

POZNATKY Z VÝVOJE TECHNOLOGIE PROGRAMOVÁNÍ

RNDr. Pavel Drbal, CSc.

V roce 1983 byla vytvořena skupina pro koordinaci vývoje technologie programování, která se formovala jako součást Komplexního programu vědecko-technického pokroku členských zemí RVHP pod šifrou 1.1.6. Tato skupina si kladla vysoké cíle, z nichž musela poněkud slevovat, byla však určitě užitečná tím, že zvyšovala vzájemnou informovanost, podporovala určitou součinnost a její činnost vyústila v užitečnou spolupráci zainteresovaných organizací. Také konkurs, který tato skupina uspořádala, přinesl zajímavé poznatky. Domnívám se, že je třeba nějakým způsobem usavrřit a zhodnotit prošlou etapu.

V rámci uvedené skupiny byly v členských zemích vyvinuty systémy TESYS, DOGA a R-technologie (RTK). Prostor tohoto semináře neumožňuje uvést ani stručný popis těchto systémů, ten lze najít v 79. čísle AVT (Aktuality výpočetní techniky), které je v tisku. Ostatně referáty o systému DOGA již na tomto semináři odezněly a referát o RTK i s ukázkami je součástí tohoto semináře.

Uvedené systémy jsou zaměřeny na grafickou reprezentaci programu, byť s různým důrazem. TESYS reprezentuje program tabulkou (políčko = výkonný modul ve smyslu normovaného programování). Moduly lze reprezentovat stromovým grafem ve smyslu Jacksonova strukturogramu (ale rozmístění uzlů určuje programátor příkazy !). DOGA je orientována na strukturogramy (stromový graf s tím, že příkazy a podmínky jsou přiřazeny uzlům), R-technologie používá R-grafy (síť s jedním vstupem a jedním výstupem, příkazy a podmínky jsou přiřazeny hranám grafu).

TESYS se snaží vyhovět všem známým metodám programování, DOGA je programovou podporou metody JSD, R-technologie je zaměřena pouze na základní metodiku - rozklad problému na podproblémy.

Již vznik těchto systémů vypovídá o zajímavé věci. Soustavné práci ve skupině 1.1.6 na technologii se věnovali zástupci šesti států (Bulharsko, Maďarsko, NDR, Polsko, SSSR, ČSSR), programové systémy pro podporu technologie programování vznikly ve třech z nich. V době vzniku (80. léta tohoto století) se země, ve kterých vznikaly technologie programování, vyznačovaly přetrvávajícím systémem direktivního řízení, kdežto zbývající tři země byly v etapě přechodu na tržní mechanismus (Bulharsko, Maďarsko, Polsko). Také intenzita účasti na práci skupiny 1.1.6 odpovídala tomuto rozdělení.

Vyvětlení je jednoduché. V etapě tvorby tržního mechanismu se firmy soustřeďují na projekty s bezprostředním ziskem. V této etapě jsou faktory s rychlejším ovlivněním produktivity práce než změna technologie programování, a to :

- zvýšení intenzity lidské práce (obecně vzato, tj. včetně zlepšení organizace práce)
- modernizace technických zařízení, tj. modernější počítače.

Prostředky pro podporu technologie programování mají velký index organického složení kapitálu, tj. doba jejich návratnosti je dlouhá. Z toho plynou dva pravděpodobné důsledky :

- Technologie programování patří do infrastruktury hospodářského systému.
- Vzhledem k obecnému trendu přechodu na tržní hospodářství lze v nejbližších letech očekávat snížení zájmu o technologie programování.

V praxi systémy TBSYS a DOGA neměly rozhodující úspěch, systém R-technologie lze označit za bojující o úspěch. Tím nechci nijak snižovat význam těchto systémů, ale snažím se o objektivní posouzení těchto systémů.

Nejsou mi známy počty programátorů v jednotlivých zemích, ale vzhledem k počtu obyvatel připadá jedna instalace přibližně na půl miliónu obyvatel. Pro srovnání počet instalací editačního systému WORDSTAR v našem institutu je větší než počet

instalací zmíněných systémů v republice.

V tomto srovnání nejlépe dopadá R-technologie. Hlavním důvodem je to, že R-technologie je již od začátku interaktivní, kdežto ostatní systémy pracují v dávkovém režimu. Ten rozdíl je dán předvídatelností v době vzniku systémů a tím, že vývoj technologického systému je záležitost dlouhodobá.

Dalším důležitým důvodem je integrita programátorského prostředí. Systém, ve kterém zapisují program pomocí grafických prostředků a ladím jej pomocí výpisů paměti, je neakceptovatelný.

Zavedení systémů typu Turbo jsou dány minimálními požadavky na nové systémy. Je tedy bezpodmínečně nutné, aby technologický systém umožňoval ladění programu ve stejném prostředí jako jeho psaní. Společnost se řídí zákony, které popsal A. Hurley v knize *Brave New World*, totiž, že nový výrobek musí být funkčně dokonalejší než všechny dosavadní výrobky.

V rozboru příčin úspěšnosti a neúspěšnosti lze pokračovat, zmíním se však pouze o dvou rysech.

Systém TSEYS je z hlediska uživatele spojen z několika různými jazyky. Jsou tam příkazy pro normované programování, příkazy pro Jacksonův styl programování, příkazy pro rozlišování uzelů strukturogramu, příkazy pro definici makr. Takové šíře příkazů sřejmě porušuje zásadu integrity a snižuje pohodlí uživatele.

Systém DODÁ byl navržen pro grafickou reprezentaci programu na papíře, jeho převod do interaktivního režimu je spojen s dosti velkou obtíží: jak reprezentovat strukturogram na obrazovce. Při porovnání strukturogramu s R-grafy je zřejmé, že jsou R-grafy výhodnější při zobrazování na obrazovce.

Celkově lze říci, že spolupráce zmíněných vešl v oblasti technologie programování vyústila ve spolupráci na rozvoji R-technologie.

Lze usoudit, že dochází k určitému zlomu v metodice programování. Pregnantně to vyjádřil jeden zástupce softwarové firmy: "Teď se programuje jen v Americe, v Evropě se pouze dělají aplikace." Tento zlom se vyjadřuje písmeny CASE (Computer Aided Software Engineering) a označuje uživatelsky přítulnou tvorbu programů. Pokusím se, podle dostupných informací, charakterizovat nové rysy.

Nejzávažnějším rysem je zvýšení úrovně automatizace. Předmětem již nejsou jednotlivé programy, ale jejich soustavy, tj. také datová komunikace mezi programy. To vše je navrhováno a vytvářeno jako jeden celek.

S tím je spojen další rys, totiž, že vytvářené programy nejsou určeny pro určité datové prostředí (pro určitý systém manipulace s daty), ale že se toto prostředí vytváří současně s programy. Totéž tvrzení lze vyložit s opačným důrazem, že centrem návrhu je systém manipulace s daty, kolem kterého se generují programy.

Třetím rysem je přítulnost, která se projevuje dvěma směry. Jednak to je vedení koncového uživatele pomocí menu, takže ten nemusí nic vědět nejen o programování, ale ani o počítači vůbec. Druhý směr je zajímavější, je to přítulnost tvorby aplikace. Je to snaha, aby člověk, který vytváří aplikaci pro určitý podnik (říkejme mu konstruktér aplikace) mohl vytvářet aplikaci bez znalosti programování. Cílem je, aby konstruktér aplikace pouze definoval data a jejich vlastnosti (z nichž se odvodí požadované funkce programů), položky menu a vysvětlující texty.

Dosažení takové přítulnosti v obecné poloze nelze považovat za rozumně možné. Musíme si však uvědomit, že to, čemu říkáme "programy" je nutno chápat nikoli obecně, ale v oblasti dosud nazývané "hromadné zpracování dat", tj. v oblasti manipulace s daty, kde lze očekávat statisíce a milióny uživatelů. Proti tomuto lukrativnímu trhu ostatní použití programů dosti zanikají - snad kromě řízení technologických procesů, jehož důležitost bude pravděpodobně vzrůstat.

Je nutno poznamenat, že programování nemíří, ale přesouvá se jinam. Značné přítulnosti se dosahuje ve třech úrovních. Základní systém má dosti složitý ovládací jazyk, kterým se definuje generátor aplikací. Teprve tento generátor poskytuje zmiňovanou přítulnost. Celý postup lze tedy popsat takto :

Základní systém

s nepřítulným ovládacím jazykem. V tomto jazyce vytvoří "konstruktor oboru" generátor aplikací, který umožňuje definovat aplikaci pouze se znalostí oboru, bez znalosti programování.

Generátor aplikací

pomocí kterého "konstruktor aplikace" formuluje konkrétní zadání. Vstupem generátoru jsou definice dat a jejich vlastnosti. Z vlastností dat odvodí generátor operace s daty a vygeneruje aplikaci, což je jedna nebo několik databází s příslušnými programy.

Aplikace

je určena pro neprogramátora, který je odborníkem ve své profesi a který se nechá tupě vést jídelníčky.

Samozřejmě vznikají otázky, co umožňuje tento přístup a jak se odrazí na naši profesi. Když se tak podíváme zpátky na to, co je vlastně programování, lze je charakterizovat jako boj o rychlost výpočtu a místo v paměti. Výše načrtnutá perspektiva vychází z toho, že potřebná data jsou k dispozici, a to s přímým výběrem (resp. se zanedbatelnou vybavovací dobou). Je to umožněno pouze tím, že současné počítače dosahují velmi vysokých parametrů - a to i v možnosti přístupu k počítačům. Trochu pejorativně lze říci, že naše "sálové" počítače jsou předstíženy ve všech parametrech počítači, umístovanými po kancelářích.

To je jedna stránka věci, přítulnost degraduje výkonnost; tedy zajištění přítulnosti je možné jen velkým výkonem. Druhá stránka věci je otázka automatizace programování. Jakákoliv automatizace má smysl tehdy, jestliže se jedná o masovou výro-

bu. Automatizování programování se často rozumělo automatizace procedurového programování (jiné jsme neznali). Ovšem co automatizovat, tvorba základních systémů? Ty jsou unikátní a jsou vytvářeny profesionálními programátory, to není příliš lukrativní trh. Automatizovat tvorbu generátorů aplikací? To je příliš nestálé prostředí a také neposkytuje dostatečně širokou oblast. Tvorba aplikací již automatizována je. S tím také souvisí otázka, co učít při výuce programování. Vzniká mnoho otázek, na které si netroufám seriózně odpovědět. Je však jasná jedna základní věc. Tvorba nějakého základního systému (ve smyslu výše uvedeném) je velmi finančně náročná. V podmínkách trhu si to může dovořit jen firma, která je kapitálově silná a má takové jméno, které zaručuje zájem trhu. V tom smyslu je třeba rozumět ovcu "programuje se jen v Americe".

Závěr tohoto článku vyznívá pesimisticky a pesimistická je představa, že v ČSR se přestanou programovat aplikace s masovým využitím a že vrstva programátorů se změní v překupníky cizích programů, ale vývoj naznačuje, že je to docela reálná představa. Ovšem tato "katastrofa" nastoupí ihned, nejdříve musí vzniknout potřeba rychlých informací (ve stávajícím byrokratickém stylu řízení byly rychlé informace přepychem a ne potřebou) a pak musí mít firmy peníze, aby si mohly příslušné systémy zaplatit (a to není úplně laciná záležitost).

Literatura :

- /1/ Drbal, Jilková, Maňková : Metody a technologie strukturovaného programování I a II. Skripta VŠE, Praha 1987
- /2/ Velhický : Technologie programování, Kyjev 1984
- /3/ Technologisches System zur Softwareentwicklung, VEB Robotron, Drážďany 1987
- /4/ Huxley : Brave New World, Penguin Books Ltd., Harmondsworth 1967

Anotace

V článku se srovnávají prostředky programové podpory technologií programování, které byly vytvořeny v rámci RVHP. Na základě zkušeností s jejich použitím jsou odhadnuty trendy dalšího rozvoje technologie programování.