

Model softvérového systému

Model softvérového systému predstavuje jeho opis, ktorý zachytáva vlastnosti systému z rôznych hľadísk. Model softvérového systému spravidla pozostáva z:

- logických prvkov (elementov, entít..), z ktorých systém pozostáva, alebo ktoré ovplyvňujú správanie systému;
- vzťahov medzi prvkami;
- diagramov – pohľadov na časti modelu;
- opisov.

Zložitý model celého softvérového systému je možné rozdeliť na menšie modely, ktoré zachytávajú jednu fázu vývoja, modul alebo typ modelovaných informácií. Takéto delenie robíme aj my v rámci projektu na PSI. Budeš model je tak výrazne prehľadnejší.

Častým problémom, s ktorým sa pri modeloch na PSI stretávame je aj zlé pochopenie rozdielu diagramu a modelu. Pokiaľ model zachytáva celú predstavu o vyvíjanom systéme z nejakého hľadiska, tak diagram je len jeden pohľad na model, ktorom chceme zobrazit' vybrané dôležité vlastnosti modelu. Povedzme, že kmeň stromu je model. V takom prípade ak rozrežeme tento kmeň na rovnej časti tak na reze vidíme pomocou letokruhov vek stromu. Čiže tento rez predstavuje diagram veku stromu. Keď prerežeme kmeň na mieste odkiaľ vyrastal konár máme iný pohľad na ten istý model (kmeň), ale tentokrát rez predstavuje diagram pripojenia konára ku kmeňu.

Podobne to funguje aj v modeloch softvérových systémov. Povedzme, ak máme model prípadov použitia systému (zjednodušene, všetkých funkcií, ktoré systém poskytuje aj s napojením na aktérov, ktorí tieto funkcie obsluhujú), tak môžeme spraviť viacero pohľadov na skupiny prípadov použitia. Napr. pri emailovom systéme môžeme v diagramoch zakresliť prípady použitia, ktoré používa používateľ na odoslanie emailu a administrátor na reaktiváciu používateľského konta. Všetky prípady použitia v oboch týchto diagramoch sú súčasťou jedného modelu. Navyše prípad použitia „Vyhľadanie používateľa“, ktorý je v modeli len raz sa zobrazí v oboch diagramoch (v prvom na vyhľadanie používateľa, ktorému posielal iný používateľ email a v druhom ho použije administrátor na nájdenie používateľa, ktorému má reaktivovať konto).

Ukážka

1. Nové elementy modelu (napr. triedy Objednávka, Položka objednávky a Tovar) zakresľujeme do diagramu opisujúceho objednávku priamo ich vytvorením v diagrame.
2. V prípade, že chceme v novom diagrame zachytiť iné vlastnosti softvéru, napr. dáta o sklade, tak existujúce entity vkladáme do diagramu pomocou Drag&Drop-u z okna "Project Browser"
3. Model softvérového systému na PSI je vytváraný v dvojiciach a trojiciach. Je preto potrebné zabezpečiť synchronizáciu medzi vašimi modelmi. K tomuto účelu sa používa takzvaná synchronizácia replík modelu.
4. Prvým krokom k tejto synchronizácii je vytvorenie hlavného modelu. Ten vytvoríte ak zvolíte možnosť Project->Data Management->Manage .EAP File->Make Design

Master a potvrdíte vytvorenie hlavného modelu. Tento hlavný model je možné štandardne používať na modelovanie softvéru.

5. Pre umožnenie práce ďalšiemu členovi tímu je potrebné vytvoriť repliku modelu. To dosiahnete pomocou možnosti Project->Data Management->Manage .EAP File->Create New Replica... . Následne zadajte umiestnenie vytvárajúcej repliky a jej meno. Túto repliku môže následne používať ďalší člen tímu na paralelnú prácu na projekte.
6. Prácu jednotlivých členov tímu je možné synchronizovať voči hlavnému modelu. Táto synchronizácia prebieha pomocou príkazu Project->Data Management->Manage .EAP File->Synchronize Replicas... vyvolaného z hlavného modelu, kde zvolíte repliku, s ktorou chcete hlavný model synchronizovať. Po synchronizácii, bude zjednotená replika aj hlavný model.
7. Ak počas synchronizácie nastane konflikt, je tento konflikt automaticky vyriešený, pričom zmenu automatického rozhodnutia o je možné riešiť aj na úrovni repliky aj hlavného modelu v Project->Data Management->Manage .EAP File->Resolve Replication Conflicts... . Viac informácií k konfliktom nájdete tu:
http://www.sparxsystems.com/enterprise_architect_user_guide/9.0/projects_and_teams/resolveconflicts.html
8. Enterprise architect nemá pokročilú logiku synchronizácie modelu a tak neumožňuje spájanie paralelnej práce na rovnakých elementoch. Čiže napríklad ak budete paralelne upravovať text úvodnej kapitoly, tak po synchronizácii ostane zachovaná len jedna verzia a nedôjde k zlúčeniu jednotlivých textov. Odporúčame preto často synchronizovať modely a medzi synchronizáciami nepracovať na rovnakých elementoch modelu.