

PEDAGOGICKÉ VZORY VE VÝUCE SOFTWAREVÝCH INŽENÝRŮ

Miloš Kudělka^a
Vladimír Sklenář^b

Katedra informatiky PřF UP Olomouc, Tomkova 40, 779 00 Olomouc, ČR
milos.kudelka@upol.cz, vladimir.sklenar@upol.cz

Abstrakt

Naše výukové zkušenosti pokrývají celý proces přípravy softwarového inženýra od základní výuky algoritmizace a programování, přes výuku technologií, až po tvorbu reálných softwarových projektů. Obdobou návrhových vzorů pro uplatnění ve výuce jsou tzv. pedagogické vzory, které nacházejí uplatnění nejen v běžné vysokoškolské výuce, ale především při dalším vzdělávání pracovníků v softwarových firmách v souvislosti se zaváděním nových technologií a metodik. Známé pedagogické vzory se nám ukazují jako použitelné v naší vlastní výuce, některé z nich už sami dlouho používáme a jejich přesná formulace nám pomáhá vidět další souvislosti, které nám pomáhají zvýšit efektivitu výuky. V článku diskutujeme využití pedagogických vzorů v jednotlivých fázích procesu výchovy softwarového inženýra, uvádíme využitelné vzory a popisujeme jejich nasazení v této specifické oblasti.

Co jsou vzory?

V sedmdesátých letech byly architektem Christopherem Alexandrem formulovány vzory pro řešení úloh v architektuře (1). Vzor je formulován nikoli jako nápad, jak řešit problém, ale jako popis a zobecnění určité zkušenosti, která vede k postupu, jak problém řešit (výhodiskem vzorů je praktická zkušenost, nikoli pouhý nápad, jak problém řešit). Návod řešení pak musí být tak pružný, aby se vzor dal aplikovat opakovaně a aby přitom výsledky nebyly stereotypní. Vzor je stručný text, který popisuje náčrt problému a popis jeho obecného řešení v konkrétních souvislostech.

Alexandrova formulace vzoru: „Každý vzor je pravidlo, které vyjadřuje vztah mezi jistou souvislostí, problémem a řešením.“

Obecně se dá říci, že vzory

- zachycují zkušenost odborníků v určité oblasti,
- umožňují jednoduchou komunikaci mezi odborníky,
- řeší přirozený, opakovaně se vyskytující problém,
- spolu mohou souviset resp. mohou na sebe navazovat

... a vzory nejsou

- momentální nápady autora,
- nějaké triky,
- popisy jediného způsobu řešení,
- popisy, které nevnášou shodu mezi odborníky.

Vzory samozřejmě existují v různých oblastech lidské činnosti, každý člověk ve své práci dříve nebo později začne používat jisté (někdy svazující) stereotypy, které vedou ke zvýšení efektivity při řešení problémů. Vzory tyto stereotypy pojmenovávají a formalizují tak, že při

jejich použití je zdůrazněna obecnost a pružnost. Z tohoto pohledu se pak určitá skupina vzorů může stát cenným slovníkem pro popis problémů v dané oblasti.

Ve kterých oblastech se vzory vyskytují?

Softwarové návrhové vzory. Inspiraci z architektury převzali v devadesátých letech lidé z oblasti informatiky a počítačů a dodnes stanovili desítky celosvětově uznávaných vzorů, které se aplikují v oblasti analýzy, návrhu a programování aplikací (např. 2, 3, 4).

Vzory pro řízení organizace nebo procesů v organizaci. Nejdříve se opět jednalo o organizace zajišťující vývoj software, což opět vyplývá z obecného pohledu lidí z oblasti informatiky na řešení problému a z pokusů opakovaně aplikovat známý postup v jiných souvislostech (např. 5, 6).

Pedagogické vzory. Mnoho odborníků z oblasti informatiky také současně přednáší na konferencích a vede semináře v oblastech, kterým se věnují. Přitom většina z nich nemá pedagogické vzdělání. Proto asi není překvapivé, že se na konferencích a v knihách o vzorech objevily i vzory použitelné ve výuce (např. 7, 8, 9). Pedagogickými vzory se zabývá několik skupin (10, 11) a jsou určeny především lektorům bez pedagogického vzdělání.

V některé z uvedených oblastí může existovat více skupin vzorů a samotných vzorů pak desítky. A to od jednoduchých a velmi často použitelných až po složitější a omezeně použitelné.

Struktura vzoru

Aby mohly být vzory popsány jednotnou formou, existují určité zákonitosti pro jejich zápis, které se mnohdy liší rozsahem, nicméně je různými autory ve shodě přijímána minimálně následující struktura:

Jméno: Jednoznačný název vyjadřující přímo (nebo zvláštním, mnohdy velmi nápaditým způsobem opisu) určení vzoru.

Problém: Stručná charakteristika úlohy, která se řeší.

Souvislosti: Popisuje oblasti a situace, ve kterých se vzor používá.

Vlivy: Příznaky, které jsou charakteristické pro použití vzoru nebo které jsou vzorem ovlivněny.

Řešení: Popis obecného přístupu k řešení problému a také upozornění na úskalí.

Důsledky: Očekávané výsledky aplikace řešení.

Příklady použití: Ukázky aplikace vzoru na konkrétní problém. Často je příklad uveden před popisem řešení.

Odkazy: Často použití jednoho vzoru předchází nebo následuje vzor jiný, uvedou a vysvětlí se souvislosti.

Jazyk vzorů

Vzhledem k tomu, že vzor se nikdy nevyskytuje v daných souvislostech osamoceně, bývají souvislosti popsány řečí, jazykem vzorů. Vzory pro danou souvislost tvoří uzavřený celek, ve kterém jsou organizovány do návazností a skupin, které mohou mít společný rozsah nebo čas, ve kterém existují. Jazyk vzorů není jen katalog vzorů, ale je to komplexní pohled na použití

vzorů v jedné (byť široké) oblasti. Vzory bývají aplikovány v sekvencích, kontext jednoho vzoru může být důsledkem jednoho nebo více jiných vzorů.

Jazyk vzorů tedy přesně definuje kontext, vzory v tomto kontextu a vztahy mezi nimi. (Podobně jako např. čeština poskytuje mechanismus skládání vět ze slov, jazyk vzorů umožňuje složit řešení složitějšího problému s jednotlivých vzorů). Důležitým faktorem pro popis vzorů je to, že i přesto, že kontext může být velmi rozličného rozsahu (od budování měst až po pasivitu studentů na semináři) a času pro použití (od let až po minuty), rozsah jejich popisu zůstává zhruba stejný (v rámci jedné nebo několika stran). Tento podstatný rys umožňuje na relativně krátkém prostoru dát k dispozici jazyk, který pokrývá určitou oblast a umožní dobře formulovat problémy, souvislosti a řešení, které se v této oblasti vyskytují.

V naší výuce se zabýváme vzory již delší dobu (12, 13, 14, 15) především v předmětech, které vedou k teoretickým znalostem (softwarové inženýrství) a praktickým dovednostem (softwarový projekt). Oblast, která je nám nejbližší jsou vzory pro návrh. Nicméně jsme sáhli i po vzorech z jiného prostředí (16).

Výuka vývoje software a vzory

Na internetových stránkách (11) lze najít velmi ucelený rámeček pedagogických vzorů (celkem 60 vzorů) aplikovatelných ve výuce studentů informatiky, přesněji řečeno budoucích vývojářů software (využívajících především objektové technologie). Ve zde popsaném katalogu lze najít opravdu bohatou zásobu myšlenek, které zkušeného učitele nepřekvapí, spíše mohou ukázat nové a přesnější pohledy na vlastní zkušenost. Autoři tohoto WEBu zmiňují fakt, že většina vzorů uveřejněných na tomto místě byla sesbírána na konferencích v letech 96 až 97 a byla (a je) neustále podrobována revizím. Vzory jsou podle dvou kritérií (souvislostí) rozděleny do skupin.

Prvním pohledem je metoda nebo rozsah výuky:

- diskuse,
- (skupinové) aktivity,
- (skupinová) procvičování,
- přednášky,
- techniky,
- celé kurzy,
- smíšené vzory.

Druhým pohledem je obsah výuky:

- abstrakce,
- přístupové metody,
- analýza,
- architektura,
- nalezení třídy,
- vize,
- návrh databází,
- návrh,
- údržba,
- modelování,
- objektové myšlení,

- spolupráce objektů,
- vzory,
- řešení problému,
- programování,
- zajištění kvality,
- problémy „reálného světa“,
- znouvupoužitelnost,
- obecné vzory.

V naší práci lze mnoho vzorů použít přímo. Výuka našeho studenta přechází od algoritmické matematiky (základní teoretická znalost algoritmů a datových struktur a věcí s nimi souvisejících a praktické dovednosti jejich implementace), přes softwarovou laboratoř (programovací techniky a technologie), paradigmatu programování (pochopení principů od procedurálního k objektovému), softwarové projekty (komplexní implementace složitějšího díla), databázové a informační systémy (teoretické znalosti a znalost postupů při použití databází) až po softwarové inženýrství se všemi jeho součástmi. Pro některé z těchto disciplín vybereme jeden ze vzorů, který nás oslovil především tím, že jsme v něm našli formalizaci námi používaných postupů (a někdy také pochopení toho, co jsme nedomysleli a z čeho vyplývaly nedostatky naší výuky).

What did you eat for breakfast?

Algoritmická matematika: Studenti mají potíže s pochopením toho, jaké problémy a jaká rizika jsou spojena s přístupem k soukromým datům (implementace datových struktur s využitím OOP). Vzor poskytuje fyzickou metaforu, na jejímž základě se dá budovat mnoho obdobných přirovnání, která studentům dají lepší představu toho, o co jde.

Programming int the Tiny, Small, Large (TSL).

Softwarová laboratoř: Při potřebě vysvětlit složitější techniku nebo technologii lze počítat s tím, že studenti způsob jejich správného využití nepochopí dříve, než sami využijí tyto koncepty při řešení vlastních (různě rozsáhlých) úloh. Vzor poskytuje návod, jak přistoupit k zadáním takových úloh a jejich realizaci na třech časově různě rozsáhlých úrovních – dnech, týdnech, měsících.

Design-Implement-Redesign-Reimplement (DIRR).

Paradigmata programování: Samozřejmou první dovedností studenta je zvládnutí principů procedurálního (nebo na naší katedře funkcionálního programování). Cílovou dovedností je pak efektivní využívání vlastností OOP. Vzor poskytuje návod, jak nejdříve navrhnout a implementovat známý problém známým a jednodušším způsobem a poté jej (se znalostí věci) znovu navrhnout a implementovat v prostředí jiného paradigmatu.

Expose The Process

Softwarové projekty: Úkolem studentů je v jednom školním roce analyzovat, navrhnout a implementovat řešení rozsáhlejší úkol. Vzor definuje postup, ve kterém se studentům ukáží obdobná hotová řešení a hned zpočátku se podhalí některé části postupu vedoucího k úspěšnému řešení. Postupně (tak jak se studenti nad postupem v diskuzích zamýšlejí) se odhalují a zpřesňují informace o způsobu řešení.

Závěrem

Na základě naší zkušenosti lze konstatovat, že vzory mohou pomoci v pochopení důvodů, proč se nám výuka někdy daří a jindy ne. Pedagogické vzory můžeme doporučit především těm, kteří

- nejsou studovanými učiteli,
- cítí, že není s výukou něco v pořádku (a neví přesně co),
- chtějí situaci zlepšit.

Literatura:

1. Alexander, Ch. A Pattern Language. New York. Oxford University Press 1977.
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley 1995.
3. Coplien, J. O. Software Patterns. AT&T Bell Laboratories.
4. Fowler, M. Patterns of Enterprise Application Architecture. Pearson Education 2002.
5. Coplien, J. O. A Development Process Generative Pattern Language. AT&T Bell Laboratories.
6. Berczuk, S., Appleton, B. Software Configuration Management Patterns. Pearson Education 2002.
7. Bergin, J. Fourteen Pedagogical Patterns.
8. J. Eckstein. Learn to Teach and Learn to Learn.
9. A. Fricke, M. Völter. A Pedagogical Pattern Language about teaching seminar effectively.
10. <http://www.pedagogicalpatterns.org/>
11. <http://sol.info.unlp.edu.ar/ppp/>
12. Kudělka, M., Sklenář, V. Design Patterns for Programs for Algorithm Demonstration. Sborník konference CBLIS '97 (Computer Based Learning In Science), Leicester 1997.
13. Kudělka, M., Sklenář, V. Deskové hry aneb návrhové vzory a výuka OOT. Sborník konference Objekty '97, Praha 1997.
14. Kudělka, M., Sklenář, V. Výuka, návrhové vzory a tvorba komponent aneb jak vytvořit deskovou hru. Sborník konference Objekty 2000, Praha 2000.
15. Kudělka, M., Sklenář, V. Implementace use case pomocí návrhového vzoru Controller. Sborník konference Tvorba Software, Ostrava 2002.
16. Kudělka, M., Mošová, V. Využití pedagogických vzorů ve výuce na VŠ. Sborník konference XX. mezinárodní kolokvium o řízení osvojovacího procesu, Vyškov 2002