

MODELOVÁNÍ JAKO METODA OPTIMALIZACE PRÁCE ŠTÁBU

Jednou z oblastí aplikace číslicových počítačů je modelování. Vojenskovevědecký výzkum pro omezenou možnost přímého experimentování a pro velkou složitost zkoumaných jevů nalezl v modelování již počátkem 50. let možnou výzkumnou techniku. V souvislosti se zaváděním rozsáhlých a složitých systémů (automatizované systémy velení, počítačové sítě, zbraňové systémy apod.) význam modelování roste. V etapě návrhu takovýchto systémů, změn v již uskutečňovaných systémech, nebo i při jejich řízení, je modelování a především jeho část simulace systémů, často jedinou možnou metodou jejich exaktního zkoumání. Častější využívání simulace systémů podněcuje její další rozvoj. Tak se v posledních letech objevila simulační technika nazývaná „předmětně zaměřené programování“. Ta se jeví jako velice progresivní, je intenzivně rozvíjena, neboť zpřístupňuje modelování i odborníkům z oblasti řízení a návrhu složitých systémů (velitelům, štábním pracovníkům a pracovníkům ve vojenském výzkumu), aniž by museli být kvalifikovanými programátory.

Podstata techniky předmětně zaměřeného programování spočívá v tom, že zdatnými programátory je v univerzálním programovacím jazyce vysoké úrovně (jakým je např. Simula 67) vybudován předmětně zaměřený jazyk. [Někdy nazývaný „balík programů“]. Tento jazyk obsahuje pojmy

(obecné typové modely částí systému, jejich činností apod.) obvyklé v oblasti, do níž náleží zkoumaný objekt. Je-li např. touto oblastí vojenská výpočetní technika, jsou pro ni typické pojmy: front, armáda, štáb, počítač, banka dat, hlášení a mnohé jiné. Tyto pojmy jsou velmi blízké potenciálnímu uživateli a ten se v nich snadno vyjadřuje. Uživatel tím, že v takto připraveném jazyce popíše svůj zkoumaný objekt, získá jeho model, který je realizován jako výpočet na číslicovém počítači. Po nutných проверkách věrnosti zobrazení zkoumaného objektu modelem (verifikace a validizace modelu) lze s modelem experimentovat, přičemž výsledky experimentů jsou interpretovány tak, jako by byly získány experimentováním se zkoumaným objektem. Podle otázek, na něž jsou pomocí modelování hledány odpovědi, jsou určovány podmínky jednotlivých experimentů. Nejčastěji jsou to otázky typu: „Jak se v činnosti objektu projeví, jestliže uskutečníme tu a tu změnu?“ Takovýmito změnami mohou být: změna některých vlastností prvků objektu, popřípadě struktury objektu, změna pracovních podmínek (zátláže), změna strategie řízení objektu apod.

Technika předmětně zaměřeného programování byla použita pro modelování automatizačních prostředků, které jsou uvažovány pro využití ve štábech RVD na stupních armády, fronty. Pro modelování byl užit předmětně zaměřený jazyk GPSS Pra-

gue, který byl k tomuto účelu rozšířen o některé další pojmy — třídy (classes), např.:

— „Class POCITAC“ což je třída, která zobrazuje počítač štábu frontu (armády), jeho technické a část programového vybavení včetně terminálu určeného pro štáb RVD a banky dat počítače. Programový popis této třídy je vybaven tak, aby během experimentu automaticky shromažďoval statistické údaje pro hodnocení účinnosti změn v modelu. O každém počítači sbírá údaje o jeho činnosti, automaticky vypočítává hodnoty jeho vytíženosti apod.

— „Class ULOHA“ zobrazuje jednotlivé, na počítači ve prospěch štábů RVD řešitelné úlohy, jejich nároky na aktuální položky z banky dat, časové charakteristiky řešení úloh atd.

— „Class ZPRAVA“ zobrazuje dílčí zprávy — hlášení, která zasílají útvary a složky RVD a jiných druhů vojsk a na základě kterých je plněna nebo aktualizována banka dat počítače. Ta nejenže zobrazuje vlastní dílčí zprávu, ale i činnosti s ní spojené, a to odeslání zprávy z místa kde vznikla, její dopravu přenosovým zařízením i aktualizaci banky dat. Třída ZPRAVA rovněž sbírá údaje. O každé zprávě zaznamenává, jak dlouho tato obsluha trvala apod.

— „Class POZADAVEK“ zobrazuje požadavek na řešení úlohy počítačem. Požadavek vznikne ve štábu RVD frontu (armády) v průběhu plánování a řízení operačního úkolu. Podobně jako „class ZPRAVA“ zobrazuje rovněž veškeré činnosti spojené s realizací požadavku, a to od vyvolání požadované úlohy operátorem počítače, přes kontrolu naplnění banky dat pro úlohy potřebnými údaji, až po předání výsledků úlohy pracovníkovi štábu. Podobně jako předchozí i tato třída sbírá údaje o průběhu obsluhy každého požadavku.

Pro získání simulačního modelu byl v rozšířeném GPSS Prague popsán zkoumaný objekt, který zahrnoval:

— počítače vševojskových štábů frontu a armád včetně terminálů a programového vybavení určeného ve prospěch práce štábů RVD na uvedených stupních,

— přenosové zařízení, které má zabezpečit přenos dílčích zpráv od útvarů a složek RVD a od některých částí vševojskových štábů i počítačům.

Zkoumaným obdobím práce objektu bylo období plánování a řízení operačního úkolu. Pracovní zátěž objektu představuje dva typy obsluh. Za prvé je to obsluha všech dílčích zpráv, které během zkoumaného ob-

dobí vznikají ve štábech a útvarech RVD a v některých částech vševojskových štábů. Dále je to obsluha požadavků na řešení úloh, které se rovněž v tomto období objevují ve štábech RVD frontu a armád. Tedy podstatnou složku, kterou nutně model zobrazoval, tvořila všechna místa vzniku dílčích zpráv a požadavků na řešení úloh (okolí objektu). Popis zkoumaného objektu a jeho okolí byl členěn do několika částí. Prvá z nich na základě vložených parametrů generuje jednotlivé počítače — prvky třídy „POCITAC“ a uspořádává je do sítě. Další části čtou z černých štítků údaje charakterizující vznik dílčích zpráv a vznik požadavků na řešení úloh v okolí objektu. Tyto části na základě přečtených údajů v průběhu simulačního experimentu generují a aktivují jednotlivé dílčí zprávy a požadavky. Pro názornost použití techniky předmětně zaměřeného programování uvádím příklad popisu generování a aktivování jednoho z požadavků na řešení úlohy.

Příklad: Ve štábu RVD frontu za určité situace v nějakém okamžiku během plánování a řízení operačního úkolu vznikne požadavek, aby počítač štábu frontu řešil úlohu s označením RXXXX. Její výsledky potřebuje mít pracovník štábu RVD k dispozici nejpozději do určité doby, kterou jsme označili písmeny TR. Splnění podmínek vzniku, jakož i vlastní okamžik vzniku, podléhá řadě náhodně proměnných faktorů. Pro vyjádření jejich vlivu byl vznik požadavku popsán stochastickými výrazy. [Model je vždy zjednodušením reality]. Po statistickém šetření, ale i s použitím kvalifikovaných odhadů, byla pro vznik tohoto konkrétního požadavku přijata následující hypotéza: Požadavek na řešení úlohy RXXX s pravděpodobností PZ v náhodném okamžiku CAS, který má normální rozložení se střední hodnotou A a rozptylem B.

Poznámka: Jazyk Simula 67, ve kterém je definován použitý GPSS Prague, obsahuje bohatou paletu procedur náhodného výběru. Pro modelování pravděpodobností PZ lze užít proceduru draw (PZ, U), pro modelování normálního rozložení N (A, B) proceduru normal (A, B, U).

Popis generování a aktivace požadavku má v modelu tvar:

```
...
if draw (PZ, U) then begin
  CAS = normal (A, B, U);
  activate new POZADAVEK (RXXXX, TR)
at CAS;
end;
...
```

Tento stručný popis vyvolá během simulačního experimentu činnost, která je rozdělena do dvou etap. V první (před spuštěním simulárního času) je v případě příznivého výsledku procedury draw (PZ, U) vyvolána procedura normal (A, B, U), která určí okamžik CAS. Poté je příkazem new vygenerován jeden prvek třídy POZADAVEK s atributy RXXXX a TR. Tím je v operační paměti počítače (který dělá simulaci) vytvořena samostatná dynamická datová struktura, do které jsou především ukládány hodnoty všech sledovaných vlastností (vnějších i vnitřních atributů). Ve druhé etapě (spuštěný simulární čas dospěl k hodnotě CAS) je tento konkrétní požadavek aktivován a začne se aktivně podílet na činnostech probíhajících v modelu, a to podle popisu daného definicí třídy POZADAVEK. Obsazuje tedy příslušný počítač, nebo čeká na jeho uvolnění, vyvolá žádoucí úlohu, kontroluje okamžitou naplněnost banky dat atd. Během této činnosti se jeho datová struktura doplňuje a jsou automaticky shromažďovány údaje pro hodnocení v konci experimentu. Pojmem automaticky je zde myšleno to, že činnost probíhá bez zásahů a jakékoli péče uživatele simulačního programu. Rovněž tak se experimentátor nemusí starat ani o paralelitu procesů probíhajících při experimentu v modelu. (Konec příkladu.)

Po dokončení popisu zkoumaného objektu a jeho okolí, tedy po získání modelu, následovaly jeho verifikace a validizační prověrky. (Verifikace je nazývána etapou, kdy jsou odstraňovány programátorské chyby a je prověřována logická konzistence jednotlivých částí popisu i simulačního programu jako celku. Při validizaci modelu jsou prováděny první experimenty s modelem. Pomocí dat, která jsou při experimentu zjištěna, je sledováno chování modelu. To je v této etapě porovnáváno s cho-

váním modelovaného objektu nebo je expertně hodnoceno. Účelem validizace je k modelu získat takovou důvěru, která umožní, aby závěry ze studijních experimentů s modelem bylo možné vztahovat na zkoumaný objekt.)

Pro vlastní studijní experimentování byla data pro naplnění modelu volena tak, aby podmínky jednotlivých experimentů vystihovaly následující změny:

- v úrovni sdílení počítačů ostatními složkami štábu (mimo RVD),
- ve výbavě terminálu určeného pro štáb RVD,
- v metodice řízení práce objektu,
- technického vybavení určeného pro datové přenosy.

Ze statistického rozboru dat z jednotlivých experimentů byly získány informace nejen o úrovni schopností modelovaného objektu v plnění jeho základních funkcí, ale i informace umožňující zhodnotit důsledky experimentovaných změn. Modelováním bylo zjištěno, že vytvořená počítačová síť, včetně terminálů, programového vybavení a uvažování přenosových cest není v plném rozsahu schopna zabezpečovat včasné splnění požadavků RVD. Byla proto navržena konkrétní opatření k její úpravě. K hodnocení účinnosti jednotlivých změn byl a nadále lze využít současný model.

Přípravou i použitím modelu bylo znovu potvrzeno, že technika předmětně zaměřeného programování je skutečně přístupnou technikou a že má velmi široké aplikační možnosti. Dále bylo prokázáno, že modelování a speciálně simulace, je-li řádně užita, umožňuje snížit náklady na vývoj, zkvalitňuje proces tvorby koncepce i praktické realizace složitých systémů.