

PROGRAMOVACÍ SYSTÉMY PRO ANIMACI ALGORITMŮ

RNDr. Zdeněk Botek, CSc., UJEP Brno

1. Úvod

Uživatele aplikačních programů zajímají především výsledky poskytované těmito programy. Programátora, který má ručit za svou práci, musí zajímat mimo výsledky také chování programu během zpracování. Program při své realizaci vykonává dynamickou akci, která je staticky popsána ve zdrojovém textu. Na správný průběh realizace programu usuzuje programátor z výsledků kontrolních chodů, neznáme zatím prakticky použitelné nástroje pro dokazování správnosti programů. Systémy pro animaci algoritmů přispívají k míře poznání o průběhu zpracování programu tím, že nám "zveřejní" zpracování jednotlivých příkazů programu spolu s dynamickou ilustrací změn hodnot použitých proměnných.

Oblast animačních systémů je součástí nové skupiny perspektivních programátorských aktivit, které ve zvýšené míře využívají vizuální vstupní i výstupní informace pro názornější, pohodlnější i rychlejší komunikaci s počítačem. Běžně používané zpracování textů je doplňováno a v některých aplikacích až nahrazováno zpracováním tvarů a podnětů. Celá problematika vizuálního programování obsahuje v současné době tři oblasti:

a/ vizualizace operačního systému - uživatel nemusí znát přesnou syntaxi jednotlivých příkazů, z nabídky zobrazené na monitoru si vhodným vstupním zařízením vybere požadovaný úkon, nabídka svou formou i obsahem odpovídá úrovni znalostí uživatele i momentálnímu vývoji komunikace. Informace zobrazené na monitoru tvarem "napovídají" svůj obsah, bohatě jsou využívána tzv. okna pro možnost současného rozpracování a sledování několika problémů. Je známo již několik velmi úspěšných vizuálních operačních systémů, jako standard se v blízké budoucnosti počítá se systémem mikropočítače MacIntosh firmy Apple.

- b/ grafické programovací jazyky - způsob zápisu algoritmu pro jeho začínání počítači se ustálil na textovém zápisu využívajícím určitý typ programovacího jazyka. Bylo to tím, že právě tato forma je pro uživatele nejvhodnější nebo neopak tím, že z technických příčin nebyla jiná možnost? Grafické programovací jazyky umožňují vytvářet algoritmy přirozeným způsobem, grafický zápis je vlastní při přípravě algoritmu začátečníkům i profesionálům. Problematika se dále dělí na grafické jazyky vycházející ze známých algoritmických jazyků a ikonické systémy, které pro návrh a zápis algoritmu používají postupy zcela odlišné.
- c/ animační systémy - systémy zobrazující průběžně zpracovávané příkazy i změny hodnot proměnných. Zatímco oblast a/ závisí především na technických možnostech počítače a oblast b/ je doposud hodnocena velmi rozpačitě, animační systémy sklízí při všech aplikacích úspěch.

2. Animace algoritmů

Dynamická ilustrace programu, která vznikne transformací textového zápisu programu na grafickou reprezentaci, se nazývá animace /oživení/ algoritmu. Cílem animace je pomocí grafických snímků zveřejnit průběh zpracování programu, co nejvýrazněji vysvětlit, jak daný program pracuje. Dynamická ilustrace zpracování programu pomáhá uživateli dosáhnout obsáhlého a přesnějšího porozumění chování jeho vlastního programu nebo programu jím zkoumaného.

Aplikační oblasti pro animační systémy lze snadno odhadnout: základní výuka algoritmů; návrh a analýza algoritmů; testování, ladění a optimalizace kódu atd. V rámci základního kurzu algoritmizace může animace sloužit jednak k pochopení obecně známých algoritmů a datových struktur, jednak k "oživení" samostatně zpracovávaných studentských programů.

Při testování zkoumáme program s cílem najít chybu. Animace umožní testování na principu "bílé schránky" /white box/ - testovaný objekt nám animace představuje, zveřejňuje. Potřebné informace jsou nabízeny, není nutné

na průběh zpracování pouze usuzovat ze statického zápisu programu a výsledných hodnot. Vybíráme-li testovací data tak, aby byly pokryty všechny potenciálně možné cesty programem, pak právě animace nás snadno přesvědčí o průběhu cesty pro daná vstupní data a po zaznamenání dalších zpracování o průběhu všech možných cest.

Návrh animačního systému spočívá v řešení tří problémových oblastí:

- a/ technické problémy - volba takového počítače, který má vhodné vstupní i výstupní zařízení a programové vybavení umožňující manipulaci s grafickou informací. Je třeba hodnotit zobrazovací schopnost obrazovky, kvalitu grafického módu, způsob řízení kurzoru na obrazovce, využití barev případně pouze inverze, užití zvukových signálů.
- b/ požadavky uživatelské - jsou specifikovány předpokládaným okruhem uživatelů a tím také předpokládanou aplikační oblastí. Systém může být zaměřen na objasnění činnosti známých algoritmů, sledování běhu programu a chování datových struktur programů vlastních, testování a ladění programů případně ke sledování běhu programů uživateli neznámých.
- c/ pohled filosofický a psychologický - jeho úkolem je zajistit co nejsrozumitelnější zobrazení informací. Je třeba posuzovat přehlednost zobrazení, zhodnotit únosnou míru současně poskytovaných informací i požadavky na estetiku vyjádření.

O úspěšnosti animačního systému do značné míry rozhoduje způsob řešení prioritního problému - reprezentace datových struktur. Nevhodná reprezentace může vést k matení uživatele, naopak vhodně zvolená ilustrace výrazně ovlivní míru získané informace. Datové struktury mohou být reprezentovány různými druhy diagramů, obrázků, skic doplněnými případně textem. Je přirozené reprezentovat datovou strukturu stejným objektem, jaký je používán v běžné praxi. To znamená, že zásobník bude zobrazen jako sloupec vertikální naproti tomu fronta nebo řetězec bude řada horizontální, kruhový buffer jako kruh apod. Volba reprezentace je závislá také

na sémantice konkrétního programu.

S reprezentací datových struktur úzce souvisí problém indikace konkrétní změny, což je pro dynamickou ilustraci běhu programu opět otázka závažná. Obvykle se používá některá z následujících technik:

- nová hodnota přepíše hodnotu původní, realizace změny je barevně nebo inverzně zvýrazněna
- nad strukturou jsou umístěny indikátory, které pohybem ve směru vertikálním nebo horizontálním ukazují na zajímavý prvek /např. vrchol zásobníku/.
- zrušení původní struktury a vytvoření nového stavu nebo stejným způsobem práce s jedním prvkem

3. Implementace animačních systémů

Po zvážení výše uvedených aspektů můžeme přistoupit k implementaci animačního systému. Dosavadní praxe ukazuje na dva základní směry implementace:

- a/ programátor zařadí do programu také příkazy realizující animační akce, píše tedy současně zápis algoritmu a zápis animace. Není samozřejmě jisté, zda realizovaná animace plně odpovídá zpracování programu.
- b/ animační systém realizuje ilustraci běhu programu nezávisle na uživateli, systém zařazuje na zajímavá místa programu volání animačních akcí měnících zobrazené informace o průběhu zpracování. Animace programu je potom pouze vedlejší efekt skutečného běhu programu, změna zdrojového textu nemusí brát ohled na problém animace.

Přístup k implementaci popsaný pod bodem a/ je vhodný především pro oblast studia vlastností algoritmů. Programátor píšící animační program zná důkladně vlastnosti zapsaného algoritmu a jeho cílem je vytvořit takovou animaci, která vlastnosti algoritmu jemu známé co nejvhodnějším způsobem představí studentům. Mimo objasnění operací nad datovými strukturami musí programátor zvážit, jaké doplňující informace poskytne animace pro dokreslení činnosti algoritmu /např. počet provedených výměn, úroveň vnoření cyklů atd./. Hodnotné animační programy dokonce povolují aktivní přístup studentů, je možné přerušit animaci atd.

Způsob využití těchto možností pak svědčí o pochopení podstaty algoritmu. Takto realizovaný animační systém je běžně využíván pro pochopení základních třídících algoritmů.

Implementace systému uvedeného pod bodem b/ předpokládá oddělení zápisu uživatelského programu od animačního programu. Pod tímto označením máme nyní na mysli program nebo programový systém, který zpracovává uživatelský program tak, že bude v době běhu současně realizována jeho animace. Z toho plyne také využití uvedeného typu animačních systémů - studium vlastností vlastních programů pro začátečníky, testování a ladění programů pro pokročilé a také návrh a analýza programů pro programátory. Nezávislost uživatelského programu na průběhu animace není obvykle úplná, uživatel může volit některé parametry animace. Rozsah jeho volby se pohybuje od jednoduchých pokynů /které proměnné a kterou část programu sledovat, určení rychlosti animace/ až po požadavky silnější /volba zobrazení datové struktury, kontrola rozsahu a dalších vlastností proměnných/.

4. Systém pro animaci pascalovských programů

Na katedře aplikované matematiky PTF UJEP je vyvíjen systém pro animaci programů zapsaných v jazyku Pascal. Vzhledem k zaměření činnosti pracoviště je systém určen především pro zvýšení efektivity výuky v základním kurzu programování a dále se počítá s využitím systému při ladění ročníkových a diplomových prací. Systém je implementován na minipočítači PDP 11/34 pod operačním systémem RSX-11/M, pro vstup a výstup je využíván terminál VT 100. Technické podmínky lze charakterizovat jako vhodné pro animaci typických školních úloh, problém spočívá v malém rozsahu obrazovky pro zachycení dostatečné šířky informací u programů řešících reálné problémy.

Požadavky uživatele na zobrazení informací jsou uspokojovány na základě strategie stanovení sledovaných struktur, ostatní údaje se budou překrývat po využití úplné kapacity zobrazení na monitoru. Animační záznam je na obrazovce rozdělen na část dat, část řídicích příkazů a informační část.

Část dat je v horní polovině obrazovky /řádek 2.-17./ a poskytuje informace o aktuálních hodnotách zvolených proměnných. Současně je možné sledovat identifikátor a hodnotu 13 skalárních proměnných a celkem 100 prvků pole /např. matice 5x10 a 5 vektorů o 10 prvcích nebo matice 10x10 apod./. Část řídicích příkazů představuje v dolní části obrazovky /ř. 20.-24./ pět řádků programu zahrnujících také právě aktivní řádek, tento je zdůrazněn inverzním zobrazením. Příkazy jsou zobrazeny textově, vzhledem k uvedeným prostorovým potížím není zatím možné spojit animaci s grafickou reprezentací programu. Informační část ve dvou řádcích informuje o úrovni vnoření a počtu opakování cyklů, je také uvedena posloupnost čísel řádek programu podle jejich skutečné realizace.

Systém dále umožňuje zastavení animačního záznamu, zjištění a případnou modifikaci libovolné hodnoty proměnné. Implementace systému je založena na instrumentaci zdrojového programu ještě před jeho kompilací, za zajímavé příkazy je vloženo volání procedur obnovujících stav obrazovky.

5. Závěr

Zatím nelze v našich podmínkách doporučit konkrétní animační systém, jehož nasazení poskytuje výše uvedené výhody. S rozvojem mini- a mikropočítačů musíme uvažovat také o aplikacích, které využijí nové technické možnosti pro zjednodušení komunikace s počítačem. Animační systémy umožní využít vizuální kontakt s počítačem pro efektivnější studium vlastností algoritmů.

L i t e r a t u r e :

1. Botek, Z.: Vizuální programovací techniky.
Programování 86, s.15-22, Ostrava 1986
2. Brown, M.H., Sedgewick, R.: A System for Algorithm Animation
ACM Computer Graphics, v.18, No.3, July 1984, s.177-186
3. London, R.L., Duisberg, A.: Animating Programs Using Smalltalk
Computer, vol.18, No.6, August 1985, s.11-25