# Realitní kancelář

*Oponentský posudek 1*

Tento dokument obsahuje oponentský posudek na vision document, business analýzu a analytický model projektu Realitní kancelář. Oponentura je sepsána podle aktuální verze dokmuentů platné k 23. 10. 2009 (revize číslo 69).

# Vize projektu

Dokument podává ucelenou představu o rozsahu a obsahu realizovaného SW. Obsahuje ovšem některé nedostatky, které jeho úroveň částečně snižují.

V oddíle Klíčové vlastnosti v kapitolce o prodeji/výkupu/pronájmu je uveden výčet možných typů nemovitostí, který svou konkrétností zasahuje nad rámec Vision documentu, jeho zásadnější vadou je ovšem nejasné rozčlenění a prolínání jednotlivých položek.

Také zbylá část popisu klíčových vlastností je nejasně strukturovaná a popis požadované funkčnosti je mírně zmatečný.

V oddíle Licencování chybí zmínka o databázovém serveru; zvláště pokud bude použit Oracle, jedná se o zásadní opomenutí. I v případě klientských počítačů pak chybí specifikace konkrétních podmínek (typu) licence.

Po formální stránce je drobnou vadou dokumentu množství překlepů a pravopisných chyb.

# Business analýza

Obecným problémem tohoto dokumentu je především takřka úplná absence popisu diagramu business process model a neúplná dokumentace k business domain modelu (prázdné kolonky v tabulkách atributů, zcela chybějící popisy tříd apod.) – a to často i v případech, kdy by detailnější textový popis mohl značně usnadnit pochopení celého konceptu.

## Business process model

Obecně o celém diagramu lze říci, že používá nejednotný styl názvů jednotlivých procesů (podstatným jménem, větnou konstrukcí), což je sice detail, ale jistě by bylo vhodné jej napravit. Následující odstavce již obsahují konkrétní výhrady k jednotlivým částem diagramu.

V diagramu jsou znázorněny dva procesy se shodným názvem „Obhlídka reality“ (Real estate inspection), což není příliš šťastné. Ovšem vzhledem k tomu, že oba reprezentují tutéž činnost (a i jejich popis v dokumentaci je totožný), nabízí se jednoduché řešení je sloučit v jeden proces (což jejich zapojení do běhu procesů to bez problémů umožňuje). U týchž procesů je ještě jedna chyba, a to jejich umístění v diagramu na hranici Makléře a Zákazníka (Real estate agent, Customer) – ta se týká ještě třetího procesu „Inspection“ (který mimochodem reprezentuje opět prakticky totéž, ovšem tentokrát ve zcela jiné fázi děje).

Sporným momentem je proces „Uložení do systému“ (Record is inserted into the system) – zachycuje tedy diagram stávající procesy v rámci realitní kanceláře, nebo je v něm zahrnuta i interakce s vyvíjeným SW?

Procesy dál pokračují s podmínkou „Buyer/Teanant“, ovšem pro variantu „Seller“ neexistuje žádná alternativa. Možná by stálo za úvahu oddělení procesu s prodávajícím do samostatného diagramu,

neboť ani předchozí průběh není s kupujícím/nájemcem společný. Přinejmenším by však mělo následovat větvení s větví „Seller“ vedoucí k ukončení transakce.

V diagramu se také nachází několik cyklů. Jejich použití je zde přinejmenší m sporné, neboť například po prohlídce nemovitosti (Inspection) může být zákazník buď spokojen, pak běh událostí pokračuje dál, nebo nespokojen, potom dojde k navrácení do fáze hledání vhodné reality. Žádná jiná možnost v diagramu není, a tak bude cyklus pokračovat do té doby, dokud zákazník s některou nemovitostí nebude spokojen. Diagram tak například nepočítá s možností, že zákazník ztratí zájem zcela; podobná situace nastává i u ostatních dvou cyklů.

## Business domain model

Patrně omylem bude vztah mezi Lokalitou a Adresou. Lokalita – alespoň podle popisu ve vision documentu, v dokumentaci k business modelu není podrobněji popsána – reprezentuje celou širší oblast, pro kterou jsou v databázi uloženy společné ratingy. Tato oblast tedy nemůže mít jednu adresu s uvedením ulice atp. Správněji by tedy měla Realita mít vztah 1:1 s Adresou (nikoliv s Lokalitou) a ta pak vztah n:1 s Lokalitou.

S adresou souvisí další problém, a sice atribut „type“. Ten podle popisu určuje, zda je adresa pro danou osobu adresou hlavní, vedlejší atp. ( „primary, secondary, ...“). To však není vlastnost adresy, ale vztahu mezi zákazníkem (Customer) a adresou – jednu adresu může sdílet více osob a pro každou z nich může atribut type nabývat různých hodnot. Stávající model sice očividně počítá s vícenásobnou evidencí shodných adres pro každou osobu zvlášť, ovšem je otázkou, zda je toto řešení vhodné (zejména například vzhledem ke vztahu s lokalitou, kde by k jedné lokalitě nesmyslně příslušelo větší množství zcela shodných adres – po opravení vztahu Lokalita – Adresa na správnou multiplicitu 1:n).

Drobná nejasnost je i u kategorií reality. Členění na hlavní kategorii („Category“) a podkategorii („Disposition“) by snad mohlo být v zásadě v pořádku (včetně navržených vztahů), ovšem rozdělení jednotlivých atributů ke kategoriím je sporné. Pokud třída Disposition určuje především dispozici objektu, jak tato souvisí například s energetickou kategorií (což je jeden ze dvou atributů třídy Category). Bylo by tedy vhodné buď logičtěji rozvrhnout jednotlivá kritéria mezi obě úrovně kagetorizace, nebo vytvořit místo vztahu Category – Disposition přímo vztah mezi Disposition a RealEstate.

Sporná je i dvojice tříd „Contract“ a „Transaction“, které spolu úzce souvisejí. Zatímco Kontrakt uzavírá zákazník s manažerem, transakci provádí zákazník s realitou. Obojí je přitom téměř neoddělitelné, neboť zákazník patrně nemůže manipulovat s realitou, aniž by to bylo podloženo smlouvou s realitní kanceláří. Možná by proto stálo za úvahu nahrazení obou tříd jedinou entitou, která by měla vztah jak k zákazníkovi a manažerovi, tak i k příslušné realitě (a reprezentovala kontrakt i související transakci).

Co se týče stavových diagramů, po formální stránce je podivné použití nadpisů ContractStatus a RealEstateStatus, jako by se jednalo o samostatné entity. V případě prvního ContractStatus není zcela jasné, proč od stavu Paid vede přechod ke koncovému bodu, zatímco od stavů Disagreed a Canceled nikoliv. V případě RealEstateStatus lze pouze uvažovat o přidání možnosti zrušení rezervace (což ovšem záleží na tom, zda taková možnost existuje) a možná také o posunutí rozdělení na dvě větve již před rezervaci (rezervováno pro prodej/rezervováno pro pronájem), to ovšem závisí na konkrétních požadavcích na systém.

# Analytický model

Dokument s analytickým modelem obsahuje zbytečně znovu také business modely, ovšem umístění celé dokumentace do jednoho souboru může mít jistě i své výhody.

V seznamu případů užití u nefunkčního požadavku „Použití pro kancelářské počítače RK“ není blíže specifikováno, co se kancelářským počítačem myslí. Přinejmenším zmínka o nutnosti podpory JRE by zde jistě být měla.

Hlavní výtkou k Use case diagramu je přílišná obecnost. Diagram obsahuje pouze malé množství velmi obecných případů užití (k některým z nich například sotva půjde vytvořit přiměřeně jednoduchý scénář). Příkladem může být ModifyItem, který zahrnuje všechny možné akce týkající se správy nabízených nemovitostí – což je jistě příliš široký okruh činností (avšak problém se zdaleka netýká pouze tohoto případu).

Dále není zcela jasná pozice Modify, který podle popisu představuje změnu atributů záznamů. Nejedná se už tedy patrně o interakci uživatele se systémem, ale spíše o implementační záležitost, jejíž zmínka v Use case diagramu nemá smysl. Možná by však více mohl objasnit podrobnější popis, každopádně by však průběh úprav probíhal zcela jinak pro různé druhy úprav, takže i zde by bylo na místě detailnější rozlišení.

V Use case diagramu se také opět objevují drobné formální nedostatky. Drobný nesoulad je v pojmenování většiny případů užití slovesnou vazbou, naproti tomu názvem činnosti (podle zvyklostí) je označeno pouze „Checking“. Dále je zde duplicitně uvedený případ užití ModifyOrder.

Posledním problémem jsou určité nesrovnalosti mezi jednotlivými diagramy. Zatímco v Business process modelu agent připravuje kontrakt a manažer ho schvaluje (a Business domain model vztah agenta a kontraktu neuvádí vůbec), podle Use case diagramu může agent i manažer kontrakt vytvořit – a nic dalšího se zde neuvádí. V případě Use case diagramu to zřejmě souvisí také s již zmíněnou přílišnou obecností. Podobně zatímco v Business domain modelu najdeme možnost rezervace nemovitosti, případy užití týkající se tohoto úkonu v Use case diagramu chybí.

Podle Vision documentu jsou v podniku tři role: hlavní manažer, manažer a makléř. Organizace poboček, spadající do kompetencí hlavního manažera, je v tomto dokumentu sice uvedena mezi klíčovými vlastnostmi vyvíjeného software, v další dokumentaci je ale zcela opomíjena (stejně jako role hlavního makléře). Především je ale ve Vision documentu uvedeno, že každý nadřízený má zároveň všechna práva podřízeného. V Use case diagramu by tedy BrandManager měl nejspíš dědit od RealEstateAgenta, neboť má i všechna jeho práva.