

Od děrných štítků k Internetu aneb 40 let programovacího jazyka Cobol

Vlastimil Čevela

Programátorské služby Cobol, Benešova 279, 664 42 Modřice, tel. 05-4721-6183
e-mail:cevelavl@mbox.vol.cz <http://www.vol.cz/pcweb/cobol>

Předkládaný materiál je pokusem o charakteristiku programovacího jazyka Cobol v kontextu jeho historie a současného stadia rozvoje informačních technologií.

Důvody vzniku jazyka

Ve dnech 28. a 29. května 1959, tedy téměř na den přesně 40 let před konáním tohoto 25. ročníku semináře Programování/Tvorba-sw se v USA na Pensylvánské univerzitě konala schůzka, která měla rozhodnout o účelnosti a možnosti vytvořit společný programovací jazyk, zaměřený na zpracování dat, vyskytujících se v podnikání. Situace byla charakterizována následovně:

- a) Je nutné vyvinout pro existující počítače i pro počítače vyráběné v budoucnosti systém, který umožňuje nahrazení jednoho počítače jiným a to i nekompatibilním tak, aby náklady, spojené s přenosem programů byly minimální.
- b) Rychlý rozvoj vědy a techniky a prudké zvyšování požadavků na řízení vedou k nutnosti častých úprav v programech. Je proto žádoucí, aby součástí programu byla tak úplná dokumentace, která by tyto změny umožňovala.
- c) Často je nutno produkovat velké množství programů v krátkých termínech, a to s pomocí relativně nezkušených programátorů.

Schůzky se zúčastnili reprezentanti významných uživatelů, výrobců počítačů, univerzit, armády a vládních orgánů USA. Účastníci se shodli na vhodnosti tvorby takového jazyka a vytvořili organizaci CODASYL (Conference On Data Systems Languages) jako dobrovolné sdružení uživatelů a výrobců počítačů, kterou pověřili konkrétní prací.

Paní Hopper a FLOW-MATIC

Pracovní skupina, kterou z počátku tvořilo 6 lidí pracovala velmi rychle (pro délku své činnosti byla nazývána "Short Range Committee") a již 17. prosince 1959 byl výkonnému výboru CODASYL předložen návrh jazyka COBOL (Common Business Oriented Language). Bylo totiž na co navazovat, protože od roku 1957 fungoval jazyk FLOW-MATIC, který včetně kompilátoru navrhla a vytvořila jistá paní Grace Murray Hopper ve firmě UNIVAC. Tento jazyk, který umožňoval převést "English-like instructions" do strojového kódu počítače je vedle IBM COMTRAN (Bob Bemer) považován za přímého předchůdce Cobolu.

Kariéra uvedené dámy je doslova historií počítačového zpracování dat. Po "PH.d in Mathematics" na Yale University v roce 1934 nastoupila v roce 1943 ve svých 37 letech k americkému námořnictvu a stala se jedním ze 3 programátorů prvního číslicového počítače MARK 1 na Harvardské univerzitě. U námořnictva působila dalších 43 roků a do důchodu odešla v roce 1986 jako "Navy's oldest officer" v hodnosti kontradmirála. V padesátých letech patřila mezi těch několik málo vizionářů, kteří správně odhadli budoucnost počítačů mimo čisté aritmetické výpočty. Za svoji práci sklízela řadu ocenění a titulů, včetně medaile od prezidenta Bushe "za pionýrský příspěvek k rozvoji počítačových programovacích jazyků, které zjednodušily počítačovou technologii a otevřely dveře výrazně širšímu okruhu uživatelů". Když pak paní Hooper počátkem roku 1992 zemřela, US-Navy na její počest pojmenovalo nový torpedoborec.

Bob Bemer a poetický název

Autorem samotného názvu COBOL je dle rozhovoru v "The Cobol Letter" na Webu Bob Bemer, který jako "IBM's chief of programming standards" byl též přímým účastníkem událostí před 40 lety (a taky vymyslel klausuli PICTURE, spolupracoval na zrodu ASCII, ESCape sekvence, standardu 8bit/byte atd.):

CR: How did you arrive at the name COBOL?

BB: Cobol to me has a nice round sound - a lyrical quality (drawing an imaginary hourglass in the air). The sound reminds me of a women's figure.

CR: Are you saying that Cobol, the language that is often considered the epitome of design by committee and bureaucracy, was named with Venus de Milo musings in mind?

BB: Yes (laughing).

CR: I must say I've been programming for over 20 years in Cobol and never heard that one. What did Grace Hopper have to say about your metaphorical naming?

BB: She just laughed and said okay.

Autoři a volné šíření specifikací

Základním znakem jazyka Cobol byla od počátku otevřenost a držitelé autorských práv povolili volné užívání jeho specifikací, pouze se žádostí o zveřejnění následujícího "Uznání":

"Každá organizace zainteresovaná na reprodukci zprávy o COBOLu a jeho specifikacích vcelku nebo v jednotlivých částech může volně používat všech nápadů (ideas) z této zprávy jako základ programovací příručky nebo k jakýmkoliv jiným účelům. Ovšem všechny takové organizace jsou žádány, aby reprodukovaly tuto kapitolu jako část úvodu k příslušnému dokumentu ..." "COBOL je průmyslový jazyk a není vlastnictvím žádné společnosti ani skupiny společností ..."

"Autoři a držitelé autorských práv: ...

FLOW-MATIC for the UNIVAC, copyrighted 1958,9 by Sperry Rand Corp.; IBM Commercial Translator, No. F28-8013, copyrighted 1959 by IBM;

FACT, DSI 27A5260-2760, copyrighted 1960 by Mineapolis-Honeywell;

povolili používání vcelku i v částech ve specifikacích Cobolu a toto povolení zahrnuje i reprodukování v programovacích manuálech a podobných publikacích ..."

V oficiální zprávě o Cobolu je v souvislosti s původním vývojem a prací podpůrné skupiny vyjmenováno celkem asi 20 organizací, včetně takových jmen jako DuPont, General Electric, General Motors, Lockheed atd včetně US Air Force, US Navy a Bureau of Standards Department of Commerce.

Za 40 let se doba změnila, ale o současném významu jazyka určitě vypovídá složení výboru pro X3J4 ANSI Standard OO COBOL z roku 1996: American Express, Computer Associates, Fujitsu Software, Hitachi Software, Hewlett-Packard, IBM, Micro Focus, EDS, Unisys.

Jazyk COBOL se skutečně povedl a brzy se stal prakticky celosvětově univerzálním prostředkem k tvorbě programů pro zpracování ekonomických dat. Mnohokrát mu byl předpovídán zánik, ale zatím se vždy ukázalo, že stále zůstává významným nástrojem v rámci informačních technologií a vše nasvědčuje tomu, že tak bude i nadále.

Podívejme se stručně na některé vlastnosti, které jsou na Cobolu tak užitečné.

Čtivost a přehlednost jazyka

Především je to všeobecná srozumitelnost, odvozená z původního zadání. Cobol totiž až na nepatrné výjimky (fd, corr, pic ...) prakticky nezná zkratky či akronymy, ale vyžaduje popsání algoritmu zpracování normálními anglickými slovy jako třeba:

move A to B
přesuň A do B

```
perform CTENI until KONEC
prováděj CTENI dokud není KONEC
```

```
compute V rounded = (A + B) * (C - D) / E
spočítej V zaokrouhleně = ...
```

Pokud vhodně zvolíme programátorská jména (jako "neangličany" nás neblokují rezervovaná slova např. PIS, CTI apod.), grafickou úpravu a doplňující komentáře, získáme skutečně dokonale samodokumentující zdrojový text. Samozřejmě, že v dávných dobách před objevem strukturovaného programování někteří "umělci" dokázali algoritmus pomocí nekontrolovaných "go to" pořádně zamotat, ale to je o něčem jiném. V uvedených souvislostech si uvedeme třeba řídicí algoritmus sekvenčního přečtení souboru SB, který má 2 typy vět:

Příklad "pg1":

```
procedure division.
  open input SB.
  set CTENI to true.
  perform until KONEC
    read SB
    end
    set KONEC to true
  not end
  evaluate TYP
    when 1 perform VETA-1
    when 2 perform VETA-2
    when other perform CHYBA
  end-evaluate
end-read
end-perform.
close SB.
```

Nebo na proceduru aktualizace vět VT dle klíče KLIC přímo přístupného indexového souboru SB:

Příklad "pg2":

```
AKTUALIZACE.
  evaluate PRACE
    when "doplněk"
      write VT
    when "změna"
      rewrite VT
    when "rušení"
```

```
delete SB
when "start"
  start SB key not < KLIC
  read SB next
when "čtení"
  read SB next
when other
  move "chyba" to PRACE
end-evaluate.
```

Univerzálnost - popisy dat - programátorský komfort

Uvedené texty přitom nejsou žádným pseudokódem pro zápis algoritmu, ale mezinárodně (ISO) normovaným zdrojovým kódem ANSI COBOL 85, který bude správně přeložen všemi kompilátory Cobol na všech platformách od 16-bitového MS DOS, přes OS/2 a 32-bitové MS Windows až po DEC-VAX a UNIX, či velké mainframe Siemens BS2000 nebo IBM OS/390.

Nejsilnější prostředky Cobolu jsou od počátku v popisu dat. Datové typy dokonale na různých úrovních strukturovaných skupinových i elementárních datových položek v něm byly již před 40 lety zavedeny s tak samozřejmou "common sense" logikou, že dodnes nebyly ničím překonány.

Samozřejmě, že dnešní vývojová prostředí COBOL už nejsou jenom jazyk jako takový, ale veškerý GUI (Grafical User Interface) a RAD (Rapid Application Development) komfort včetně SQL komunikace s databázemi. A samozřejmě též "Source Code Control System" řízení verzí - To je ale trochu jiná kapitola mimo naše téma. Podrobnější charakteristiky produktů jednotlivých výrobců je možno se dočíst přes odkazy na Web stránky v závěru.

OO COBOL

Důkazem toho, že "Old Faithful COBOL" to ani ve sféře jazykových prostředků nemíní jen tak vzdát, je mimo jiné i skutečnost, že je "true OO", tedy "Object Oriented". OO COBOL běžně pro Windows/Mainframe dodávají Merant-MicroFocus, Hitachi a Fujitsu a samozřejmě IBM v rámci OS/390.

Z hlediska zdrojového programového kódu je přitom OO rozšíření velice jednoduché: Namísto "program-id." píšeme "class-id." s příp. "inherits" pro definici tříd a "method-id." pro rekurzivně volatelné algoritmy metod. V rámci "class-control" uvedeme seznam využívaných tříd a v "local-storage section" ukládáme data pro instance objektů, identifikované datovými položkami "object reference". V proceduře zpracování pak máme nový příkaz "invoke" pro předávání zpráv objektům - a to je prakticky vše podstatné.

K tomu pozdní dynamická vazba objektů, spolupráce s objekty Windows, C++ a jiných jazyků, včetně respektování rozhraní CORBA (Common Object Request Broker

Architecture) a DCOM (Distributed Common Object Model) - určitě docela slušná výbava k přežití roku 2000.

Od děrných štítků k Internetu

O tom, že COBOL opravdu začínal na děrných štítcích svědčí stále dodržovaný praktický formát jeho zápisu, tj. na každém řádku prostor pro komentáře od 1. do 6. (číslování štítků) a od 73. do 80. sloupce (jméno programu), aby bylo možno rozsypané štítky správně poskládat.

A na závěr tedy slíbená komunikace s COBOL programem přes Internet, která je velmi jednoduchá. Např. při využití MF NetExpress totiž stačí deklarovat skupinovou položku jako "external-form" a elementární datové položky pomocí "identified by" doplnit "NAME" jmény HTML formuláře a použít příslušný příkaz "accept". Pro výstup pak buď lze využít obdobnou deklaraci a příkaz "display" pro předem (chcete-li pomocí RAD) připravenou výstupní HTML šablonu, anebo do textu Cobol vložit pomocí "exec html"/"end-exec" přímo příslušné tagy. Všechno ostatní pro tvorbu CGI/ISAPI/NSAPI pak již vyřeší kompilátor.

Zdrojový text CGI programu, komunikujícího s grafickým Internetovým rozhraním, tedy může třeba vypadat následovně:

Příklad "pg3":

```
$set preprocess(htmlpp) endp
  program-id.
    pg3.
  working-storage section.
    1 INP external-form.
    2 JMENO pic x(10) identified by "jme".
    2 POPIS pic x(50) identified by "txt".
    2 PRACE pic x(10) identified by "kod".

  procedure division.
    accept INP.

    exec html
    <HTML><HEAD><TITLE>PG3-OUT</TITLE></HEAD><BODY>
    <H1>Výstup z PG3</H1>
      JMENO = :JMENO <BR>
      POPIS = :POPIS <BR>
      PRACE = :PRACE <BR>
    </FORM></BODY></HTML>
    end-exec

  stop run.
```

Příklad vstupního HTML formuláře "form-i" pro "pg3":

```
<HTML><HEAD><TITLE>Vstupní formulář pro PG3</TITLE></HEAD><BODY>
<H1>Vstupní formulář pro PG3</H1>
<FORM METHOD="POST" ACTION="/cgi-bin/PG3.EXE"><PRE>

Jméno <INPUT TYPE="TEXT" NAME="jme" VALUE="">
Popis <INPUT TYPE="TEXT" NAME="txt" VALUE="">
<P>
  <INPUT TYPE="SUBMIT" NAME="kod" VALUE="čtení">
  <INPUT TYPE="SUBMIT" NAME="kod" VALUE="start">
  <INPUT TYPE="SUBMIT" NAME="kod" VALUE="doplňk">
  <INPUT TYPE="SUBMIT" NAME="kod" VALUE="změna">
  <INPUT TYPE="SUBMIT" NAME="kod" VALUE="rušení">

</PRE></FORM></BODY></HTML>
```

Literatura:

1. Kolář J., Fiala J., Machačka I., Podešva P., překlad, Cobol Tesla 200, Tesla Dataservis Praha 1970
2. Drbal Pavel, Historie Cobolu, in Základy technologie programování v jazyku Cobol, DT ČSVTS Pardubice 1981
3. Firemní dokumentace Micro Focus NetExpress, 1988
4. Firemní dokumentace IBM Cobol for OS/390, 1998
5. Čevela V. a další, <http://www.vol.cz/pcweb/cobol>, 1999