

Modelovanie spracovania elektronického podania ÚPVS sekvenčnými UML diagramami*

Viktor Uhlár

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xuhlar@stuba.sk

6. november 2021

Abstrakt

Ústredný portál verejnej správy (ÚPVS) je centrálnym miestom na podávanie a spracovanie elektronických podaní. Pre agendové informačné systémy poskytuje možnosť integrácie tak, aby bolo možné zasielať elektronické podania priamo z vlastných informačných systémov (teda bez nutosti prihlasovania sa do elektronických schránok pomocou eID - elektronického občianskeho preukazu).

Keďže postupnosť spracovania elektronického podania závisí od jeho účelu, musí byť pred fázou samotnej integrácie vypracovaný model “workflow“. ÚPVS pre tento účel vyžaduje tzv. Dohodu o integračnom zámere, kde je spracovanie modelované vo forme sekvenčného UML diagramu. V tejto práci sa teda budem venovať teórii modelovania sekvenčných UML modelov s konkrétnym príkladom pre spracovanie elektronického rozhodnutia “fiktívneho“ OVM (orgánu verejnej moci SR).

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Vladimír Mlynarovič

1 Úvod

Ústredný portál verejnej správy (ÚPVS) je centrálnym miestom na podávanie a spracovanie elektronických podaní. Pre agendové informačné systémy poskytuje možnosť integrácie tak, aby bolo možné pristupovať k elektronickej schránke priamo z vlastných informačných systémov - teda bez nutosti prihlasovania sa eID - elektronického občianskeho preukazu.

Keďže spôsob pripojenia môže byť rôzny, správca ÚPVS (Národná agentúra pre sieťové a elektronické služby - NASES) vyžaduje podrobný integračný zámer, ktorého súčasťou je aj modelovanie integrácie vo forme sekvenčného UML diagramu.

V tejto práci sa teda budem venovať:

- Rámcovému popisu UML a typov diagramov
- Spôsobu tvorby sekvenčných UML diagramov
- Jednoduchému nástroju pre modelovanie UML diagramov
- Modelovej integračnej situácie pripojenia na ÚPVS a popisu tvorby súvisiaceho sekvenčného UML diagramu

Informácie pre vytvorenie tejto práce som čerpal:

- Z integračnej dokumentácie spoločnosti NASES
- Z internetových zdrojov ku UML modelovaniu
- Z praktickej skúsenosti pri spolupráci v rámci integračného ÚPVS projektu

Zoznam konkrétnej použitej literatúry je uvedený v prílohe Literatúra. Práca môže byť užitočná nielen pre analytikov, ktorí zodpovedajú za modelovanie integračných "ÚPVS" projektov, ale aj pre všeobecnejšie pochopenie zmyslu sekvenčných diagramov (ako integrálnej súčasti software development cyklov).

2 Popis UML a rozdelenie typov diagramov

UML znamená Unified Modeling Language, teda grafický jazyk na vizualizáciu, špecifikáciu, navrhovanie a dokumentáciu programových systémov.

UML ponúka štandardný spôsob zápisu tak návrhov systémov vrátane konceptuálnych prvkov ako sú business procesy a systémové funkcie, tak konkrétnych prvkov ako sú príkazy programovacieho jazyka, databázové schémy a znovupoužiteľné programové komponenty. [3]

Pomocou UML je možné modelovať niekoľko rozličných diagramov. Ich charakteristika je zrejmá z nasledovného obrázku [1]:

UML is a coherent system of languages rather than a single language. Each language has its particular focus.				
Structure	Class Diagram	static structure (generic/snapshot)		
	Composite Structure Diagram	logical system structure		
	Component Diagram	physical system structure		
	Deployment Diagram	computing infrastructure / deployment		
	Package Diagram	containment hierarchy		
Behavior	Use Case Diagram	abstract functionality		
	Activity Diagram	controlflow and dataflow		
	Interaction	Sequence Diagram	interactions by message exchange	message exchange over time
		Communication Diagram		structure of interacting elements
		Timing Diagram		coordinated state change over time
		Interaction Overview Diagram		flows of interactions
	State Machine Diagram		event-triggered state change	

Diagramy je teda možné rozdeliť do 3 základných skupín [3]:

1. Štruktúrne diagramy, kam patria:

- diagram tried (class diagram)
- diagram komponentov (component diagram)
- diagram zloženej štruktúry (composite structure diagram)
- diagram nasadenia (deployment diagram)
- diagram balíčkov (package diagram)
- diagram objektov (object diagram), nazýva sa aj diagram inštancií

2. Diagramy správania, kam patria:

- diagram aktivít (activity diagram)
- diagram prípadov použitia (use case diagram)
- stavový diagram (state machine diagram)

3. Diagramy interakcie, kam patria:

- sekvenčný diagram (sequence diagram)
- diagram komunikácie (communication diagram) - predtým diagram spolupráce (collaboration diagram)
- diagram prehľadu interakcií (interaction overview diagram)
- diagram časovania (timing diagram)

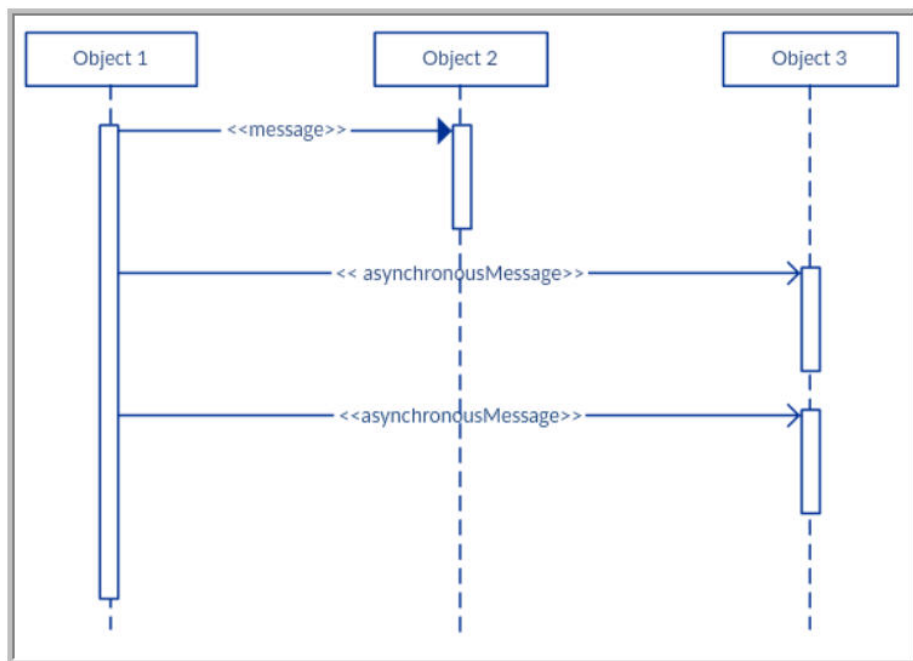
3 Sekvenčný diagram a jeho modelovanie

Sekvenčný diagram modeluje interakciu medzi objektami počas konkrétneho “use case“, teda prípadu použitia. Graficky ilustruje, ako rozličné časti systému navzájom spolupracujú tak, aby zrealizovali požadovanú funkciu. Zjednodušene povedané: sekvenčný diagram ukazuje postup činností rozličných častí systému [2].

Pri zobrazení sa využívanie niekoľko značiek, pričom k najpoužívanejším patria:

- Lifeline – životná línia
- Actor – účinkujúci
- Activity – aktivita
- State – stav
- Message arrow – tok informácie, ktorý môže byť:
 - Synchronous (synchronná)
 - Asynchronous (nesynchronná)

Na nasledovnom obrázku je ilustrácia jednoduchého sekvenčného diagramu, využívajúceho vyššie uvedené značky [2]:



4 Nástroj na modelovanie formou sekvenčného UML diagramu

..... nástroj bude doplnený...

5 Integračný scenár a jeho modelovanie

..... test a diagramy budú doplnené...

6 Zhodnotenie

..... zhodnotenie bude doplnené...

zdôrazniť kurzívou.

1

Veľmi dôležitá poznámka. Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

¹Niekedy môžete potrebovať aj poznámku pod čiarou.

Literatúra

- [1] Harald Storrle, Alexander Knapp. Unified Modeling Language 2.0.
University of Innsbruck, University of Munich. 2006. (M. Word) .
- [2] Creately.com. Sequence diagram tutorial.
<https://creately.com/blog/diagrams/sequence-diagram-tutorial/>.
- [3] Wikipedia. Unified modeling language.
https://sk.wikipedia.org/wiki/Unified_Modeling_Language/.