R E C E P T N A J Í D E L N Í Č E K A N E B
M E T O D I K A D I A L O G U F O R M O U " M E N U "

Ing. Richard Bébr
Výzkumný ústav spojů

David Bowman: Hale, nemáš snad ty sám nějaké neenáze - víš, něco, co by samo mohlo být podnětem k našim obtížím?
HAL 9000: Podívej, Dave, já vím, že to dobře myslíš, že chceš pomoci věci vyřešit. Ale mé zpracování informací je absolutně normální. Přezkoumej mé záznamy, zjistíš, že jsou zcela bez chyb.
David Bowman: Znám tvé pracovní záznamy do detailu, Hale - jenže to nedokazuje, že se právě teď nemůžeš mýlit. Chybu může udělat každý.
HAL 9000: Nechci to nijak zdůraznovat, Dave, ale já chyby dělat nemohu.
David Bowman: No dobře, Hale, chápu, jak se na to díváš. Nechme to prostě být.

(Arthur C. Clarke: 2001 - Vesmírná odysea).

1. Úvod

17 let před konáním semináře "Programování 84" předpokládal A. C. Clarke, že již za 17 let po tomto semináři bude možno diskutovat s počítačem tak, jak předvádí ukázka ze známého filmu.

V nedávno natočeném filmu "Vetřelec" (jehož děj se odehrává několik set let v budoucnosti) však poněkud střízlivěji probíhá komunikace s počítačem na obrazovce systémem výběru z nabídky.

Konečně i na seminářích v Havířově se ještě před 7-8 lety diskutovalo o skvělých možnostech přímé uvolněné debaty s počítačem v blízké době. Řečníci se předháněli v nastiňování smělých perspektiv; i autor těchto řádků podlehl atraktivním ideám a řešil "Malou českou banku dat" (viz sborník 1979).

Leč mezitím byla zpřístupněna technika pro interaktivní provoz a praxe velmi rychle ukázala, že většina uživatelů

- neví, co chce vlastně počítači říkat
- chce komunikaci rychlou, pohodlnou a jednoduchou (a ukázalo se, že dialog přirozeným jazykem je pomalý, nepohodlný a komplikovaný)
- zajímá se o užitečné výsledky a ne o to, jak umným způsobem byly získány.

Tento příspěvek si klade za úkol shrnout různé teoretické zásady i praktické zkušenosti o tom typu komunikace s počítačem, který v současné době přináší dobré výsledky a zdá se být i dostatečně perspektivní. Budeme mu říkat "nabídkový dialog".

2. Interaktivní systémy

2.1 Obecné úvahy:

Celý příspěvek vychází z jedinečné knihy /1/, z níž neustále cituje a vybírá; toto konstatování umožňuje vyhnout se v dalším textu únavným opakovaným odkazům.

Za nejdůležitější poznatek vezmeme základní rozdělení interaktivní komunikace s počítačem:

a) Dialog s iniciativou uživatele:
počítač reaguje na podněty, přání a příkazy uživatele, který musí sám formulovat své požadavky; používá se k tomu různých typů jazyků, mnemonických instrukcí, kódových slov a znaků, speciálních kláves a technických prvků - to vše musí uživatel bezpečně ovládat.

b) Dialog s iniciativou počítače ("nabídkový dialog"):
uživatel reaguje na podněty, přání a příkazy počítače, který podává i jednoduchá vysvětlení; uživatel má znát dobře předmět (věcný obsah) zpracování, z oboru výpočetní techniky je seznámen s několika málo elementárními funkcemi (jak posunout kurzor, co je to "ENTER" a pod.).

V dávných dobách (před několika lety) při prvních úvahách o interaktivních systémech se zdálo, že nabídkový dialog se neuplatní z těchto důvodů:

- je velmi zdlouhavý
- pokrývá pouze omezené, předem stanovené činnosti
- nehodí se na ovládání složitých, kombinovaných úkonů.

Praxe však nalezla řadu fint a triků, kterými lze uvedené nevýhody redukovat na takřka zanedbatelnou míru (jak si dále ukážeme).

Je ovšem nutno přiznat, že z hlediska programátora je velice atraktivní dialog s iniciativou uživatele: hloubaví duchové se přímo vyžívají v návrhu mocného jazyka, psaní kompilátorů a interpreterů se z nelidské dřiny stalo zábavou studentů a není tudíž problémem, celé se to dobře ladí a je to velmi efektní. V podstatě však byla dřina přenesena na uživatele, který

se má naučit opoustu látky, prokousávat se na pohled ohavnými manuály a především musí u terminálu neustále myslet na to jak něco udělat místo aby myslel na to, co vlastně dělá.

Tvorba jazyka a překladačů vyčerpá obvykle všechnu programátorovu potenci, což poškodí jednak jeho partnerku a jednak i budovaný systém (protože všechno ostatní bylo zákonitě odbyto).

Zajímavé je, že popsaný způsob tvorby je bližší programátorům mužského rodu; ženy raději řeší a programují nabídkové dialogy (a to velmi dobře a úspěšně); plyne to patrně z toho, že většina žen hospodaří se všemi druhy potence (včetně tvůrčí) mnohem lépe než muži.

Návrh interaktivního systému s nabídkovým dialogem je velmi obtížný právě pro svůj základní přístup:

- nic nemůžeme přenést na uživatele;
- počítač (tedy program) musí za uživatele i myslet;
- každý řádek a každý znak musí být pečlivě zvážen, každý úkon analyzován; jak říká Mr. Martin/1/: při psaní každé řádky programu musíme mít neustále před očima uživatele.

Navíc takřka neexistují prostředky, které by psaní nabídkových dialogů usnadnily.

A k tomu všemu se ještě ukázalo, že pro tvorbu systému s iniciativou uživatele nemusíme vlastně ani pořádně vědět, co uživatel od systému chce (ať si to řekne systému sám !); při psaní nabídkových dialogů musíme naopak ovládat uživatelovy problémy zcela detailně.

Pokusme se nyní uvésti do takto uvedené ponuré oblasti několik světlých bodů.

2.2 Vybavení systému s nabídkovým dialogem:

a) Oblast určení nabídkového dialogu:

Nejprve konstatujme, že systémy s nabídkovým dialogem bývají specializované, zabývající se určitou konkrétně vymezenou problémovou oblastí. Ideál obecného, ničím nespoutaného a vše umějícího systému zůstává zatím v nedohlednu; klasický představitel věeznalosti brouk Pytlík nedosáhl výrazných úspěchů (nebereme-li ovšem v úvahu oblibu u čtenářů). Komplikované soustavy, kde autor byl nucen použít dialogu s iniciativou uživatele většinou v praxi vyžadují specialistu (rodem programátora), který působí jako mezitvář (cizím slovem interface)

a převádí uživatelovy požadavky do jazyka systému.
Z těchto hledisek je uplatnění nabídkových dialogů v dnešních systémech neobyčejně široké.

b) Technické prostředky:
Ve světě se nosí specializovaná technika, pomáhající dialogu, na příklad:
- světelné pero
- obrazovka citlivá na dotek prstu operátora s interní adresací tohoto doteku
- pomocné klávesy (softkeys) umístěné v rámu obrazovky, jejichž význam a popis stanoví v každém okamžiku program

a podobně.
Poznamenejme, že jde sice o jisté zvýšení uživatelova pohodlí, avšak za neúměrnou cenu snížení univerzálnosti zařízení, což je v našich podmínkách nevýhodné (totéž technické vybavení se často efektivně využívá pro různé programové systémy).
Dále tedy předpokládejme jen běžnou, dostupnou a dostatečně univerzální techniku.

2.3 Uživatel:
Existují zásadně dva typy uživatelů:
- uživatel speciální, který většinu nebo celý pracovní čas tráví u terminálu nebo pro nějž je terminál hlavním pracovním nástrojem
- uživatel obecný, který využívá terminál jako pomocný prostředek v různých časových intervalech nebo pro kterého je terminál jen jednou z řady pracovních pomůcek.

Pozn.: Někdy se osvědčuje silnoproudařské dělení na uživatele znalé, poučené a neznalé.

K tomu konkrétní rady:
- speciální uživatele nezdržujme, snažme se usnadnit jejich rutinu; ačkoliv můžeme počítat s jejich důkladným zaškolením, nechtějme je přetěžovat znalostmi úkony, které se vykonávají zřídka, řešme raději jako pro uživatele obecného;

- u obecného uživatele předpokládejme že
  - je inteligentní
  - nemá čas na hluboké studium a dlouhé školení
  - je dosti netrpělivý
  - není nevlídný (pokud ho systém neprovokuje)
  - chce hodnotné a použitelné výsledky
  - rozumí své profesi a ne počítačům.

2.4 Základní zásady:

Pro tvorbu nabídkových dialogů vykrystalizovala v praxi celá řada zásad, které má autor systému respektovat. V dalším textu citujeme rady ze tří pramenů:

- lit. /1/
- zásady, interně publikované v JZD Slušovice a přednesené v diskusi na semináři "Programování 83"
- zkušenosti a interní pravidla VÚS, použitá při tvorbě rozsáhlých interaktivních systémů (viz kap. 6).

Autor věří, že zveřejnění těchto zásad ve sborníku vyvolá diskusi, ve které se objeví i nové, dosud nezpracované náměty.

Z obecného pohledu platí:

- uživatel zná aplikaci, ne výpočetní techniku;
- systém musí být odolný vůči nevhodné obsluze;
- počítač vždy nějak reaguje na operátorovu akci;
- uživatel musí být neustále v kontaktu se systémem;
  uživatel má být informován:
  - když něco dělá, má vědět, co dělá
  - když čeká, má vědět že čeká a proč
  - když končí dílčí činnost, má vědět, co může dělat dál a jak to zařídit (vždy jasná další cesta);
- uživateli má být dávána možnost přerušit činnost a vrátit se k předchozí činnosti případně přejít přímo k jiné činnosti;
- jedna obrazovka má obsahovat jedinou základní myšlenku (1 obrazovka = 1 idea);
- počítač je stručný, věcný, neužívá obtížná slova a znaky; uživatelova akce je jednoduchá, krátká;
- obrazovky mají respektovat podobnost pro rychlou orientaci (vždy totéž tamtéž).

3. Systém s nabídkovým dialogem

3.1 Model systému:

Systém se obecně skládá z bází dat a výkonných programů, jinak řečeno z údajů a funkcí nebo chcete-li z operandů a operací. Otázky uspořádání dat, způsobu jejich uložení a výběru, problémy relevantnosti údajů a pod. jsou sice při návrhu systému klíčově důležité, my je však ponecháme stranou, neboť dialogem se vždy realizují funkce (přičemž dialog sám je též funkcí).

3.2 Typové funkce:

Typové funkce realizují základní úkony s daty, tedy vložení dat ("přírůstek"), změnu existujících dat, likvidaci uložených dat ("úbytek") a prostý opis dat. Tyto typové funkce se vyskytují v různých podobách v každém systému. Přitom existují dva extrémy:

- systém, obsahující výhradně typové funkce (na př. pořizování dat záznamníkového typu)
- systém bez datových bází (na př. vědeckotechnický výpočet, kde uživatel vloží data a systém vydá výsledky aniž by cokoli dále uchovával).

Prostředky pro realizaci typových funkcí volíme podle toho, ke kterému extrému se navrhovaný systém přiklání. Vždy se však snažíme zachovat v celém systému jednotné postupy pro všechny typové funkce.

Na uživateli nežádáme, aby porozuměl datovým strukturám (od toho je správce banky dat), ale snažíme se data interpretovat věcně i formálně tak, jak jsou uživateli blízká (formuláře, kartotéky, . . .).

3.3 Speciální funkce:

Speciálními funkcemi uskutečňujeme činnosti, které jsou dány vlastním účelem systému. Pro soustavy s nabídkovým dialogem neexistuje obecný návod, jak speciální funkce řešit. Zde musí pracovat naplno fantazie a tvůrčí schopnosti autora systému. Dobré nápady pomohou realizovat zdánlivě nemožné funkce, bez nápadu nelze některé požadavky vůbec rozumně vyřešit !

Speciální funkce mohou být na př.

- výběry (kde nabídkový dialog nahrazuje výběrový jazyk typu QUERY)
- výpočty (kde dialogem určujeme postup a varianty výpočtu)

a podobně.

Návrh speciálních funkcí má respektovat určité obecné zásady (viz na př. kap. 2,4,5). Podněty pro užitečné nápady se dají získat studiem dobrých hotových systémů (komplexní literatura u nás neexistuje).

4. Dialog z pohledu uživatele

Zde uvedeme přehled zásad, které má řešitel respektovat, aby byl jeho systém uživatelsky "přítulný".

4.1 Směr počítač - člověk:

a) Typové funkce:

Počítač vede uživatele při vkládání dat tím, že vždy označí co vkládat a jak vkládat.

Metod je celá řada, nejvýhodnější je "vyplňování formuláře". Osvědčil se tento postup:

- zobrazí se stručný a výstižný název položky
- podle potřeby se přidá vysvětlivka (na př. měrná jednotka)
- zobrazí se zjednodušený formát (na př. se tečkami vyznačí maximální přípustný počet znaků; uživatel píše přímo na tyto tečky).

Nepoužíváme pevné formáty, nežádáme zarovnání napravo.

Lze zobrazit najednou celý formulář (všechny položky věty) a pak

- buď ponechat uživateli, aby nastavil kurzor na položku a vyplnil ji (pak musíme nakonec kontrolovat úplnost datové věty)
- nebo vést uživatele počítačovým nastavením kurzoru po jednotlivých položkách.

Při změně položky je vhodné zobrazit celou větu; uživatel ukáže kurzorem měněnou položku a přepíše její hodnotu. Ukončení změn oznámí uživatel speciálním znakem (na př. HOME UP).

Při vedení kurzoru pro změny klávesa VPRAVO znamená přeskok na položku vpravo (nikoliv posun o znak vpravo) a podobně.
Pro typové funkce s výhodou užíváme "malé menu" , které popíšeme níže.

b) Speciální funkce:
Uživateli nabízíme "menu" (český překlad "jídelníček" není nejvytříbenější, lepší snad je "nabídka"), ze kterého lze vybírat
① uvedením čísla žádané funkce
② nastavením kurzoru na žádanou funkci.
"Velké menu" zabírá celou obrazovku a určuje vstup do nějaké činnosti nebo do dalšího velkého menu.

Po vykonání funkce bývá užitečné použít "malé menu", umístěné na př. v pravém dolním rohu obrazovky (výstup, formulář a ostatní texty na obrazovce zůstávají), kterým se určí "co s provedenou činností"; na příklad po "vyplnění formuláře" pro přírůstek objeví se malé menu

ZAPIS	(zapiš větu do báze, nabídni další formulář přírůstku)
DALSI	(nezapisuj do báze, nabídni další formulář)
ZAPIS+MENU	(zapiš větu a vrať se k nadřazenému "velkému menu")
MENU	(nezapisuj a vrať se k "velkému menu").

V různých fázích umožníme uživateli návrat:
jestliže na př. zvolil přírůstek, vyplňuje jako první položku třeba "číslo zaměstnance"; vložením nuly neb záporného čísla "zastaví" se přírůstek a program se vrátí k nadřazenému "velkému menu".

Texty na obrazovce mají vysvětlovat funkce srozumitelně, avšak stručně. Ve starších systémech bývaly obvyklé vzkazy typu

PREJETE SI, PROSIM, PRAVIDLA HRY?
JESTLIZE ANO, ODPOVEZTE '1', JESTLIZE NE, ODPOVEZTE '0'!
Moderně řešený systém uvede týž vzkaz takto:

PREJETE SI PRAVIDLA ? (1=ANO, 0=NE)

V podstatě to znamená přechod od rozšafného konverzování k racionální práci; moderní systém nezavrhuje zdvořilost, ale neplýtvá zbytečnými slovy.

Velmi užitečné jsou "rady a informace uživateli"; osvědčilo se vyhradit jednu až dvě řádky obrazovky na základní informace

- co se právě dělá (MATERIAL - PRIRUSTEK) - v podstatě jde o opis vyhraného prvku předchozího "velkého menu"
- rady k tomu, co se dělá, na př. jak ukončit sekvenci (0 = KONEC)
- zda došlo k chybě

atd.

Rady a informace mají mít své stálé místo (oblast v řádku); neuvádějí se při výstupech výsledků (zabíraly by cenné řádky a stejně nejsou v tomto případě zapotřebí).

4.2 Směr člověk - počítač:

Po obecném uživateli můžeme chtít, aby
- ovládl práci s kurzorem a základními klávesami
- porozuměl významu jednoduchých slov a pojmů (v omezeném počtu), které označují důležité úkony
- zvládl logický princip práce s nabídkovým dialogem.

Užitečné bývá (alespoň pro začátek) nakreslit základní strom dialogu a pověsit ho na zeď k terminálu.

Jednotlivé obrazovky přizpůsobujeme účelu. Pro typové funkce vkládání hromadných dat budou obrazovky co nejjednodušší a práce s nimi elementární. Design obrazovky má být podobný designu formuláře, s kterým uživatel pracuje.

Často zdržuje "prokousávání se" obsáhlým a rozvětveným stromem dialogu. Označíme-li funkce v menu čísly, můžeme umožnit vkládání několika čísel najednou a realizovat tak "zkrácenou cestu stromem". Normální cesta vypadá třeba takto:

1. PRACE S DATY
2. VYPOCET

- volba "1" -

1. DATA - ZAMESTNANCI
2. DATA - MATERIAL
   - volba "2" -

1. PRIRUSTEK
2. UBYTEK
3. ZMENA atd.

Jestliže chci provádět přírůstek materiálu, mohu buď projít celým stromem nebo odpovědět zkráceně:

1. PRACE S DATY
2. VYPOCET
   - volba "121" -

Nejpoužívanější cesty si uživatel zapamatuje (nebo někam zapíše) a pracovní proces se zkracuje.

4.3 Všeobecné rady:

Uveďme si ještě bez ladu a skladu soubor drobných rad a doporučení:

- vzkazy a pojmy musí být stručné, ale jazykově čisté (dbáme na správnou češtinu neb slovenštinu);
- vždy všechno kontrolujeme (viz též kap. 5);
- v jednom kroku dáváme pouze nezbytný objem informací (nepožadované vyloučíme);
- pracujeme s jasnými, "čistými" a graficky úhlednými formáty; zvýrazňujeme vynechávání řádků, použitím rámečků nebo negativem (INVERSE VIDEO); zarovnáváme sloupce; vyhneme se grafické "roztříštěnosti" obrazovek;
- používáme novinářského principu rozdělení textu do více krátkých sloupců: max. 30 - 40 znaků lze přečíst bez pohybu oka;
- totéž slovo musí mít vždy týž význam;
- pro tutéž funkci použijeme vždy téhož slova (v systému se na př. nesmí objevit pro pokračování akce POKRACUJ jinde POKRAC. a jinde POKR.);
- co nejvíce omezíme použití speciálních znaků směrem k uživateli i od uživatele (aby se nemuselo pořád hledat v manuálu co znamená hvězdička a k čemu je zavináč).

5. Dialog z pohledu programátora

5.1 Kontroly a ošetření chyb:

Interaktivní systémy vylučují některé běžné metody kontroly dat jako na př. přezkoušení. Z povahy práce systému vyplývá však nutnost co nejdůkladnějších kontrol. Lze použít těchto principů:

- Široké využití kontrolní číslice (vypočtené nějakým algoritmem ze základního čísla a přidané k němu); pro alfanumerické údaje lze použít i kontrolní písmeno!
- Potvrzení popisem: vložíme-li na př. číslo zaměstnance, počítač napíše ihned jeho jméno.
- Přezkoušení opisem: při "úbytku" oznámí uživatel číslo věty; počítač větu vypíše a žádá rozhodnutí "NECHAT - ZRUSIT" ("malé menu"!).
- Kontrolují se všechny existující vazby mezi daty.
- Kontroluje se logická přípustnost dat i operací.
- Je-li to možné, kontroluje se logický sled transakcí a logické vazby mezi operacemi.

Pro speciální případy lze navrhnout speciální kontroly. Příklad: V jednom systému VÚS vkládají se zeměpisné souřadnice klíčových bodů státních hranic pro různá teritoria. Po vložení množiny bodů se na obrazovce nakreslí zadané teritorium; letmou konfrontací s mapou lze snadno odhalit chyby v údajích.

5.2 Programovací jazyk:

Pro psaní programů nabídkového dialogu neexistuje dobrý speciální jazyk (ač byly činěny pokusy o jeho vytvoření - na př. "DIAGEN"). Největší obtíží je programování typových funkcí - vyplňování a změny na formulářích. Pro tento účel je vhodné vyvinout nějakou pomůcku. VÚS používá na př. "popis obrazovek", kdy ve zvláštním souboru dat je pro každou obrazovku uložen detailní popis textů, položek, formátů atd. včetně údajů o posloupnosti vyplňování. Soubor je ovládán několika podprogramy pro elementární funkce a akce.

5.3 Zajištění a zabezpečení systému:

Programátor musí zabezpečit data i funkce

- proti neznalosti a chybné manipulaci
- proti zneužití a proti zlomyslnému narušení
- proti technickému nebo provoznímu narušení.

Otázka zajištění systému je složitá a její rozbor by vydal na samostatný článek. Základní zabezpečení se provádí soustavou hesel a priorit přístupu k datům i funkcím. Běžně se užívá systém bezpečnostních kopií datových bází. Z provozního hlediska je účelné

- evidovat "status" datových bází (datum posledního přístupu do každé báze) a na vyžádání tento status zobrazit
- zabudovat funkce vytváření přehledů o obsazení bází s případnou evidencí hesel, pod kterými byly v bázích prováděny zásahy.

6. Závěr

Autor se čtenářům omlouvá za přehnanou stručnost kapitol 3 až 5. Je jasné, že zásady a pokyny by měly být rozvedeny, doplněny příklady (ilustrativních příkladů je dostatek) atd. Navíc existuje ještě řada dalších pravidel a teoretických i praktických poznatků. Komplexní zpracování dnes dostupného materiálu by si vyžádalo celou knihu.

Na jedné straně je škoda, že rozsáhlé ucelené znalosti nelze předat širšímu okruhu zájemců. Na druhé straně je příjemné zjištění, že teorie i praxe nabídkových dialogů je již tak hluboce rozvinuta.

Závěrem pozusmenejme, že ve VÚS byly vyvinuty a předány k praktickému využití velké výpočetní systémy (s nabídkovým dialogem) pro plánování zemských i kosmických radiokomunikačních služeb. Všechny popsané zásady (a některé další) byly v systémech respektovány a uživatelská praxe potvrdila jejich správnost. Přitom se jedná o systémy s rozsáhlými datovými bázemi a komplikovanými výpočetními a simulačními funkcemi.

Autor nemíní tímto konstatováním opěvovat práci, na které se podílel, pouze jeví snahu doložit čtenáři užitečnost tézí, obsažených v tomto příspěvku.

7. Literatura a prameny

/1/ Martin: Design of Man-Computer Dialoguee. Prentice Hall,1973.
/2/ Bébr: příspěvky ve sbornících semináře Havířov a Ostrava (DT ČSVTS Ostrava) z let 1977,1979,1980,1981,1982.
/3/ Podklady z diskuse na semináři "Programování 83".