

RNDr. Zdeněk Botek, UJEP Brno

## 1. Úvod

O úspěšnosti hromadného nasazení levné mikropočítačové techniky (v nadávne době ve světě a v blízké budoucnosti u nás) rozhoduje také vhodně navržený způsob komunikace uživatele s počítačem. Jedním z faktorů ovlivňujících kvalitu počítačové komunikace je míra využití vizuálních komunikačních prostředků. Není proto překvapením, že naprostá většina osobních mikropočítačů přivezených ze zahraničí (Sinclair, Commodore, ...) je u nás využívána na hraních, které se vyznačují jednoduchým ovládním a bohatým využitím vizuálních vlastností monitoru, jen malé procento uživatelů se pokusí přimět počítač sekvenčním zápisem příkazů programovacího jazyka k tomu, aby svou činností sloužil jejich profesionálním nebo soukromým požadavkům. Tato skutečnost napovídá, že je třeba hodnotit vhodnost v současnosti běžných programovacích prostředků na všech fázích komunikace s počítačem při vývoji, ladění i zpracování programů a posoudit podíl pro lidskou mysl přirozených vizuálních prostředků na současném softwaru.

## 2. Co jsou to vizuální prostředky?

Vizuální prostředky představují novou dimenzi, nový kvalitativní krok ve způsobu komunikace člověka s počítačem. Doplnují a často i nahrazují dnes běžně používané zpracování textů zpracováním tvarů a podnětů. Tento způsob komunikace je lidskému smyslovému systému přirozenější, neboť textové zpracování jako móda tohoto tisíciletí je mnohem náročnější než zpracování reálné skutečnosti (nebo alespoň jejího věrného obrazu).

Vizuálními prostředky rozumíme především dvojrozměrné grafické zobrazení informací na obrazovce monitoru, případně na obrazovce simulované zobrazení trojrozměrných prostorových útvarů. U současných počítačů běžně používané barevné monitory dávají svou barevnou stupnicí další možnosti k diferenciaci informací, různé barvy objektů spolu s dočasným barevným případně světelným zdůrazněním určité jednotky jsou ve vizuálních systémech často a úspěšně

využívány. Vizualní komunikace řízená obvykle pohybem řídicí páky nebo pohyblivé kyši bývá doplněna zvukovými signály, které informují především o úspěšné či neúspěšné reakci uživatele na požadavek počítače.

### 3. Vizualní komunikace versus text

Je obecně známo, že lidský intelekt je silně vizuálně orientován a že lidé získávají informace významně vyšší měrou při analýze grafických vztahů i ve složitějších obrázcích než při zkoumání textových informací. Lidské oko poskytuje okamžitý přístup k libovolné části obrázku a umožňuje pohled podrobný i přehledný. Náš mozek rychle vyhodnotí známé skutečnosti a zaměří pozornost vnímání na problematické části. Vizualní systém v tomto okamžiku jistě umožní podrobnější zobrazení zkoumaného výseku.

Vizualní prostředky umožňují nové pohledy na vnější i vnitřní stránku všech softwarových komponent. Jazyk dvou- a třírozměrných obrázků je bohatší než text, zakódování každé informační jednotky je jednoduché a přehledné s výsledným snížením dekodovacího času. Vhodně lze využít specifickou vlastnost grafiky, její přirozenou interakci. Používané objekty není třeba přesně identifikovat jejich jménem, je možné na ně ukázat. Běžné programovací jazyky odkazují na objekty nepřímou, jejich jménem. Vizualní programovací prostředky určují objekty přímým, aktuálním odkazem.

Zobrazení objektů reálného světa vizualními prostředky slouží k ilustraci abstraktních myšlenek, pro usnadnění následných myšlenkových pochodů. Dobré ilustrace napomáhají nejen pro rychlejší komunikaci specialistů, ale jsou také velmi užitečnou pomůckou pro začátečníky. Dobrý obrázek, dobrá ilustrace poskytuje vhodnou metaforu, užitím textu si musí čtenář vytvořit vlastní mentální představu. Např. nejlepším vizualním pomůckou pro výuku počítačového programování je sám konkrétní model počítače.

### 4. Vizualizace programovacích prostředků

Předchozí části oběrně uváděly novou techniku komunikace s počítačem především z toho důvodu, že jsme běžně některé prvky vizualní komunikace využívali i při textovém zpracování, ale přechod na "obrázkové" programování se nám zdá být z různých důvodů nereálný pro současnost i pro blízkou budoucnost. Pomineme

bariéry současného hardwaru u nás a naznačíme oblasti, ve kterých je ve světě již nyní princip vizuálního programování běžně využíván případně právě zaváděn.

První, nejobecnější oblastí je vizualizace operačního systému a všech jeho komponent. Podstatné informace, které o operačním systému musí mít různé skupiny uživatelů, jsou značně rozdílné. Jinou úroveň komunikace má programátor systémový a aplikační, jinou technik, jinou operátorka, která pouze zadává data. Vizuální operační systém umožní odpovídající úroveň komunikace, nevyžaduje znalost celého systému od všech a především nevyžaduje znalost přesné syntaxe jednotlivých akcí, neboť uživatel "vede" na úrovni odpovídající jeho znalostem. Systém nabídne uživateli např. formou menu možné aktivity, uživatel "najeď" na požadovaný úkon pohyblivým ukazovátkem (myší). Každá komponenta takového systému je také vizuálně orientována. Např. editor zobrazuje část zápisu algoritmu, kterou chceme upravovat (což je již dnes běžné), program pro práci se soubory zobrazí graficky jednotlivé soubory našeho adresáře a jejich charakteristiky, sledovací program v určených místech zpracování algoritmu informuje o hodnotách kritických datových struktur i stavu zpracování úlohy. V každé takové vizuálně řízené komunikaci je opět další možný postup vhodně vizuálně indikován, není nutné hledat potřebné příkazy ve vlastní paměti či manuálech.

Schopnosti vyšších programovacích jazyků pro efektivní vyjádření programátorské aktivity jsou téměř vyčerpány, nové typy jazyků přinášejí pouze nepatrné modifikace. Hledáme nové prostředky, které podstatně zvýší naši schopnost vyjádřit své požadavky na počítač. Jsou dvě reálné možnosti, první z nich je komunikace v přirozeném jazyku, druhou je jazyk grafický. Tato druhá varianta chce dát uživateli lehce pochopitelné fyzické univerzum a prostředky pro práci s ním ve dvourozměrném prostoru. Grafickým programovacím jazykem tedy rozumíme přesně definované uspořádání grafických symbolů ve dvoudimenzionální reprezentaci, přičemž je definováno zobrazení významu symbolů do syntakticky korektní formální struktury. Tato myšlenka není nová, naopak, vrací se k původnímu, přirozenému zápisu algoritmů grafickým způsobem, i v současné době si téměř každý programátor zapisuje své myšlenkové postupy při tvorbě programu ve formě grafické. Grafický programovací

jazyk bere více ohled na uživatele, na jeho schopnosti, potřeby, snaží se přenést na obrazovku více abstraktních vlastností programu.

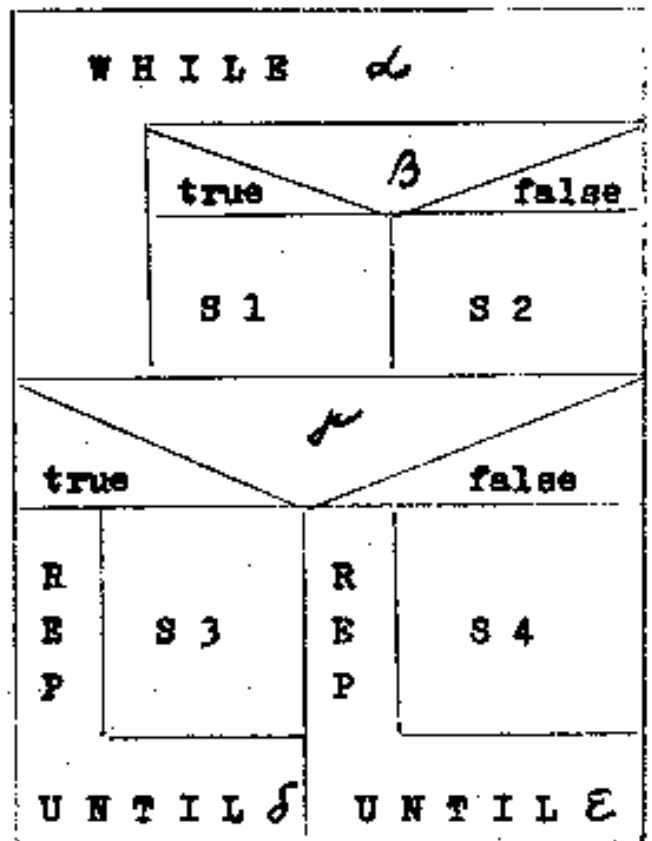
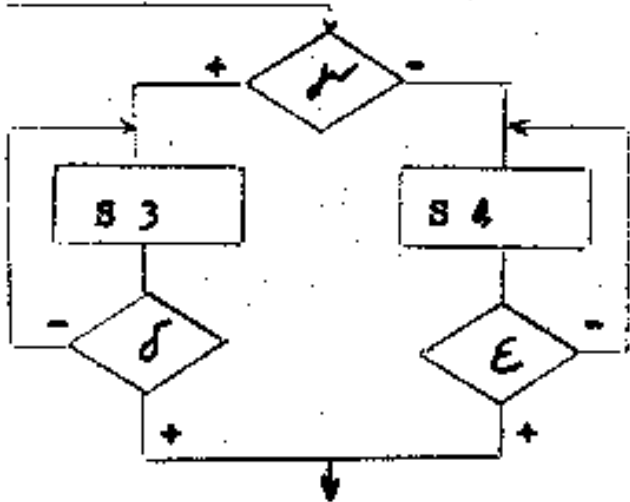
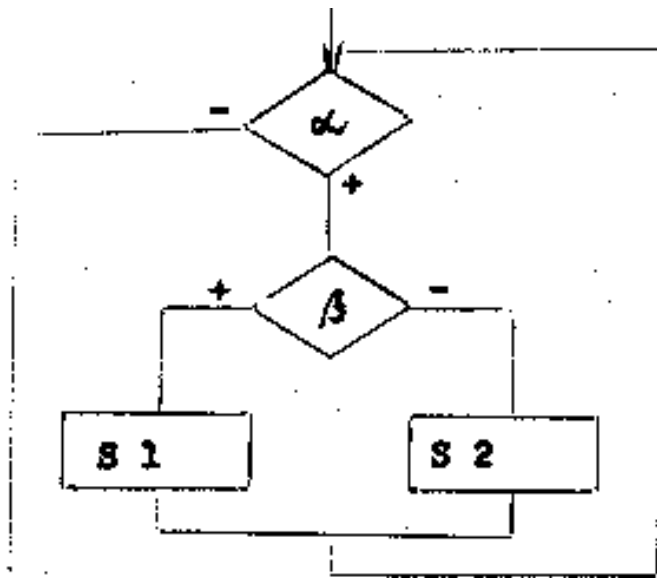
Další vhodnou oblastí pro aplikaci vizuálních prostředků je oblast testování. Program při své realizaci vykonává dynamické akce, které jsme popsali ve zdrojovém textu, tato realizace je hlavním cílem našeho programátorského úsilí, ale nemusí vždy přesně korespondovat s naším záměrem. Cílem nasazení vizuálních prostředků je zveřejnění průběhu zpracování programu, tzv. oživení (animation) algoritmu. Na obrazovce se vhodným způsobem zobrazí datové struktury i program, oživení se realizuje dynamickou změnou hodnot dat ve shodě s činností programu a průběžnou světelnou indikací zpracovávaného příkazu. Realizovat uvedený záměr je možné třemi způsoby. Uživatelsky orientovaný přístup nechá volat nový obraz zpracování programu tehdy, když nastane nějaká zajímavá změna. Programátor v tomto případě píše paralelně dva programy a není jisté, zda si tyto dva vždy odpovídají. V tomto případě je aperát oživovací grafiky skryt do uživatelského programu. Druhý možný způsob realizace "oživovacího" systému spočívá v určení speciálních bodů v systémových programech realizujících běh programu, které zajistí obnovu na obrazovce předváděného stavu zpracování programu a dat. Grafická ilustrace je potom pouze boční efekt skutečného běhu programu, změna zdrojového programu nemusí brát na problematiku ožívání zřetel. Třetí varianta implementace systému oživení algoritmu využívá separátní, nezávislý monitorovací proces, který si sam vyťahuje požadované informace v době běhu programu.

## 5. Implementace vizuálních systémů

Typickým příkladem vizuální komunikace je práce s mikropočítačem Macintosh firmy Apple. Tento mikropočítač se 128 kB RAM, 64kB ROM, jednobarevným monitorem a disketami o kapacitě 400 kB je plně ovládan pohybem myši, na obrazovce se mimo názvy aktuálních činností a stavu systému nabízí široká škála parametrů pro jednotlivé činnosti. Např. systém pro zpracování textů MacWrite má k dispozici devět druhů písma, systém MacPaint pro malování obrázků a projektování útvarů umí různé tvary volitelných velikostí, pro vyplnění ploch je k dispozici 40 variant barevného pokrytí.

Práce se systémem je vskutku velká pohodlná, podle prospektu firmy označují již nyní odborníci právě tento systém jako budoucí standard pro osobní počítače. O úspěchu právě tohoto typu mikro-počítače svědčí prodej 120 000 kusů za necelé dva roky výroby.

Vývojový diagram je nejnámějším a nejrozšířenějším grafickým nástrojem pro zápis algoritmů. Původně vznikl pro assembler, neboť mohl popisovat také nestrukturované předání řízení a vzhledem k jednoduchosti nebylo nutné zabývat se datovými strukturami. Pro současný trend strukturovaného programování (ukáznuté řídicí struktury, netriviální datové struktury) je jazyk vývojových diagramů jako obecný grafický prostředek již zastaralý. Dosud se však užívá především z důvodů nedostatku jiných, systematických grafických nástrojů. V r. 1973 uvedli I. Nassi a B. Schneiderman grafický jazyk tzv. struktogramů. Jejich způsob zápisu nepovoluje libovolné předání řízení, zápis jednotlivých řídicích struktur je rigoróznější, ale přehlednost výsledného zápisu algoritmu se oproti vývojovým diagramům příliš nezvýšila. Bohužel není opět možné vhodným způsobem zapsat definici a změny hodnot datových struktur. Zápis algoritmů pomocí struktogramů je dnes běžně užíván v učebnicích i v přímé výuce základů programování na VŠ v NDR. Jednou z nesporných výhod struktogramů je jednoduchost používaných grafických značek, což předurčuje právě tento systém pro přímé zpracování grafického zápisu algoritmu počítačem. Firma AID (Automatisierung, Informatik, Datatechnik) nabízí softwarový systém označovaný X-TOOLS, jehož složky umožní editovat grafický zápis algoritmu na obrazovce terminálu, jeho syntaktickou kontrolu, transformaci do zdrojového textu ve vyšším programovacím jazyku, případně naopak převod zdrojového textu ve vyšším programovacím jazyku do struktogramů. Tato nabídka není pouze prototypem v omezeném rozsahu, již dnes lze převádět struktogramy do Pascalu, Basicu, Fortranu-77, C-jazyka, PL/M a Cobolu. Systém tedy neřeší převod grafického zápisu algoritmu přímo do kódu počítače, ale využívá vyšší program, jazyk na zápis mezivýsledku. Systém lze provozovat pod operačním systémem běžných minipočítačů a mikropočítačů firem DEC, IBM, Intel a Siemens. Obrázky na této straně srovnávají zápis stejného algoritmu ve vývojovém diagramu, struktogramu a jazyku Pascal.



Begin while  $\alpha$

```

do if  $\beta$  then S1;
   else S2;
if  $\gamma$  then repeat S3;
   until  $\delta$ 
else repeat S4;
   until  $\epsilon$ 

```

end.

Na závěr uvedeme vizuální systém vyvinutý na univerzitě ve Washingtonu, pracovišti, které se systematicky problematikou vizuálního programování zabývá. Programovací metodologický systém PICT chce lidem umožnit využít jejich přirozenou inteligenci v programování. Systém představuje počáteční stádium nového metodologického směru, je na rozhraní několika disciplín, např. návrh a implementace programovacího systému včetně jazyka, ergonomie a psychologie zabývající se komunikací člověka s počítačem a další.

Implementovaný systém chce poskytnout nový typ algoritmickeho programovacího prostředku, v němž hraje ústřední roli počítačová grafika. Uživatel nepíše své programy s použitím písmen, číslic a oddělovacích znamének, klávesnice je při činnosti systému nepotřebná. Místo toho sedí před barevnou grafickou obrazovkou a kreslí programy s použitím vhodného vstupního zařízení, např. řídicí páky, která určuje pohyb kurzoru na obrazovce. Grafika, barvy, zvuk a ožívování jsou použity tak, aby se uživatelské programy jevíly co nejvíce multidimenzionální a konkrétní. S výjimkou číselných a krátkých pomocných zpráv uživatelé plně spoléhají na netextové symboly. Jména programů, datové struktury, proměnné i operace jsou reprezentovány ikonami různých druhů, zatímco řídicí struktury jsou na obrazovce vyznačeny barevnými cestami. Systém PICT poskytuje všechny nástroje, které uživatel potřebuje k tvorbě, zpracování i úpravám svých programů. Uživatel komunikuje se systémem ukazováním na položky v hierarchickém menu, systém reaguje vhodnou změnou obrazovky nebo zprávou v případě uživatelské chyby. Syntaxe uživatelského programu je kontrolována nepřetržitě během programovacího procesu, syntaktické nedostatky se indikují i když je program pouze částečně sestaven.

Prototyp systému PICT byl implementován v Pascalu na počítači VAX 11/780 pod OS VMS. Obrazovka monitoru je schématicky rozdělena na pět částí - označení programu, označení segmentu, zobrazení datových struktur nebo pomocné informace, vstupní menu, uživatelský program. Výsledné programy vypadají jako vývojové diagramey, jednotlivé uzly tvoří obrázky reprezentující různé typy proměnných, konstant, operací, následnost jednotlivých objektů a jejich

vztahy jsou vyjádřeny různobarevnými spojnicemi. Uživatel systému PICT nejdříve z nabídky všech dostupných objektů zařadí do prostoru programu všechny potřebné, v další fázi je spojí řídicími hranami. Použité objekty mohou představovat složitější akce a je tedy možné jejich strukturaci dále specifikovat. Takto sestavený program lze po spuštění sledovat, průběžný stav datových struktur je zobrazován.

Systém byl testován na 60 studentech a 60 absolventech univerzity. Následující reakce charakterizují názory testujících:

- Popisání vývojového diagramu je už popisem programu, k zápisu programu není třeba žádný "počítačový jazyk".
- Síla systému je ve fázi testování. Při sledování běhu programu můžeme rychle přijít na problematická místa.
- Na omezené ploše obrazovky nelze vyjádřit složitější algoritmy.
- Repertoár operací je příliš omezený.

Celkově přijali vizuální systém velmi příznivě, kdežto absolventi většinou svou vcelku příznivou reakci doprovázeli komentářem, že by svůj oblíbený textový jazyk ze vizuálního systému PICT neměnili.

## 6. Závěr

Vizualizace programových prostředků představuje jeden z možných směrů zvyšování efektivity programátorské činnosti a především usnadnění počítačové komunikace pro velmi početný okruh zájemců o počítače. V článku jsme představili perspektivní oblasti nasazení vizuálních prostředků, se kterými se u předních výrobců mikropočítačů již dnes můžeme setkat. Naše pracoviště se v rámci úkolu "Problematika správnosti programů" začíná zabývat metodou ožívání algoritmů, v pedagogickém procesu zkoumáme grafické metody pro vyjádření algoritmů.

## L i t e r a t u r a :

1. Reeder, G.: A Survey of Current Graphical Programming Techniques, Computer, vol. 18, No.6, August 1985, str.11-25
2. Glinert, E.P., Tanimoto, S.L.: PICT: An Interactive Graphical Programming Environment, Computer, vol.17, No.11, str.7-25
3. Brown, M.H., Sedgewick, R.: A System for Algorithm Animation, ACM Computer Graphics, vol.18, No.3, July 1984, str.177-186