

Studentská Vědecká Konference 2010

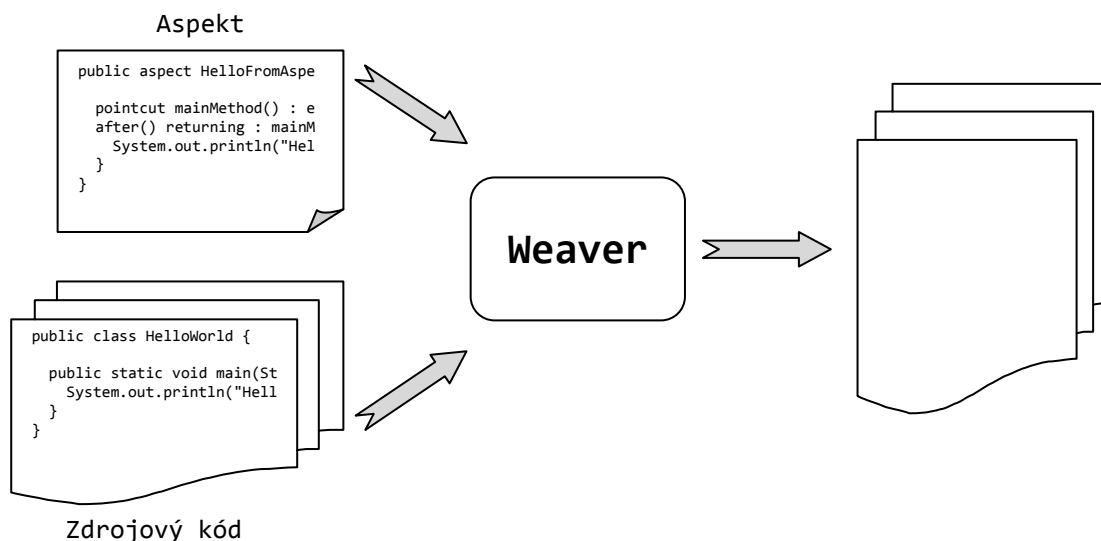
HLEDÁNÍ CEST K VYUŽITÍ CELÉHO POTENICÁLU ASPEKTOVĚ ORIENTOVANÉHO PROGRAMOVÁNÍ

Jan TICHAVA¹

1 ÚVOD

Při vývoji softwaru je možné používat nejrůznější programovací paradigmatu. K velice rozšířeným patří Objektivě orientované programování (OOP), které se v dnešní době značně využívá. OOP se používá již delší dobu, i komerční firmy navrhují a implementují software objektivě. Na rozdíl od Aspektově orientovaného programování (AOP), které je poměrně nové a zatím se příliš nepoužívá.

Pomocí AOP můžeme ovlivnit nebo i úplně změnit chování našeho algoritmu. Aspekt je nově přidaná funkcionality do programu bez zásahu do vlastního zdrojového kódu, někdy i bez nutnosti nové kompilace. Chceme-li aspekty využít, nemusíme se omezovat pouze na Objektivě orientované jazyky, ale můžeme je použít například i u funkcionálních jazyků. Na obr. 1 vidíme princip AOP Weaveru, který kombinuje zdrojový kód aplikace s aspekty.



Obr 1.: AOP Weaver a jeho funkce

Aspekty můžeme použít pro řadu různých případů. Ilustrativní příklad na použití aspektů jsem si vypůjčil z knihy Ramnivas (2003). Stavíte-li dům, zajímáte se primárně o hlavní části, jako jsou základy, velikosti místností, počet podlaží a místo stavby. Současně ale musíte vyřešit i elektroinstalaci. Nyní si představte, že stavíte most, který se skládá z pilířů, nosníků a podobně, ale také musíte opět vyřešit elektroinstalaci. Analogicky se můžeme dívat na software a požadavky na ověřování přístupu a definování různých stupňů uživatelského oprávnění.

¹ Ing. Jan Tichava, student doktorského studijního programu Inženýrská informatika, obor Informatika a výpočetní technika, e-mail: jtichava@kiv.zcu.cz

2 DALŠÍ ROZVOJ ASPEKTOVĚ ORIENTOVANÉHO PROGRAMOVÁNÍ

Během mého následujícího studia se budu věnovat dalšímu rozvoji odvětví týkající se AOP. Je řada oblastí, které jsou velmi zajímavé, mohly by mít slibnou budoucnost a přitom se příliš nevyvíjí.

Zejména se budu zabývat vlastním AOP a jeho nevyužitými možnostmi, jako je například jazykově neutrální AOP. To znamená, že aspekty by nebyly vázány na konkrétní programovací jazyk, ale jednou napsaný aspekt by bylo možné použít kdekoli. Tato oblast je velice zajímavá, ale není příliš aktivní.

Jazykově neutrálním AOP se již v minulosti zabývali Lafferty a Cahill (2003), kteří zapisovali aspekty pomocí XML, ale vlastní výkonný kód byl psaný v jazyce konkrétního jazyka. Dalším omezením je vazba pouze na programovací jazyky .NET. Bram (2005) ve své práci naopak uvažuje o implementaci takového weaveru přímo do kompilátoru jazyka C, který by umožňoval pro programovací jazyk C psát aspekty v libovolných jazycích jako je například Java nebo Fortran. Ve své budoucí práci chci navrhnout způsob, jak definovat aspekty zcela bez ohledu na programovací jazyk. Přínosem bude to, že programátor navrhne a implementuje aspekt pouze jednou a pak ho může používat kdekoli. Typické použití aspektů je logování a právě zde vidíme přínos jazykově nezávislého AOP.

Dále bych se rád, alespoň okrajově, zaměřil i na použití aspektů v komponentových systémech. Na první pohled se použití aspektů, při výběru vhodných komponent za běhu, jeví jako dobrý nápad a chtěl bych toto tvrzení ověřit hlubší analýzou. V neposlední řadě bych se rád věnoval zapojení aspektů i v rámci celého softwarového procesu, hlavně při návrhu budoucí architektury vyvíjeného systému.

3 ZÁVĚR

Ačkoli se AOP v dnešní době příliš nepoužívá, přesto existují různé nástroje, které jeho použití velice usnadňují. Dokonce jej implementují i některé frameworky.

Bude zajímavé sledovat, jakým směrem se budou technologie ubírat a jestli se v budoucnu začne AOP používat aktivněji nejen na akademické půdě, ale i v komerčních firmách. Osobně věřím, že se tak stane, protože tento způsob programování dokáže zpřehlednit a zjednodušit zdrojový kód, který je pak snáze znovu použitelný v jiných softwarových projektech a tím bude produkt levnější.

LITERATURA

- Ramnivas, L., 2003. *AspectJ in Action: Practical Aspect-Oriented Programming*, Manning Publications Co., Greenwich. ISBN: 1-930110-93-6.
- Lafferty D., Cahill V., 2003. *Language-Independent Aspect-Oriented Programming*, Distributed Systems Group, Department of Computer Science, Trinity College Dublin.
- Bram A., 2005. *Language-independent aspect weaving*, Software Engineering Lab, Department of Information Technology, Ghent University, Belgium.