

Problémy tohoto stavu

Z hlediska studenta:

- zbytečné zdržování při příchodu či odchodu z prostoru počítačových učeben:
 - nutnost čekat na obsluhu, až vybere kartu (při příchodu) či až kartu vyhledá a vrátí (při odchodu),
- nutnost opustit učebny ihned po skončení vyučování:
 - o přestávkách jsou počítačové učebny uzavřeny, protože vyučující požadovali, aby jejich učebny po přestávce byly obsazeny jen studenty, kteří patří do jejich kurzu, což ovšem nebylo možno zajistit. Proto bylo zavedeno opatření, jež přikazovalo studentům čekat na svého vyučujícího před vchodem do prostoru počítačových učeben. Vyučující si své studenty odváděl do učebny sám.

Z hlediska vyučujících

- do učebny „proklouzl“ mezi přichozími i studenty, kteří na výuku nepatřili. Vyučující ztrácel část vyučovacího času tím, že se tyto nepoctivé studenty snažil přimět k opuštění učebny.

Z hlediska pracovníků obsluhy

- nepříjemná práce s identifikačními kartami - neustálé třídění či vyhledávání identifikačních karet podle příjmení (už při počtu 100 karet je velmi zdoluhavé a monotónní),
- střety se studenty, kteří nechťejí kartu odevzdat,
- střety se studenty, kteří nechťejí opustit učebny o přestávkách,
- celková monotónnost práce - pracovník musel být neustále „ve střehu“, připraven na různou minu ochoty přichozího studenta odevzdat kartu.

Z hlediska Výpočetního centra

- vysoká fluktuace pracovníků obsluhy- kvůli nepříjemné práci,
- nedostatečná evidence uživatelů.

Definice cílů systému

Cíle systému EDIK a jejich priority byly stanoveny takto:

1. **Odstranit nepříjemnosti z pohledu obsluhy** - odstranit vybírání identifikačních karet, zajistit, aby nebylo nutno bránit studentům ve vstupu na učebny, v nichž ještě není vyučující, zajistit automatické uvolňování učeben před výukou.
2. **Odstranit nepříjemnosti z pohledu vyučujících** - zajistit rezervaci učebny jen pro ty studenty, kteří jsou zapsáni na předmět, jehož výuka probíhá v daném okamžiku na dané místnosti, ostatní (nezapsaní) studenti nesmí v dané učebně na počítači pracovat.
3. **Minimalizovat „administrativní“ požadavky na studenty** - systém by neměl zbytečně omezovat studenty v jejich práci na počítačích - veškeré kontroly by měly být maximálně rychlé (řádově desítky sekund), systém by měl být schopen okamžité aktualizace nejdůležitějších databází, maximální lhůta pro promítnutí změny reality by měla být 24 hodin., v případě pádu systému nesmí být ohrožena výuka či individuální práce.

Současný stav systému

V současné době se provádějí při přihlašování studenta do počítačové sítě VŠE následující kontroly:

1. *Kontrola protažené karty.* Kontrolu karet je možno zapnout či vypnout pro následující skupiny místností:
 - ve všech místnostech,
 - na studovnách,
 - na učebnách, pokud na nich neprobíhá výuka,
 - na učebnách v průběhu výuky.

Studentovi je znemožněna práce v počítačové síti, pokud je kontrola zapnuta a pokud student v patře, v němž se snaží přihlásit k počítačové síti, neprotáhne svou identifikační kartu čtečkou u vchodu.

2. *Kontrola příslušnosti studenta k vyučovanému předmětu.* Kontrolu lze zapnout či vypnout najednou pro všechny učebny a nebo výběrově jen pro určitou místnost a čas (je možné zapnout či vypnout kontrolu pro každou čtvrt hodinu).

Pokud na učebně probíhá výuka, kontroluje se, zda přihlašovaný student je zapsán na tuto výuku či nikoliv. Pokud je kontrola zapnuta a v místnosti, v níž se student přihlašuje, probíhá výuka a pokud student na ni není zapsán, je mu znemožněn přístup do počítačové sítě VŠE.

Systém je schopen obsloužit 8-10 požadavků za sekundu, mimo špičku přichází požadavek zhruba každé 3 sekundy, ve špičce 5-7 za sekundu.

Splnění cílů k 1. 3. 1997

1. Odstranit nepříjemnosti z pohledu obsluhy

- Obsluha se nemusí o kontrolu karet zajímat, EDIK provádí kontrolu automaticky, studenti jsou nuceni si kartu protáhnout čtečkou.
- Kontrola příslušnosti k výuce je prováděna už o přestávce před výukou. EDIK provádí kontroly na příslušnost studenta ke kurzu automaticky, obsluha učeben tedy nemusí držet studenty přede dveřmi do příchodu vyučujícího.
- Uvolňování učeben před výukou není automatizované, zkušební provoz prokázal, že tuto funkci není třeba zajišťovat.

2. Odstranit nepříjemnosti z pohledu vyučujících

- vyučující mají zajištěno, že na jejich výuce budou přítomni jen ti studenti, kteří mají předmět zapsán (vyučující může – pokud mu nevadí individuálně pracující studenti – nechat tuto kontrolu vypnout na jednu hodinu či na více týdnů, resp. na celý semestr - kombinací je mnoho).

3. Snížit vysokou fluktuaci pracovníků výpočetního centra

- vzhledem ke krátké době plného provozu systému EDIK není možné v současné době splnění cíle hodnotit. Pracovníci obsluhy jsou s novým systémem spokojeni.

4. Minimalizovat „administrativní“ požadavky na studenty

- student je povinen protáhnout si při vstupu do počítačových učeben svou identifikační kartu čtečkou. Pokud kartu ztratí, je nucen zajistit její zablokování a obdrží „Dočasné povolení vstupu“, které nahrazuje protažení jeho identifikační karty po dobu jednoho měsíce. Veškeré tyto úkony byl student povinen dodržovat již dříve, ke zvýšení „administrativních“ požadavků nedošlo, pouze se začalo s jejich kontrolou. Naopak některé nepříjemnosti ubyly: čekání na vyučující před začátkem výuky, povinnost uvolnit učebny ihned po výuce (uvolnění učeben před další výukou)

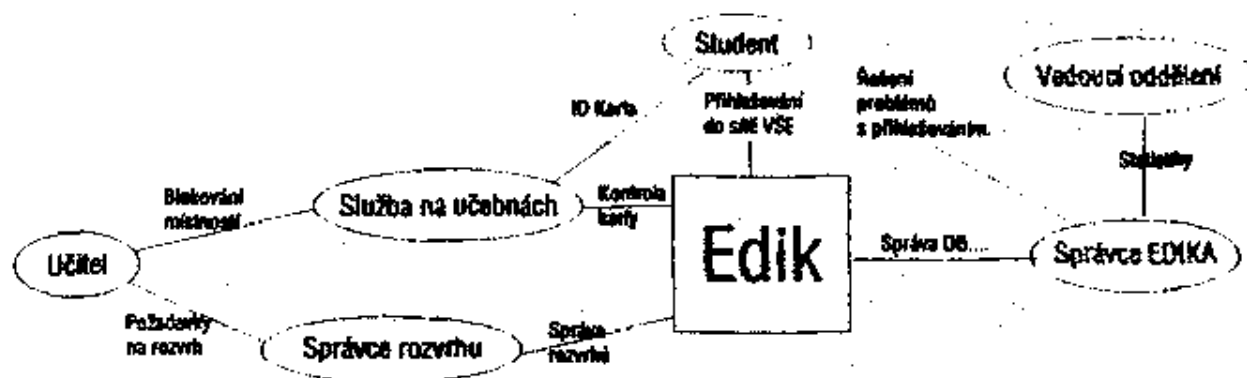
Postup návrhu

Při použití OOMT prochází projekt zhruba těmito fázemi:

- *Analýza* - tvorba logického řešení úlohy - bez ohledu na nároky a na možnosti techniky.
- *Systémový design* - rozvržení logických částí systému na technické prostředky.
- *Objektový design* - podrobné rozpracování jednotlivých částí (aplikaci), výstupem je předpis pro programátory.
- *Implementace* - kódování programů, ladění.

Analýza - model okolí systému

Analýzou problému jsem dospěl k následujícímu modelu struktury okolí systému:



Studenti komunikují:

- se *Službou na počítačových učebnách* při příchodu (protážení a kontrola ID karty),
- s *programem BRÁNA*, který získává informace o uživateli (uživatelské jméno a jméno počítače, na kterém se tento uživatel hlásí do počítačové sítě); program BRÁNA požádá jádro systému EDIK o kontrolu údajů vzhledem k protážení identifikační karty a vzhledem k rozvrhu výuky; v případě neprotážené ID karty či v případě, že student nepatří na výuku, BRÁNA znemožní studentovi přihlášení do počítačové sítě VŠE,
- se *Správce EDIKa* v případě, že mají pocit, že systém nereaguje správně (korekce chyb v datech).

Učitelé přichází do styku s EDIKem prostřednictvím:

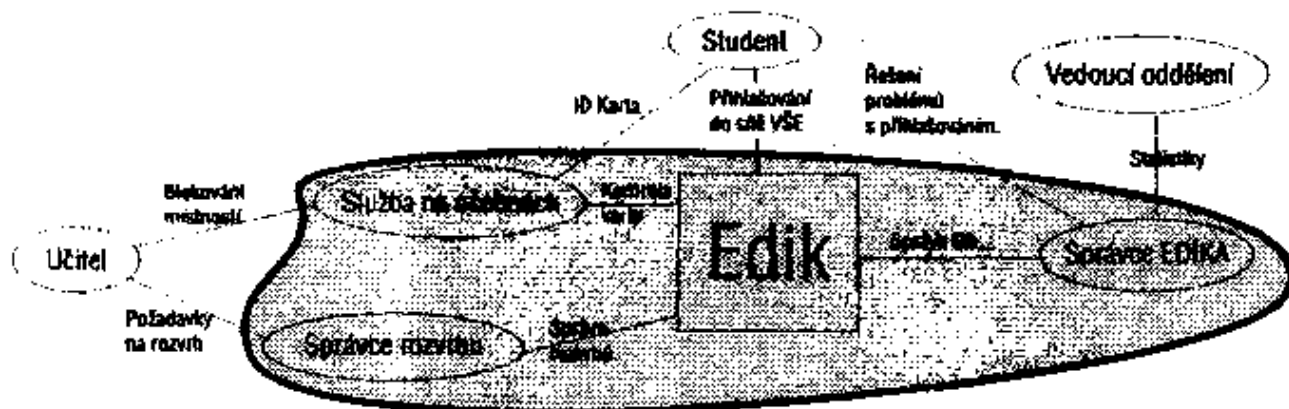
- *Správce rozvrhů*, který může přesouvat, mazat či přidávat výuku dle požadavků vyučujících,
- *Službou na učebnách* v případě, že chtějí okamžitě zablokovat či odblokovat učebny pro studenty nezapsané na jejich výuku.

Vedoucí oddělení vyžaduje od *Správce EDIKa* statistiky, např. o obsazenosti kurzů studenty, a pod.

Při podrobnějším rozboru okolí informačního systému EDIK je patno, že funkce „*Správce rozvrhů*“, „*Služba na učebnách*“ a „*Správce EDIKa*“ je třeba zahrnout do systému.

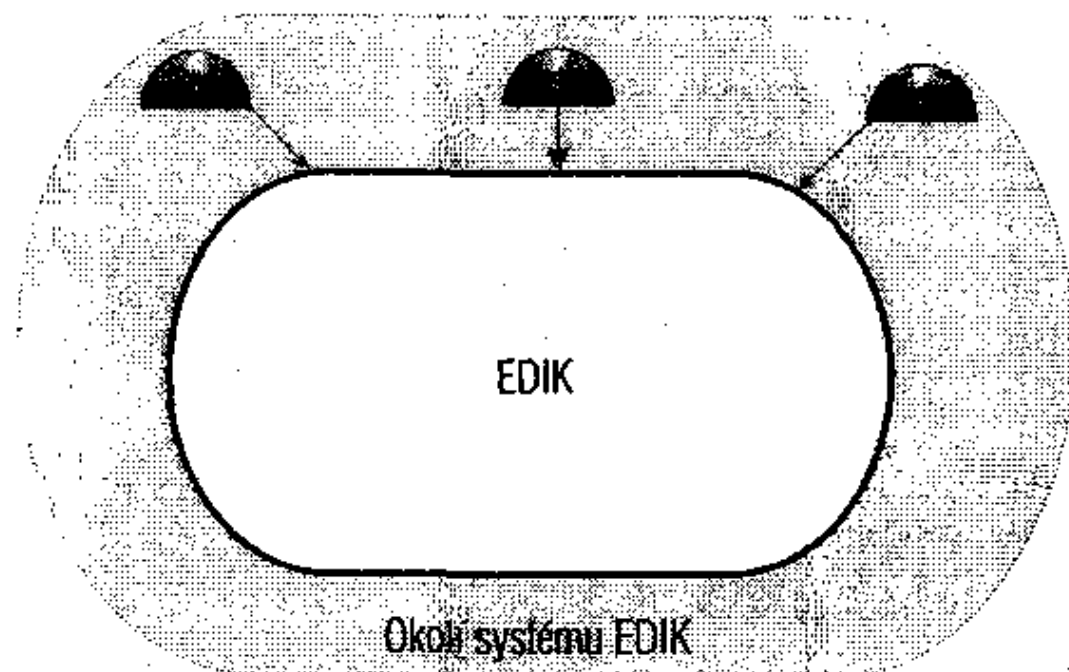
OOMT mi umožnila pracovat s lidmi (přesněji s funkcemi, které budou pracovníci zastávat) stejným způsobem, jako kdybych pracoval s nějakými částmi programu. V průběhu tvorby logického modelu (analýzy) jsem nemusel rozlišovat mezi lidmi, stroji a programy.

Rozsah celého systému EDIK (tj. softwarového jádra a obslužných pracovníků) je naznačen zde:



Architektura systému v rámci analýzy

Na nejvyšší úrovni jsou shrnuty informace o okolí systému. Významnými prvky okolí jsou: Učitel, Student a Vedení.

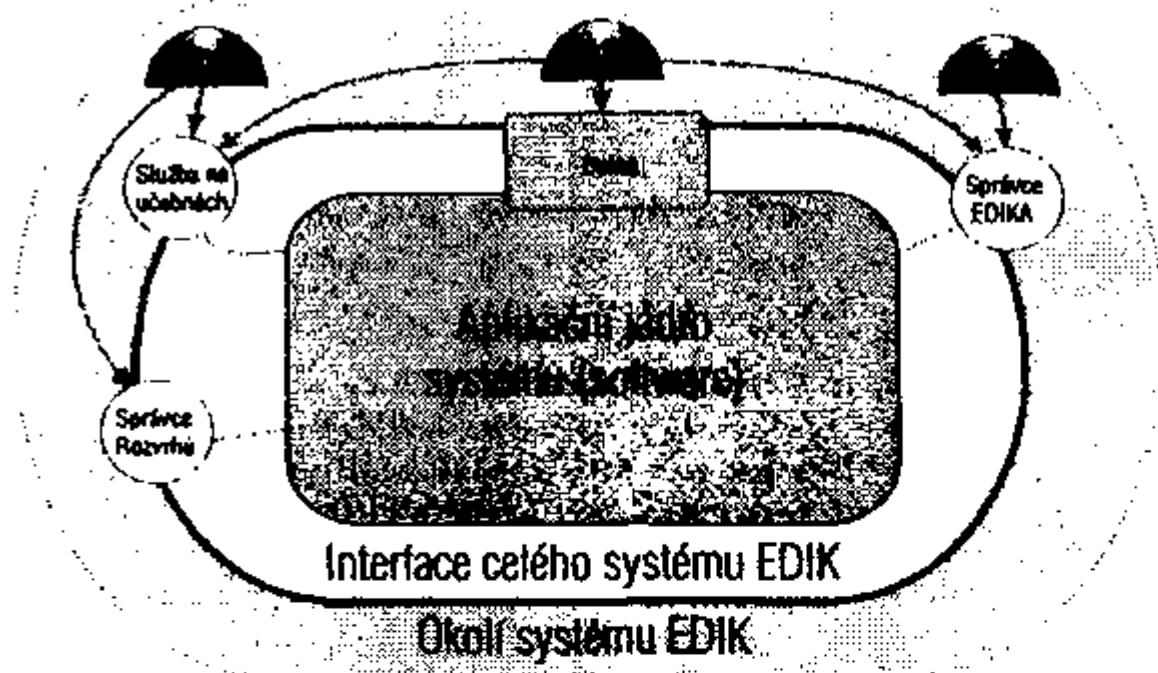


Na jemnější úrovni pohledu se systém EDIK rozpadá na třídy pro styk s okolím (interfaceové třídy) a na třídy vnitřní, pro uživatele skryté, které jsou zatím zobecněny v aplikační jádro systému.

Interfaceovými třídami jsou:

- lidé (pracovníci oddělení) ve funkcích Služba na učebnách, Správce rozvrhů a Správce EDIKa,
- programy - program BRÁNA, který se spouští při přihlašování studenta do sítě VŠE (podrobněji viz výše). Program BRÁNA je jediná aplikace, se kterou přicházejí uživatelé do přímého styku.

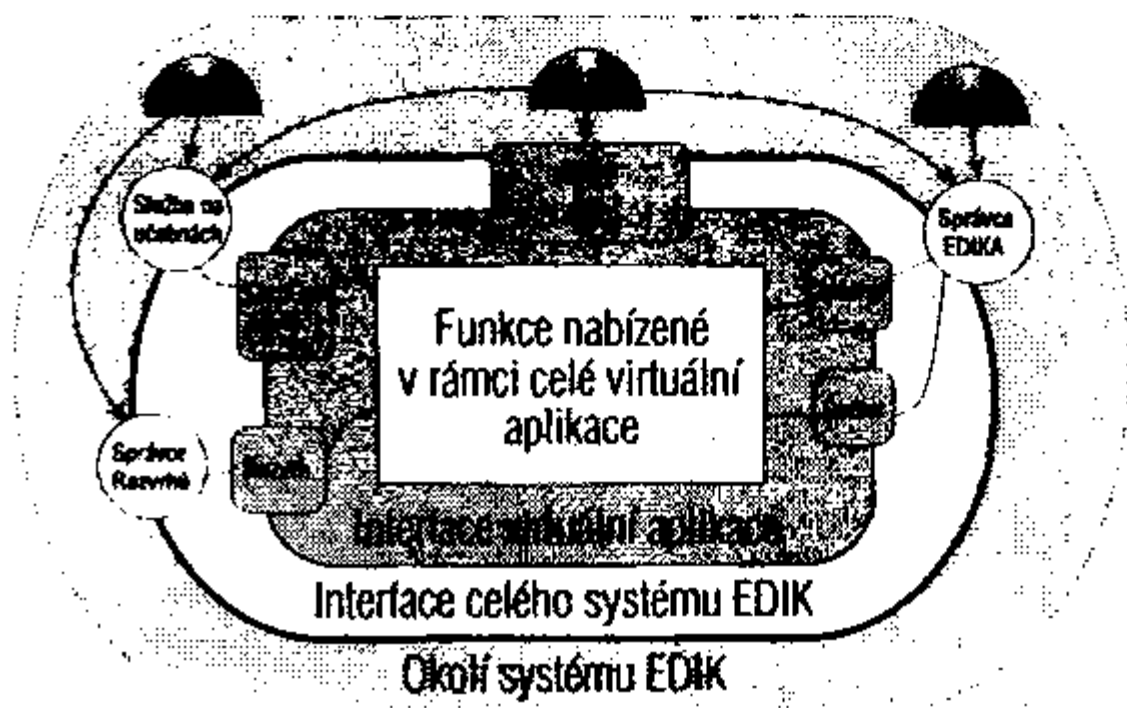
Aplikační jádro je množina programů, které umožňují chod systému.



Při dalším rozboru se aplikační jádro systému rozpadá na následující programy:

- *Server* - spravuje databáze studentů, provádí některé testy,
- *Inform* - aplikace pro vzdálenou údržbu serveru (vč. správy databázi),
- *Rozvrh* - aplikace pro práci Správce rozvrhů,
- *Cardgate* - aplikace pro práci Služby na učebnách.

Na výše uvedené aplikace je možné pohlížet jako na různé interface jedné virtuální aplikace. Pomocí těchto interface uživatelé vyvolávají funkce, které virtuální aplikace nabízí.



Systémový design a změny v architektuře

V rámci systémového designu je třeba rozmístit třídy na technická zařízení, resp. přiřadit funkce lidí skutečným pracovním místům na pracovišti.

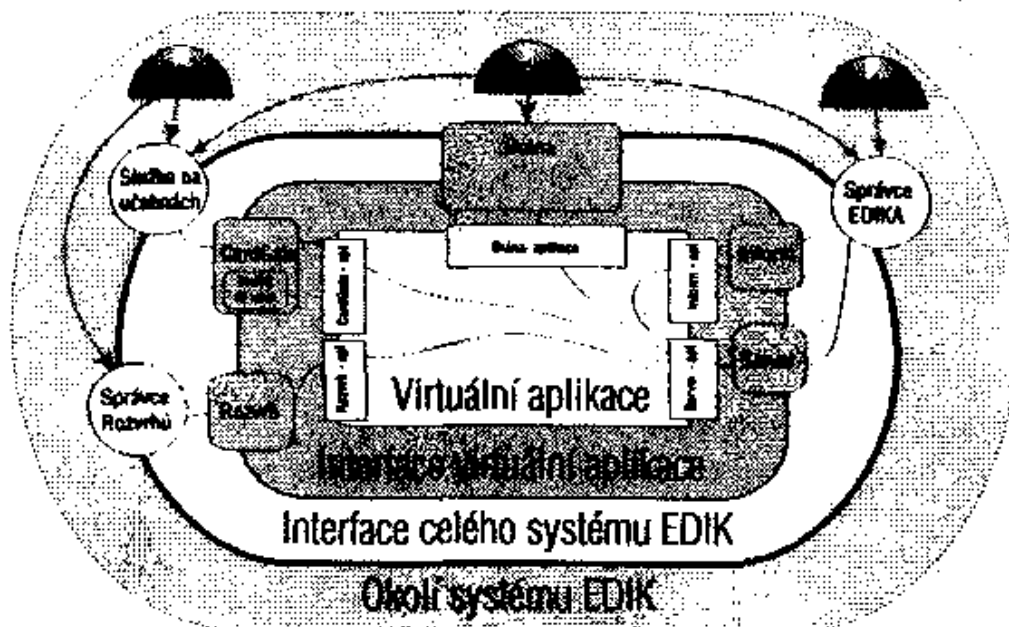
V případě „lidských“ tříd šlo u systému EDIK pouze o úpravu specifikace již existujících pracovních míst v oddělení provozu počítačových učeben a studoven.

Složitější bylo rozložení aplikací na počítače. Byla zvolena architektura klient – server. Pro komunikaci je použit protokol IPX. Operační systém MS-DOS. Uživatelské rozhraní je postaveno na bázi TurboVision.

Vzhledem k tomu, že na síti VŠE se budou používat také počítače pod Windows'95 či NT, bylo třeba zajistit snadný převod aplikací do prostředí Windows. Proto bylo při návrhu systému rozhodnuto oddělit v každé aplikaci třídy interfaceové od tříd výkonných. Oddělení bylo provedeno v maximální možné míře – interfaceové třídy zasílají požadavky na vyvolání nějaké funkce (např. zkontroluj uživatele, najdi záznam v databázi apod.) druhé polovině aplikace - třídě výkonné. Ta zajišťuje provedení funkce a navrácí interfaceové třídě návratovou hodnotu.

V této souvislosti jsme se rozhodli vytvořit jakousi virtuální aplikaci, která by byla rozložena na více strojích, její funkce by bylo možné umístit na kterýkoli z těchto strojů. Princip je jednoduchý - pokud výkonná třída aplikace obdrží od své interfaceové třídy nějaký požadavek, pokusí se jej obsloužit sama. Pokud nemá ve svém kódu danou funkci, obrátí se (pomocí protokolu IPX) na server, který se pokusí obsloužit funkci sám. Pokud není výkonná třída serveru schopna požadavek obsloužit, předá jej k obsluze své interfaceové třídě (jedním z požadavků může být také požadavek na obsah pracovní obrazovky serveru). Výhledově se předpokládá, že pokud nebude existovat obslužná funkce pro daný požadavek ani v interfaceové třídě, odešle server dotazujícímu se klientovi informaci, na který jiný počítač je možno se obrátit s daným požadavkem. Tímto způsobem jsme zajistili možnost optimalizovat výkon systému tím, že se některé časově náročné funkce ponechají na klientovi a jiné se naopak delegují na server.

Rozetnutí aplikací na dvě části - interfaceovou a výkonnou má ještě další efekt. Při přechodu na Windows stačí změnit část interfaceovou. Výkonná část může zůstat beze změny, je tak zajištěna poměrně vysoká stabilita logického řešení úlohy i při přechodu na jiné interfaceové prostředí.



Objektový design

Ve fázi objektového designu se jedná o podrobné rozpracování tříd, navržených v předchozích krocích.

U tříd, které budou realizovány ve formě pracovních míst (např. Správce Rozvrhů), se v rámci objektového designu definují organizační pravidla (práva a povinnosti) pracovníka, který takové pracovní místo obsadí. Je zajímavé, že je třeba definovat také práva a povinnosti tříd z okolí (Studentů, Učitelů). Patří tedy i tyto třídy do systému?

V případě aplikačních tříd se jedná o podrobný rozbor jednotlivých aplikací, navržených v průběhu analýzy a systémového designu. V případě EDIK byl proveden v rámci objektového designu celého systému kompletní postup dle OOMT pro každou aplikaci. Vstupním zadáním těchto „subanalýz“ byly popisy tříd ze systémového designu celého systému. Při rozboru některých částí aplikací (především interfaceových modulů) by bylo možno opět provést celý postup OOMT. Vzhledem k trivialitě modulů však toho nebylo zapotřebí.

Implementace

Implementace celého systému probíhala tak, že byly nejprve implementovány aplikace (Server, Inform, Rozvrh, Brána, Cardgate), poté byli zaškoleni pracovníci, informováni studenti a vyučující. Následně byl systém provozován po dobu několika týdnů v poloprovozu a od počátku března byl systém uveden do plného provozu.

Zhodnocení výhod a nevýhod plynoucích z použití OOMT

Použitím OOMT jsem získal především nástroj pro podrobný průzkum reality. OOMT nabízí několik diametrálně odlišných pohledů na navrhovaný systém. Tuto vlastnost si cením nejvíce. Další velmi užitečnou charakteristikou OOMT je důsledné vedení návrháře k promyšlení zvoleného postupu návrhu – zda není ve slepé uličce, zda není svázán nějakým předchozím rozhodnutím zbytečně. Iterativnost celé metodiky, průhlednost všech vrstev návrhu a průhlednost mezi návrhem a implementací je silným nástrojem pro nalezení a odstranění chyb.

Hierarchický rozklad problému umožňuje rozdělit od určité fáze práci mezi více vývojářů - v případě systému EDIK pracoval jeden člověk na interfaceových třídách a druhý na třídách výkonných, aniž by nastaly potíže se vzájemným přepisováním kódu.

Objektová implementace aplikací nabízí velkou úsporu práce díky sdílení kódu mezi aplikacemi. Výkonné třídy klientských aplikací systému EDIK jsou v podstatě shodné. Interfaceové třídy jsou rozčleněny na základní skelet a sadu specializovaných modulů (pohledy na databázové tabulky, editace a prohlížení rozvrhů, zobrazování informací o identifikační kartě). Základní skelet je shodný jak pro klienty, tak pro server. Také moduly mohou být připojeny jak k interface serveru, tak také k interface klientů. Tím je zajištěn jednotný způsob ovládání všech aplikací. Pokud je na více místech dostupná stejná funkce, má vždy pro uživatele shodný interface, přestože může jít o dva různé programy (např. prohlížení a editace databázových tabulek musí být přístupná jak na serveru přímo, tak také v klientovi, který je k dispozici pracovníkovi ve funkci Správce EDIKa (tj. v programu Inform – viz výše kapitola Analýza).

Neméně příjemným je fakt, že v případě pečlivé práce na modelech v průběhu analýzy a systémového i objektového designů je hotová velká část kvalitní, podrobné dokumentace.

OOMT je náročné na psychickou otužilost návrhářovu – je dosti často nutné přepracovat část modelů, mnohdy jen pro zajištění konzistence mezi jednotlivými vrstvami návrhu. Za zanedbání takových úprav se platí dezorientováním návrháře v okamžiku, kdy se k dané problémové oblasti vrací po nějaké delší době (mě stačil měsíc).

Návrhář pracující s OOMT musí sám hledat cestu sítí modelů, které metodika nabízí. Neexistuje jediný správný postup. Počet modelů a rekurentnost návrhu (např. objektový design celého systému je souhrnem kompletní analýzy, systémového a objektového designu některých částí), sice nabízejí možnost vytvoření ideálního postupu pro každý projekt, ale je třeba takový ideální postup napřed vytvořit.

OOMT lze výhodně použít pro špatně strukturované problémy, pro rozsáhlé projekty s velkým podílem interakce mezi uživateli a navrhovaným systémem. Lze jej však použít i pro úlohy malé, přehledné. Domnívám se, že v takovém případě budou metodiky strukturovaného návrhu stejně efektivní, a možná i efektivnější než OOMT.