

Modelovanie bankomatového softvéru v UML*

Beáta Belková

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xbelkovab@stuba.sk

6. oktober 2021

Abstrakt

Článok sa zaoberá modelom bankomatového softvéru. Na začiatku stručne definuje čo je Unified Modeling Language, ktorý je na modelovanie použitý. Následne pomocou vybraných diagramov demonštruje model softvéru pre bankomat. Použité diagramy sú sekvenčný diagram, diagram tried a diagram prípadov použitia.

,

1 Úvod

Bankomaty prešli za posledných 60 rokov veľkými zmenami. Z jednoduchých zariadení, ktoré vymieňali jednorazové poukážky za obálky s bankovkami sa stali zariadenia, ktoré požívane denne. Softvér bankomatu sa počas rokov stával čoraz zložitejším, a preto si jeho tvorba vyžaduje sofistikovaný model.

Základné informácie o grafickom jazyku, ktorý je použitý na modelovanie,, ako aj popis jeho diagramov sú v časti 2. Príklady vybraných UML diagramov pre bankomatový softvér a ich vysvetlenie článok ponúka v časti 3. Záverečné poznatky a zhrnutie celého článku prináša časť 4.

2 Unified Modeling Language (UML)

(definícia UML je sformulovaná na základe článku [2])

UML je grafický jazyk, ktorý je štandardom v oblasti analýzy a návrhu pri vývoji softvéru. Vo fáze analýzy pomáha odhadnúť cenu systému a taktiež uľahčuje komunikáciu s klientom, keďže diagramy sú jednoduché na pochopenie. Vďaka UML je jednoduchšie reagovať na zmeny v zadaní klienta. Vo fáze návrhu pomáha riešiť otázku ako bude softvér naprogramovaný.

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

Typy diagramov

UML v súčasnosti pozostáva zo 14 diagramov. Tie sú rozdelené do dvoch základných kategórií: diagramy štruktúry (Structure diagrams) a diagramy správania (Behaviour Diagram).

Diagramy štruktúry	Diagramy správania
Diagram tried	Diagram aktivít
Schéma komponentov	Diagram prípadov použitia
Objektový diagram	Stavový diagram
Zložený štruktúrny diagram	Komunikačný diagram
Schéma komponentov	Sekvenčný diagram
Schéma nasadenia	Graf prehľadu interakcií
Profilový diagram	Synchronizačný diagram

V článku bude pre model bankomatového softvéru použitý sekvenčný diagram (zobrazuje interakcie medzi zákazníkom, bankomatom a bankou), diagram tried (obsahuje všetky triedy, ktoré bude softvér používať a vzťahy medzi nimi), a diagram prípadov použitia.

3 Model bankomatového softvér

3.1 Diagram prípadov použitia

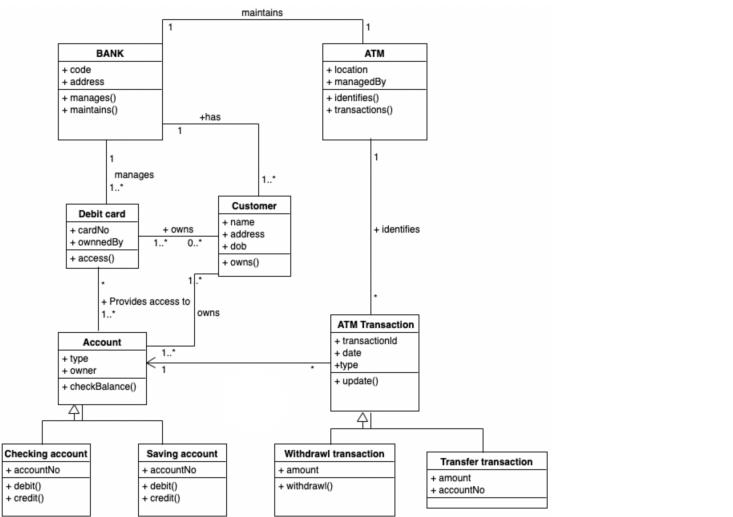
Diagram prípadov použitia je typ diagramu, ktorý zobrazuje vzťah medzi aktérmami (v tomto prípade bankou a klientom) a systémom. Taktiež znázorňuje funkcie, ktoré by mal systému vykonávať. Na obrázku 1 sú prípady použitia znázornené pomocou elipsy, v ktorej sa nachádza popis práce softvéru. Čiary spájajúce aktérov a prípady použitia predstavujú interakcie medzi nimi.



Obr. 1: Diagram prípadov použitia [3]

3.2 Diagram tried

Diagram tried je najpoužívanejší UML diagram. Zobrazuje triedy, ktoré bude softvér obsahovať a vzťahy medzi nimi. Diagram musí byť kompletný aby sa podľa neho dal napsať kód a preto musia triedy obsahovať všetky atribúty a metódy. Pod pojmom atribúty si môžeme predstaviť dátá, s ktorými bude trieda pracovať. Metódy sú operácie, ktoré bude daná trieda vykonávať.



Obr. 2: Diagram tried [3]

Na obrázku 2 je ako prvá zobrazená trieda "BANK". Atribúty, ktoré bude obsahovať sú kód na jej identifikáciu a adresa. Metódy, ktoré banka používa na spoluprácu s bankomatom sú spravovanie a údržba. Znamienko plus pred názvami atribútov a metód znamená, že sú verejné.

Ďalšou triedou v diagrame je ÄTM „, čiže bankomat. Metódy tejto triedy deklarujú transakcie, ktoré vykonáva.

Ako uvádzajú článok [1], každá banka potrebuje klientov, a preto diagram obsahuje triedu "CUSTOMER". Údaje, ktoré trieda potrebuje sú meno klienta a adresa. Klient nevykonáva žiadne operácie iba vlastní produkty banky.

Klient môže vlastniť účet a debetnú kartu. Preto bude diagram obsahovať triedy Äccount“ a "Debit card“. Účet má v atribútoch zapísaného vlastníka a typ účtu. Táto trieda môže vykonať kontrolu stavu na účte. Trieda pre debetnú kartu má v atribútoch zapísané číslo karty a vlastníka karty. Metóda karty je "prístup", pretože umožňuje prístup k bankomatu a tým aj k účtu vlastníka.

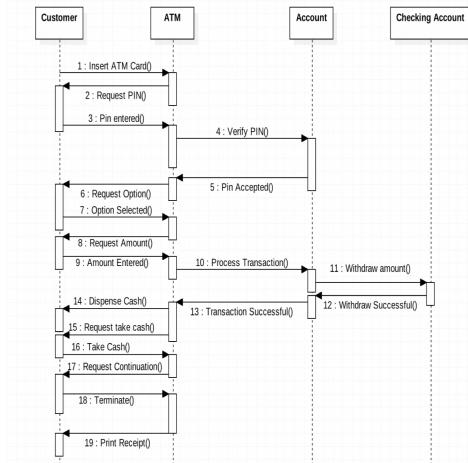
Ďalšie triedy, ktoré musí diagram obsahovať popisujú transakcie vykonávané počas používania bankomatu. Trieda ÄTMtransaction“ má tri atribúty – ID transakcie, dátum a typ transakcie. Jej jedinou metódou je aktualizácia.

V prípade diagramu 2 sa predpokladá, že zákazník môže iba vybrať peniaze alebo peniaze previesť na iný účet. Preto obsahuje triedy "WithdrawalTransaction“ a "TransferTransaction“. Tieto dve triedy sú podriedami „ATMtransaction“ a preto môžu zdediť jej metódy. Transakcia výberu potrebuje údaje o sume, ktorú chce klient vybrať. Potom metóda výberu uskutoční vydanie hotovosti klientovi a zdedená metóda aktualizácie aktualizuje stav na účte. Transakcia prevedenia

peňazí na druhý účet má iba dva atribúty. Sumu, ktorú chce klient poslať a číslo účtu, na ktorý má byť suma pripísaná. Trieda po schválení transakcie využije zdedenú funkciu aktualizácia, ktorá aktualizuje stav na účte.

3.3 Sekvenčný diagram

V tejto podkapitole článok bližšie popíše obrátok 3.



Obr. 3: Sekvenčný diagram [3]

4 Záver

Literatúra

- [1] Constantine Nalimov. Class diagram for an atm system: step-by-step guide. <https://www.gleek.io/blog/atm-system-class.html>.
- [2] David Čápka. 1. diel - Úvod do uml. <https://www.itnetwork.sk/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>.
- [3] DAV University. Sample of uml diagrams for atm system. <https://www.davuniversity.org/images/files/study-material/UML-ATM.pdf>.