

GEPRA – popis systému

Jaroslav Janda

GEPRA – GEnérátor PRogramových Aplikací nabízí integrované vývojové prostředí pro rozsáhlé aplikace v relačním databázovém prostředí FoxPro (FoxBASE+), podporuje vývoj software po celou dobu životního cyklu software od počátečních specifikací až po dlouhodobou údržbu u uživatele.

1. Důvody vytvoření systému GEPRA

Vše souvisí s podnikatelským záměrem firmy BM z Českých Budějovic, která vznikla v prvních měsících roku 1990 na základě poptávky po kvalitních a komplexních službách v oblasti řízení státních, družstevních a soukromých podniků. Cílem firmy bylo vytvořit integrovaný systém řízení a zpracování informací použitelný v různých typech organizací. Základním heslem tohoto systému bylo: Pracovní místo = pracovní náplň = odpovídající software.

Hned zprvopočátku se rýsovalo několik problémů, které bylo nutno systémově vyřešit před zahájením programátorských prací.

1. Řízení vývojového typu – eliminace zkreslení informace na cestě analytik-programátor; zajištění jednotné komunikace slového řešení s uživatelem; průběžná tvorba projektové dokumentace; spojení dílčích řešení v jeden celek.
2. Generování na míru – na každé pracovní místo je nutné dodat jinou množinu funkcí, která podporuje plně pracovní náplň.
3. Ochrana proti zneužití – na každém pracovišti může přistupovat k systému několik uživatelů s různou kompetencí a zodpovědností.
4. Zajištění integrity dat – zabezpečení dat proti ztrátě v případě havárie; zajištění časové posloupnosti zpracování dat; zajištění bezpečnosti při přenosu dat mezi výpočetními systémy pomocí disket.
5. Generování koncového kódu – eliminace pracného kódování v programovací jazyce; automatizované promítnutí změny v analýze do slového kódu; zajištění dlouhodobé údržby systému.

K těmto problémům, které se vyskytují u každého většího projektu, se řadila nezbytnost dosažení přijatelné ceny pro konečného uživatele.

Dále bylo nutno přihlídnout k tomu, že na základě legislativních změn a nově získaných poznatků z ekonomicky vyspělých států bude nezbytné v dohledné době celý systém přebudovat, a to v poměrně krátkém čase. Každá takováto velká změna si vyžádá reinstalaci systému u všech uživatelů.

Po zvážení všech těchto problémů byl jako základ vybrán systém FoxBASE pro svou cenovou dostupnost a možnost šíření aplikace i s RUNTIME modulem, což velké výkonnější databáze neumožňují. Současně se jevílo jako neúnosné takovýto systém programovat manuálně, a tak vyplynula nutnost generování programového kódu.

Systém FoxBASE nabízel pro oblast generování koncového kódu několik generátorů, které pracují nad tabulkami FoxView. Současně byl k dispozici i jazyk pro tvorbu těchto generátorů. Po analýze těchto prostředků bylo zjištěno, že pro tvorbu rozsáhlých aplikací jsou nedostatečné, neboť podporují jen malou část z životního cyklu software.

Zahraníční systémy CASE, které podporují větší část životního cyklu software, byly v té době málo známé, cenově nedostupné a pokud generují kód tak ne do jazyka Fox.

Konečné rozhodnutí znělo: Musíme si vytvořit svůj vlastní CASE systém, který vytvoří integrované vývojové prostředí v systému FoxBASE a následně FoxPro.

2. Stručný popis práce se systémem GEPRA

Nejprve analytik společně s konzultantem (uživatel) modelují základní funkce systému, navrhnou sledované objekty. Funkce (činnosti) systému i sledované objekty popíší textovou informací, která bude využita při vypracování analytického návrhu.

V další části návrhu definuje analytik jednotlivé položky (atributy) sledovaných objektů a definicí vazeb mezi nimi je uspořádá do relačního datového modelu. V základním funkčním schématu definuje elementární činnosti a jejich atributy. Některé atributy činnosti se uplatní jako podklad pro generování kódu, jiné se uplatní jako podklad pro práci programátora (komunikační zóna analytik – programátor).

Nad úplným schématem funkcí definuje časové závislosti jednotlivých činností a vazby, které zajišťují přenosy dat mezi několika výpočetními systémy v případě distribuovaného zpracování.

Takto připravený projekt je možno předat programátorům, kteří definují uložení datového modelu do databázových souborů včetně indexů. Na základě definice analytika rozhodují, zda programový kód koncové činnosti vyprogramuje běžnými prostředky nebo jej vygeneruje automatizovaně z definice datové základny pro tuto činnost. Spojením práce všech programátorů vzniká výsledný celek.

Po otestování takto připraveného produktu je možné jej usadit do organizační struktury podle popisu jednotlivých pracovních míst. Součástí této fáze je i vytvoření instalačních disket.

Změny, které jsou vyvolány novými požadavky uživatelů nebo chybou během vývoje, se promítnou do definic datové základny, dojde k přegenerování systému, vytvoření nové verze včetně modulu, který zajistí přechod z datové základny předchozí verze na novou verzi.

3. Co podporuje systém GEPRA

3.1. Funkční schéma systému

V této části systému GEPRA se definují jednotlivé funkce (činnosti – tento termín je používán v dokumentaci) nového projektu a jejich závislosti. Funkční schéma je možno definovat „nahrubo“ ve stádiu prvotních specifikací a konzultovat s potenciálními uživateli a později je zjemnit jako ucelený podklad pro práci programátora.

Každá činnost má tyto atributy:

Typ činnosti – udává, o jaký typ činnosti se jedná (kořenové, uzlové, koncové)

Název činnosti – tento název se zobrazí v systému menu za chodu vygenerovaného programu.

Poznámka – doplňující informace o činnosti. Je zobrazována při pohybu v systému menu na poslední řádce obrazovky.

R/w – tento atribut definuje, zda činnost provádí změny dat nebo ne a váže na systém zálohování dat.

Popis činnosti – libovolně dlouhé textové pole, použitelné po dobu celého životního cyklu software (popis úvodních specifikací, podklady pro práci programátora, výsledky betatestů...).

Pomoc – textové pole, které se zobrazí uživateli za chodu vygenerovaného programu v systému menu po stisku klávesy F1 (HELP).

Program – jeden nebo několik příkazů jazyka Fox, které realizují koncovou činnost, nebo příkaz #VJMDAT, který definuje, že tato činnost je generována na základě datového typu V se jménem JMDAT.

Podmínka – logický výraz, který po vyhodnocení vygenerovaným programem určuje, zda se tato činnost zařadí do systému menu jako spustitelná.

Volání, FKEY, Periodicita – definují, že tato činnost je spustitelná i jiným způsobem než pouhým výběrem v systému menu.

Id. činnosti – jednoznačná identifikace činnosti automaticky přidělovaná každé činnosti při jejím založení.

Modul, adresář – definuje uložení vygenerovaného kódu do modulů a adresářů, významné u rozsáhlejších projektů.

Ve fázi vývoje je možno z těchto činností skládat hierarchické funkční schéma (vazby činností). Každá činnost se může vyskytovat na libovolném místě (místech) schématu a je možno ji v něm libovolně přesouvat.

Na základě těchto definic a definic činností vygeneruje systém GEPRA programový kód, který zajišťuje tvorbu a obsluhu systému menu (název činnosti, poznámka, pomoc, podmínka, volání, FKEY, periodičita). Pokud dojde ke změně funkčního schématu (nové činnosti, zrušení činnosti, přeskupení činnosti), je možno přegenerovat výsledný programový kód.

3.2. Datová základna

Za účelem definice dat si systém GEPRA udržuje svůj vlastní adresář dat (data dictionary). Pro možnost definice logických datových typů a logických vazeb si zavádí nový (oproti systémům Fox) typ K a R, pro popis fyzického uložení dat typ * (soubor) a typ I (indexový soubor). Propojení dat a činností probíhá pomocí typu V, který definuje tabulku FV pro jednu konkrétní činnost.

Logický návrh dat začíná na úrovni typu K, který je identifikován svým názvem. Je použitelný pro definici typů *, I nebo V, které popisují fyzické uložení dat na disku. Typ K se může skládat z libovolného počtu položek typu D—date, N—numeric, C—character, L—logical nebo M—memo nebo i typu K.

Příklad 1:

Budeme definovat datový typ K, který odpovídá adrese:

JMÉNO POLOŽKY	TYP	DÉLKA
ADRESA	K	-

tento typ se skládá z:

název firmy	NAZ_FIR	C	20
ulice	ULICE	C	20
číslo popisné	CISLO	C	4
město	MESTO	C	20
PSC	PSC	N	5
datum zařazení do adresáře	DATUM	D	8
číslo adresáta	CISL_KL	C	5

Ke každé položce je možno si poznamenat poznámku a několik dalších atributů jako validační funkce, inicializační hodnota apod.

Při návrhu fyzické reprezentace dat navrhujeme soubor ADRESÁŘ.

ADRESAR *

z položek

	ADRESA	K	-
případně i indexový soubor	ADRESAR	I	klíčový výraz

3.3. Propojení datové základny a funkčního schématu

Toto propojení se realizuje pomocí datového typu V, který je obrazem tabulky FV v adresáři dat systému GEPRA. Položka typu V se v adresáři dat definuje svým jménem, jménem template, který bude použit po vygenerování koncové činnosti ve funkčním schématu a samozřejmě je možnost i položky typu V okomentovat poznámkou.

Položky podřízené položce typu V mohou být libovolného typu včetně logického typu K. Vztahy mezi jednotlivými položkami typu * (soubor) lze definovat pomocí typu R (relace). Aby byl lépe patrný význam adresáře dat i jeho propojení na funkční schéma, vrátíme se k příkladu 1.

Na základě nových požadavků je nutno doplnit adresu o název státu (STAT C 20) a rozšířit číslo adresáta na CISEL_KL C 6. V adresáři dat opravíme atributy položky CISEL_KL a k položce ADRESA typu K přiřadíme další elementární položku STAT. Na základě těchto změn systém GEPRA přegeneruje struktury všech souborů, kde se vyskytuje položka CISEL_KL nebo položka ADRESA. Dále se přegenerují všechny tabulky FV definované typem V, který obsahuje uvedené položky, a současně se vygeneruje nový programový kód, který odpovídá změnám v datové základně.

3.4. Časové závislosti

Systém GEPRA podporuje analýzu jednotlivých funkcí (činností) i z pohledu času. Tato definice probíhá nad celým funkčním schématem.

Příklad:

Ve funkčním schématu jsou činnosti:

- uzávěrka mezd
- uzávěrka zásob
- uzávěrka účetnictví.

Pomocí možnosti definice časových závislostí lze definovat, že uzávěrce účetnictví musí předcházet uzávěrka mezd a zásob. Systém GEPRA ve vygenerovaném kódu zajistí, že za chodu vygenerovaného programu nebude možné spustit uzávěrku účetnictví dříve než uzávěrku mezd a zásob.

Toto zajištění časových vazeb je realizováno i v případě distribuovaného zpracování na několika výpočetních systémech s přenosem pomocí disket.

3.5. Vsazení projektu na pracovní místo (generování na míru)

Nad uceleným funkčním schématem se definuje, které činnosti je nutno na řešené pracoviště zařadit a které nikoliv. Též se definuje, které přenosy dat budou realizovány z tohoto a do tohoto pracovního místa. Tyto definice si systém GEPRA vkládá do svého adresáře zákazníků, uživatelů vygenerovaných programových systémů, takže je možno je využít opakovaně.

3.6. Nové verze

Na základě vývoje dochází ke změnám ve funkčním schématu. Jsou doplňovány nové činnosti a vyřazovány nepotřebné činnosti, jiné mohou být modifikovány. V datové základně dochází k rozšíření o nové položky, nepoužívané položky jsou vyřazovány ze struktur, jiné mění svůj rozsah nebo i typ. Též časové vazby jsou modifikovány, například doplněním nových uzávěrkových činností. Mění se i počet pracovních míst, na kterých musí být vygenerovaný systém provozován. Toto vše nutí vývojáře k poskytování nových verzí programů, ale i nových struktur datových souborů.

Za tímto účelem si systém GEPRA nejen pamatuje, jaké činnosti je zapotřebí vygenerovat pro to či ono pracoviště, ale současně vygeneruje i program nazývaný VERZÁTOR, který zajistí transformaci uživatelských dat ze staré struktury do struktury nové.

3.7. Integrované vývojové prostředí

Systém GEPRA, jak je patrné z předchozích kapitol, podporuje maximální část životního cyklu software. Na jednom projektu mohou úspěšně spolupracovat programátor s analytikem. A aby nebylo nutné v jakékoliv fázi vývoje opouštět prostředí systému GEPRA, má v sobě zabudovány veškeré funkce pro ladění, včetně spuštění libovolného příkazu v jazyce Fox nebo operačního systému. Tímto je systém doplněn tak, že vytváří integrované vývojové prostředí.

4. Vlastnosti vygenerovaného systému

Systémy vygenerované pomocí systému GEPRA mají několik společných vlastností:

1. Jednotný systém menu roletového typu, aktuálně k dispozici jednořádková poznámka a po stisku klávesy F1 i několikastránkový komentář v kontextu s aktuální položkou menu.

2. Možnost uživatelského definování funkčních klíčů. Na funkční klíče může uživatel definovat spuštění tzv. „horkých činností“.

3. Přístup k systému pouze pomocí hesel. Se systémem může pracovat až 30 různých uživatelů a každý má definovány činnosti, za které je zodpovědný.

4. Systém je zabezpečen archivační nadstavbou, která nutí uživatele po uplynutí zvoleného času k archivaci a v případě havárie k rearchivaci.

Za nejdůležitější je nutno považovat jednotné chování vygenerovaného systému vůči uživatelům, a to i v těch místech, kde spolupracovalo několik autorů.

5. Závěr

Systém GEPRÁ podporuje práci tvůrčího týmu, především u rozsáhlých projektů. Její hlavní význam spočívá v tom, že pokrývá celý životní cyklus software, a to způsobem, který značně urychlí a usnadní celkovou realizaci projektu.

Autor: Ing. Jaroslav Janda
BM – výrobní družstvo
Školní 2235
370 10 České Budějovice
telefon: 038/28542, fax: 038/28634