitu prográmátorů píšících ve Visual Basic nebo VBScript
- Lze pomocí něho psát jak rutiny, tak celé metody
- Caché Basic obsahuje rezervovaná slova
- Je možné kombinovat Caché Basic a Caché ObjectScript i v rámci definice jedné třídy

Caché Server Pages 1/6

- Caché obsahuje veškeré nástroje pro tvorbu serverové strany aplikační logiky webových stránek
- CSP je založena na podobném principu jako ASP.NET nebo JSP.
- Stránka je při prvním dotazu zkompilována do Caché ObjectScriptu. Při dalších dotazech se již volá zkompilovaný kód.

Caché Server Pages 2/6



Caché Server Pages 3/6

- Základem pro tvorbu dynamických CSP stránek je sada předdefinovaných značek
- CSP značky lze podle účelu rozdělit:
  - Základní značky – propojují HTML stránku s aplikačním serverem. Např. <csp:class>
  - Řídící značky – řídí běh programu. Např. <csp:if>, <csp:while>
  - Ostatní značky – například pro vykonávání dotazů do databáze <csp:query>

Caché Server Pages 4/6

- CSP značky jsou definovány pomocí XML
- Pro vývoj komplexních aplikací zpravidla sada značek nedostačuje, proto Caché poskytuje možnost definovat vlastní CSP značky
- V libovolném místě CSP je možno vyhodnotit jakýkoliv platný výraz v Caché ObjectScriptu pomocí konstrukce

```
#(výraz Caché ObjectScriptu)#
```

Caché Server Pages 5/6

- Příklad zobrazení aktuálního data serveru:

```
<body>
  Dnes je #($zd($h,4))#
</body>
```

- Stejného výsledku můžeme dosáhnout např. také pomocí CSP skriptování

```
<script language="cache" runat="server">
  write "Dnes je"_$zd($h,4)
</script>
```

Caché Server Pages 6/6

- Pro ulehčení práce při vývoji CSP stránek Caché podporuje šablony
- CSP technologie podporuje nejen psaní klasických (D)HTML stránek, ale též tvorbu WAP a XML stránek
- Technologii CSP podporuje:
  - Netscape
  - Internet Explorer (Linux, Windows)
  - Mozilla (Linux, Macintosh, Windows)

Caché SQL Gateway

- Pomocí SQL brány umí Caché přistupovat k datům, která jsou uložena v jiném databázovém systému
- K těmto datům můžeme přistupovat objektově a to dokonce, i když se připojujeme k relačnímu databázovému serveru
- Podporovanými servery jsou:

  Microsoft SQL Server, Oracle, Sybase, Informix Online, …

Caché versus Oracle

- Na oficiálních stránkách InterSystems jsem nalezl několik zajímavých srovnávacích testů. Uvedu zde jen několik málo výsledků. Definice databáze, parametry testovacího stroje, použité dotazy apod. naleznete na www.intersystems.com

Caché versus Oracle

	Časová ztráta při indexování bitovou mapou
Caché	**1,45**
Oracle	**6800**

	Doba aktualizace 1 000 řádků	Počet aktualizovaných řádků za sekundu
Caché	**0,11 s**	**8 787**
Oracle	**34,01 s**	**29**

Caché versus Oracle

		Doba operace (sekundy)	
Dotaz	Počet vrácených řádků	Caché	Oracle
Select count(*) from Osoby where Stav = 'ŽV'	200 108	0,0028	0,0050
Select count(*) from Osoby where Pohlaví = 'M'	4 998 018	0,0056	0,0100
Select count(*) from Osoby where Věk < 40	3 038 594	0,0367	0,0400
Select count(*) from Osoby where Věk between 10 and 40	3 184 366	0,0366	0,0300
Select Jméno from Osoby where Stav = 'ŽV' and Věk = 50	4 448	13,6239	14,0200
Select Jméno from Osoby where (Věk between 5 and 6 or Stav = 'ŽV') and BarvaVlasů='zrzavá'	42 562	17,8577	18,0100

Odkazy

- Oficiální stránky InterSystems
  - www.intersystems.com
  - www.intersystems.cz
- Databázový svět
  - www.dbsvet.cz