

Pseudointerakce, a jak dál?

Emil Kopčıl

Motto : Netvrď, že to nejde - přiznej, že to neumíš

Nepoučený laik věří, že údaj jednou vložený do počítače bude mít kdykoliv k dispozici. Zkušený uživatel už dobře ví, že to tak není.

Problém není ani tak v tom, že by se nám informace z počítače ztrácely, jako spíše v tom, že je nedokážeme vyhledat a adekvátně využít.

Obecně známým, avšak nedoceněným řešením opakovaného přístupu k uloženým datům je technologie číselníků.

Pro potřeby tohoto textu si představme číselník jako množinu (soubor) formalizovaných datových struktur (vět), složenou z přístupového klíče, umožňujícího jednoznačnou identifikaci věty v rámci souboru, a z vlastního těla věty, v němž jsou obsaženy větou nesené informace. Pro úplnost poznamenejme, že tělo věty často obsahuje také vedlejší přístupové klíče, obvykle jednoznačnou identifikaci věty neumožňující.

Implementujeme-li číselník jako indexsekvenci soubor, zdálo by se, že už není co řešit. Podívejme se však na věc očima uživatele, pracujícího na personálním počítači s programy, jaké jsou v našich podmínkách obvyklé.

Přepokládejme, že je potřeba doplnit do číselníku vozidel následující údaje o novém vozidle :

- číslo vozidla
- kód typu vozidla
- kmenové středisko vozidla
- osobní číslo řidiče

Jak bude náš uživatel postupovat :

- zavolá program pro aktualizaci číselníku vozidel
- zadá číslo vozidla
- zjistí, že nezná kódy typů vozidel
- ukončí program pro aktualizaci číselníku vozidel
- zavolá program pro prohlížení číselníku typů vozidel
- vyhledá si příslušný kód typu vozidla
- ukončí program pro prohlížení číselníku typů vozidel
- zavolá program pro aktualizaci číselníku vozidel

- zadá číslo vozidla a doplní nalezený kód typu vozidla
- u čísla střediska program nahlásí, že je nezná
- uživatel ukončí program pro aktualizaci číselníku vozidel
- zavolá program pro prohlížení číselníku středisek
- zjistí, že ono středisko v číselníku skutečně není
- ukončí program pro prohlížení číselníku středisek
- zavolá program pro aktualizaci číselníku středisek
- doplní požadované údaje o kmenovém středisku vozidla
- ukončí program pro aktualizaci číselníku středisek
- zapudí neodbytnou vzpomínku na pohádku o kohoutkovi
- zavolá program pro aktualizaci číselníku vozidel
- zadá číslo vozidla a doplní číslo kmenového střediska
- podiví se, že program nezná ani číslo řidiče
- ukončí program pro aktualizaci číselníku vozidel
- zavolá rovnou program pro aktualizaci číselníku pracovníků
- zadá číslo řidiče a ostatní požadované údaje
- zaskřípe zuby, protože program nezná středisko řidiče
- ukončí program pro aktualizaci číselníku pracovníků
- zavolá program pro aktualizaci číselníku středisek
- doplní požadované údaje o kmenovém středisku řidiče
- ukončí program pro aktualizaci číselníku středisek
- zavolá program pro aktualizaci číselníku vozidel
- zadá číslo vozidla a číslo řidiče
- zakleje, protože program řidiče pořád ještě nezná
- uvědomí si, že nedokončil aktualizaci číselníku pracovníků
- ukončí program pro aktualizaci číselníku vozidel
- zavolá program pro aktualizaci číselníku pracovníků
- zadá číslo řidiče a jeho nešťastné kmenové středisko
- ukončí program pro aktualizaci číselníku pracovníků
- zavolá program pro aktualizaci číselníku vozidel
- zadá číslo vozidla a číslo řidiče
- ukončí program pro aktualizaci číselníku vozidel
- setře z čela pot a pokračuje v budování světlých zítřků

Zdá se vám to přehnané ? Samozřejmě máte úplnou pravdu. V opravdovém životě pracovník pověřený údržbou číselníku pravděpodobně nebude mít zároveň oprávnění k udržování všech potřebných číselníků, takže může mluvit o štěstí, podaří-li se mu popsanou akci dokončit v průběhu jediného dne.

Musím připustit, že se vyskytují i aplikace zmenšující uživatelské utrpení alespoň vnořeným voláním podrízených programů (s návratem na místo přerušeno postupu). Kompetenční bariery však pořád přetrvávají a mohu potvrdit, že o to víc vyniknou.

Častěji se však setkáme s aplikacemi, které se nechovají ani tak málo rozumně, jak jsem popsal v našem hororu. Zdá se to až neuvěřitelné, ale setkal jsem se s produktem (firmy zabývající se vývojem a prodejem software), který nechá uživatele natypovat cokoli. Chyby indikuje až po ukončení práce, jen spustí-li uživatel příslušný kontrolní program. Data se aktualizují jiným programem než kterým se pořizovaly, ale to už snad ani neudiví.

Ukončíme však tento malý, ale věru vyčerpávající výlet do klasické pseudointerakce. Přímě jako bych slyšel moudré hlasy: těmito problémy je naprosto zbytečné se zabývat, všechno vyřešíme zavedením počítačových sítí. Vážení propagátoři takovýchto postojů, obávám se že lžete v nejlepším případě sami sobě.

Mám zato, že je v podstatě jedno, zabráním-li kolegovi v přepsání mých dat tím, že je zamknu do šuplíku, nebo tím, že mu k mým datům nedovolím přístup v síťové aplikaci. Přepíše-li mi nějaký zloduch nekvalifikovaným způsobem data, nebude mě asi zajímat, bylo-li to v síti nebo bez sítě.

Ať už v síťovém prostředí nebo lokálně, musíme vyřešit následující rozpor:

- Chceme-li zajistit komplexní zpracování informací (data pořízená jedním pracovištěm využívat i na jiných místech), musíme mít k dispozici navzájem jednotné, tedy centrálně udržované číselníky, tvořící integrační prvek aplikace.
- Chceme-li dosáhnout opravdu interaktivního způsobu práce, musíme aktualizaci informací umožnit každému uživateli tehdy, kdy ji potřebuje - jinými slovy, údržbu číselníků důsledně decentralizovat.

Bude seriózní přiznat už na tomto místě, že to neumím. Přesněji řečeno, že nemám v praxi ověřeno, jestli navrženým řešením lze dosáhnout potřebného efektu. Zavedení opravdové interakce do práce s číselníky však mám za tak aktuální, že jsem se své následující úvahy rozhodl publikovat.

V každém reálném číselníku zákonitě musí v průběhu času vzniknout potřeba provedení změn. Má-li tedy být aplikační program dlouhodobě použitelný, musí aktualizaci číselníků nejen umožňovat, ale i rozumným způsobem aktivně podporovat.

Podrobnější analýzou zjistíme, že při změnách číselníků nevystačíme s prostým přidáváním, měněním a rušením vět, jak je programátorovi nabízí systém ovládání souborů (SOS).

Řekněme, že zmíněný číselník vozidel má jako přístupový klíč státní poznávací značku (SPZ) vozidla. Tabulka s SPZ se ale celkem snadno poškodí a novou tabulku se stejnou SPZ nelze získat. Nezbývá než vozidlu přidělit jinou SPZ.

Klasický SOS přímou změnu přístupového klíče nedovolí, takže by uživatel musel nejdříve vytvořit novou větu, do ní přepsat všechny informace o vozidle a nakonec nezapomenout původní větu zrušit. Tak komplikovaný postup však poskytuje příliš mnoho příležitostí k zavlečení chyby.

Naštěstí i začínající programátor snadno dokáže tyto nebezpečné činnosti zautomatizovat. Musí si však uvědomit, že je to vhodné a potřebné.

Hůř řešitelné potřeby však vzniknou například při ročním vyhodnocování výkonů vozidla, zvláště v případě, že ke změně SPZ došlo v průběhu roku několikrát.

Komplikace přinese změna SPZ také tehdy, má-li vozidlo zpracován plán práce na celý měsíc a pracovník pověřený zpracováním plánu je třeba na školení.

Nepřipustíme odbytí naznačeného problému krátkozrakou námitkou, že naprosto postačí zvolit za přístupový klíč něco stabilnějšího než je SPZ. V tomto případě by to možná i bylo řešení, obecně ale přece nelze přecházet na nový přístupový klíč jenom proto, že nám nelíbí ošetřovat změny hodnot klíče původního.

Komplikacím se nevyhneme ani tehdy, musíme-li používat číselník pro rozdílné úlohy (plán-operativa-evidence) : pro evidenci zřejmě budeme potřebovat i údaje o vozidlech, která v dané okamžiku už byla z provozu vytazena, a proto nesmějí být v číselníku pro operativní řízení provozu. Naopak, pro plánování zase musíme mít informace o vozidlech do provozu ještě nezafazovaných, tedy v číselnících pro operativní řízení a evidenci rovněž nepřipustných.

Poměrně snadno nahlédneme, že zdánlivě nejjednodušší řešení (používání různých číselníků pro jednotlivé účely) je vlastně jenom zbavením se zodpovědnosti na úkor uživatele. Převážná většina informací se zřejmě v každém takovém dílčím číselníku bude opakovat, takže nejenom zkomplikujeme údržbu o synchronizaci souborů, ale také obsadíme víc paměti.

Na podstatě věci nic nezměníme ani bohorovným tvrzením, že za chyby uživatele programátor nemůže odpovídat.

Pravý opak je pravdou : programátor musí vědět, že kde nechá něco dělat člověka, tam se naprosto zákonitě musí vyskytnout chyby. Rozumný programátor proto omezuje činnost uživatele na nevyhnutelné minimum. Odpovídáme za ty, které jsme k sobě připoutali.

Nepoddám se ale temným moralistickým pudům (přestože by možná ani referát na téma "Programátor a profesionální čest" nebyl nepodnětný) a vrátím se k původní problematice - práce s číselníky.

Zamysleme se nad číselníky, tentokrát z hlediska jejich distribuovaného zpracování.

Při práci na libovolné úloze musíme dát každému možnost práce bez rušivých zásahů jiných uživatelů. Nemožou pracovat, když někdo mi bude bez mého vědomí měnit, nebo dokonce mazat data vzápětí po tom, co jsem je vložil do počítače.

Umožníme-li každému provádění změn v číselníku, musíme zajistit, aby tyto změny nenabourávaly práci někoho jiného. Moje změny tedy budou platit jenom pro mne.

Číselníky ale musejí plnit také svoji integrační funkci v systému zpracování dat. Mají-li se lidé spolu domluvit, je nutné, aby používali pro stejné jevy stejné pojmy. Bude zřejmě nutné zajistit, aby jeden z uživatelů (správce) každého číselníku měl právo a povinnost kontrolovat a podle potřeby upravovat zásady ostatních uživatelů.

Vzniká zde otázka, za jakých předpokladů bude správce číselníku schopen požadavky ostatních uživatelů smysluplně koordinovat.

Z hlediska činnosti správce má v číselníkové větě každá položka buďto exkluzivní, nebo aditivní způsob aktualizace. Exkluzivně aktualizovatelná je položka, jejíž každá změna vylučuje všechny jiné změny. V našem číselníku vozidel měly všechny údaje exkluzivní charakter. Aditivní aktualizací rozumíme postupné kumulování všech oprávněných změn. Tento charakter by měla například položka "hodiny v provozu".

U položek exkluzivních musí správce číselníku určit, který z uživatelů měl pravdu. Může to kvalifikovaně provést jen za předpokladu, že má k dispozici jak údaje o požadované změně, tak informace o tom, kdo, kdy a proč změnu navrhuje. V některých případech může správce potřebovat také informace kdy a k čemu byly použity původní hodnoty údajů, které mají být aktualizovány. Tyto všechny informace by zřejmě měly být připojeny k jednotlivým větám číselníku.

Položky aditivního charakteru mohou být aktualizovány několika uživateli současně, přičemž výsledná diference mezi původní a novou hodnotou položky by měla být součtem diferencí u jednotlivých uživatelů. Ke kvalitnímu promítnutí aditivní změny do číselníku správci zřejmě nepostačí znalost jednotlivých diferencí, protože několik uživatelů může stejný údaj promítnout do změn současně.

Do věty číselníku obecně nelze uložit všechny informace pro posouzení správnosti aditivní aktualizace. Má-li správce číselníku ručit za jejich kvalitu, musí mít k dispozici informace srovnatelné s daty, z jakých vyšli jednotliví uživatelé. V opačném případě by bylo možná lepší delegovat odpovědnost za aktualizaci aditivní položky na uživatele.

Průběh práce uživatele potom může být následující :

- při zjištění potřeby provede uživatel změnu číselníku dle vlastní úvahy, bez nutnosti na cokoli čekat,

- ve vhodný okamžik předá uživatel svůj číselník správci, k provedení koordinace a potvrzení změn,
- zároveň s číselníkem předá správci i případné podklady, umožňující provedení změn aktivního charakteru,
- výsledky získané s užitím správcem dosud nepotvrzených číselníků dostane uživatel s označením za "DOČASNÉ",
- výsledky získávané s použitím číselníků, které správce už potvrdil, budou automaticky označeny jako "TRVALÉ",
- uživatel musí respektovat, že zásah správce může vyvolat rozdíly mezi výsledky, označenými "DOČASNÉ" a "TRVALÉ".

Protože nic není zadarmo, bude nutno zavést do každé věty číselníku následující pomocné informace :

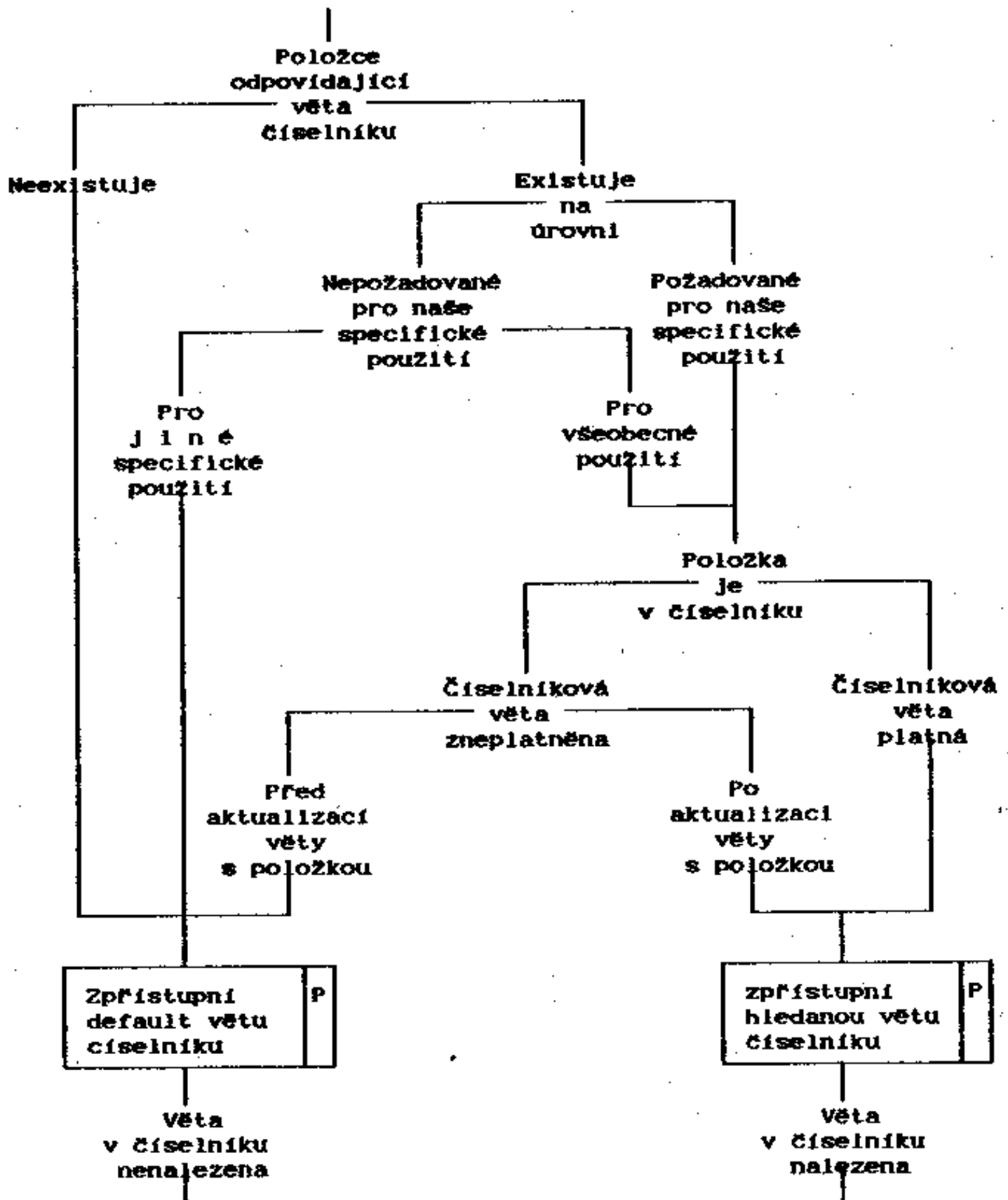
- správcem přidělená "TRVALÁ" hodnota přístupového klíče
- datum poslední aktualizace (vytvoření, zneplatnění) věty
- kód pracovníka, který poslední aktualizaci provedl
- datum posledního využití údajů z číselníkové věty
- kód aplikace, která posledně číselník použila
- příznak případného zneplatnění věty
- kód způsobu použití věty, např: 4 ... evidence, 3 ... operativa, 2 ... plánování, 1 ... bez omezení, 0 ... default-hodnoty

Navrhovanou koncepcí zpracování položky jsem se pokusil znázornit na obrázcích 1 a 2.

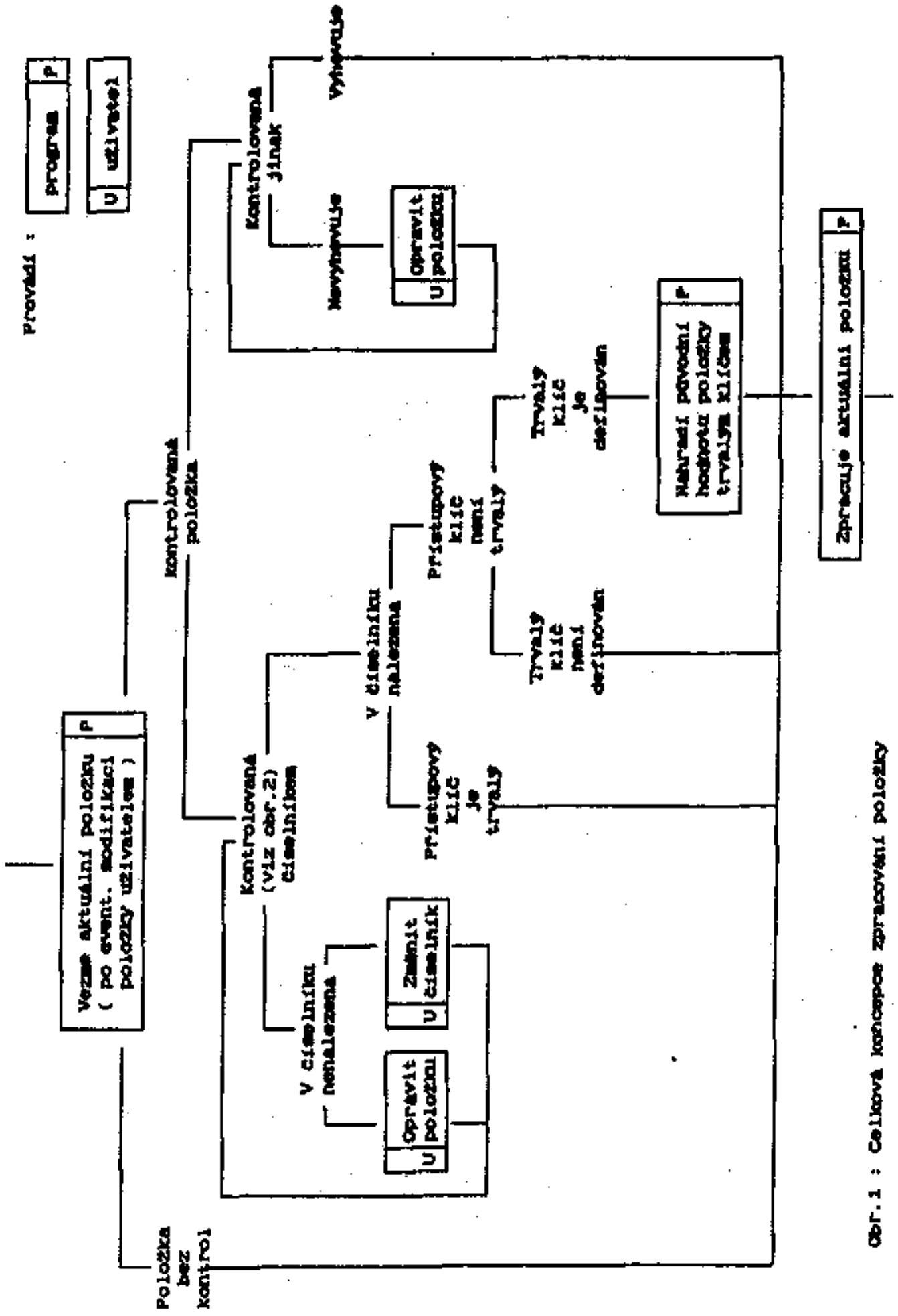
Snad ani není potřeba zdůrazňovat, že největší výhodou používání číselníků je možnost uložení opakovaně používaných informací na jediném místě - tedy v číselníku. Kopírováním údajů z těla věty číselníku do jiných souborů o tuto výhodu přicházíme. Ovšem pozor při archivování dat. Nezapomenout na archivaci příslušných číselníků.

Vím, že pokud jde o jednotlivé detaily, nedověděli jste se ode mne prakticky nic, co byste sami už nevěděli. Hlavním cílem tohoto referátu však bylo upozornit, že pro stromy nevidíme les. Že příliš často vytváříme programy, které jsou pro uživatele spíš překážkou, než pomocníkem. Nebo se pletu?

Autor: Ing. Emil K o p ě i l, Vítkovické stavby s. p.
1. úderky 41, 703 00 Ostrava, tel. 069-3628 kl. 355



Obr.2 : Koncepte kontrolování položky číselníkem



Obr.1 : Celková koncepce zpracování položky