VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

AOS: NÁSTROJ PRO ANALÝZU GRAFU

AOS: TOOL FOR GRAPH ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **MARTIN SIEBERT**
AUTHOR

VEDOUCÍ PRÁCE **Mgr. ROMAN TRCHALÍK, Ph.D.**
SUPERVISOR

BRNO 2017

Zadání bakalářské práce/20164/2016/xsiebe04

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Siebert Martin**
Obor: Informační technologie
Téma: **AOS: Nastroj pro analýzu grafu**
AOS: Tool for Graph Analysis
Kategorie: Informační systémy

Pokyny:

1. Seznamte s obchodní platformou Ninja Trader.
2. Analyzujte požadavky a vypracujte detailní návrh nástroje pro analýzu grafu (časových řad). Povinnými požadavky na vyvíjený systém je nezávislost komponent ve smyslu SW architektury MVC nebo MVP a možnost integrace nových analytických modulů. Analytický modul provádí specifickou analýzu nad grafem (časovou řadou).
3. Takto navržený nástroj implementujte podle pokynů vedoucího. Součástí nástroje se předpokládá i implementace aspoň 5 analytických modulů.
4. Na libovolné časové řádě proveďte testování implementovaného nástroje a své výsledky prezentujte vhodnou formou.

Literatura:

- Ninja Script Reference [cit. 2016-10-13] <https://ninjatrader.com/support/helpGuides/nt7/?alphabetical_reference.htm>
- ESPOSITO, Antonio. *Learning .NET High-performance Programming*. Packt Publishing, 2015. ISBN 9781785288463.
- R.M.C. Pinto. Strategic methods for automated trading in Fore. JCM Silva. 2012. ISBN: 978-1-4673-5117-1
- J. J. Murphy. Study Guide to Technical Analysis of the Financial Markets, 1999. ISBN: 978-0735200654
- R. D. Edwards. Technical Analysis of Stock Trends, Tenth Edition. 2012. ISBN: 978-1439898185

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Trchalík Roman, Mgr., Ph.D.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Cieľom tejto bakalárskej práce je navrhnúť a implementovať nástroj slúžiaci k analýze grafu tržných dát. Jednou z hlavných požiadaviek návrhu je jednoduchá modifikovateľnosť a udržiavateľnosť zdrojového kódu. Tento cieľ je dosiahnutý pomocou achitektúry MVC a vytváraním na sebe nezávislých komponentov systému. Výsledná aplikácia zaisťuje získanie dát z obchodnej platformy, vykonáva výpočet podľa požiadaviek a zobrazuje informácie v grafe. Výpočet je realizovaný pre každý modul zvlášť s tým, že jednotlivé moduly môžu tvoriť viacero postupností rôznej dĺžky. V rámci tejto práce je implemetovaných päť analytických modulov a vytvorené dve vzorové prípadové štúdie.

Abstract

This Bachelor's thesis is focused mainly on design and implementation of a tool, that analyses price graphs. One of the main requirements of the design is modifiability and maintainability of source code. This goal is achieved with the MVC architecture and by creating independent system components. The application takes care of collecting the data from the trading platform, performs the calculation based on the requirements and displays the information in the graph. For each module, the calculation is being done independently, whereas the modules can form multiple sequences of different lengths. In this Bachelor's thesis, there are five analytical modules implemented and two sample case studies created.

Kľúčové slová

Burza, obchodovanie, automatický obchodný systém, technická analýza, finančné trhy, analýza grafu, grafové formácie, indikátory, NinjaTrader, C#, .NET

Keywords

Stock exchange, trading, automated trading system, technical analysis, financial markets, graph analysis, chart patterns, indicators, NinjaTrader, C#, .NET

Citácia

SIEBERT, Martin. *AOS: Nástroj pro analýzu grafu.* Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Mgr. Roman Trchalík, Ph.D.

AOS: Nástroj pro analýzu grafu

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Mgr. Romana Trchalíka, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Martin Siebert
14. mája 2017

Poďakovanie

Ďakujem vedúcemu tejto bakalárskej práce, Mgr. Romanovi Trchalíkovi, Ph.D., za trpezlivosť, odbornú pomoc a cenné rady, ktoré mi vždy ochotne poskytol.

Obsah

Kapitola 1

Úvod

Obchodovanie a investovanie je základom ľudskej činnosti už veľmi dlhú dobu. Kedysi sa jednalo o výmenu tovaru (tzv. barter) a postupom času sa stali univerzálnym platidlom peniaze a rôzne finančné aktíva. Finančné trhy sú pre veľký okruh ľudí stále veľkou neznámou, avšak banky a rôzne investičné spoločnosti sa na nich stretávajú a súperia každodenne. Väčšina obchodov na týchto trhoch je v dnešnej dobe vykonávaná na základe rozhodnutí automatických obchodných systémov.

Obchodovanie na burze môže byť ziskové aj stratové. Obchodník, ako jednotlivec, si musí jasne stanoviť svoje ciele (čo očakáva), vymyslieť obchodný plán a dôkladne ho otestovať. Ďalším krokom k úspechu je vôľa neustále sa vzdelávať a skúšať nové a nové postupy a stratégie. V neposlednej rade je dôležité, aby obchodník zvolil vhodné nástroje, ktoré mu pomôžu splniť jeho plán.

Hlavným cieľom tejto práce je analýza požiadaviek, následný návrh, implementácia a testovanie nástroja pre analýzu grafu, ktorý by obchodníkom, matematikom, štatistikom a programátorom pomáhal vykonávať rozsiahle technické analýzy rôznych trhov.

Začiatok tejto práce tvorí kapitola, v ktorej je poskytnutý celkový pohľad na systém obchodovania. Sú v nej vysvetlené pojmy ako burza, kapitálový trh, nachádza sa tu aj zoznam účastníkov trhu, rozdelenie burzy a predstavenie základných obchodovaných aktív. Ďalej obsahuje definíciu technickej analýzy, ako jedného z troch hlavných mechanizmov predvídania budúceho vývoja cien na trhu a v neposlednej rade obsahuje pohľad na automatické obchodné systémy a obchodnú platformu NinjaTrader.

Ďalšia kapitola je zameraná na špecifikáciu a analýzu požiadaviek na vyvíjaný nástroj. Obsahuje zoznam požiadaviek na funkčnosť aplikácie a ich znázornenie pomocou diagramu prípadov použitia a požiadavky na uchovávanie dát v systéme.

Tretia kapitola tejto práce sa zameriava na samotný návrh systému. Je v nej znázornená architektúra systému a jeho rozdelenie na isté podsystémy, ktorých činnosť je podrobne popísaná. Dôležitý je aj pohľad na uchovávanie dát v systéme. Záver tejto kapitoly tvorí návrh užívateľského rozhrania vyvíjanej aplikácie.

Štvrtá kapitola popisuje implementáciu jednotlivých súčastí systému. Sú v nej bližšie priblížené dve stratégie umiestnené v obchodnej platforme NinjaTrader, tri najdôležitejšie moduly aplikácie a podrobný popis piatich analytických modulov, ktorých implementácia je súčasťou vyvíjaného nástroja.

V poslednej kapitole je zobrazený pohľad na dve prípadové štúdie, ktorých cieľom je overenie hypotéz (potencionálnych stratégii) pomocou implementovaných analytických modulov.

Kapitola 2

Zoznámenie s obchodovaním na burze

Burzové obchodovanie existuje už niekoľko storočí, ale až v posledných desiatkach rokov je vďaka internetu a znižujúcim sa požiadavkam na otvorenie obchodného účtu dostupné takmer pre všetkých. Taktiež aj požiadavky na počiatočný kapitál sa znižujú a na niektorých trhoch je možné začať obchodovať aj s niekoľkými desiatkami eur. Princíp obchodovania na burze však stále zostáva rovnaký a spočíva v predvídaní pohybu cien určitých aktív.

Burza je inštitúcia, ktorá organizuje trh s investičnými nástrojmi a je jednou zo základných súčastí kapitálového trhu. Prostredníctvom burzy je možné nakupovať a predávať rôzne investičné nástroje. Na burze sa stretávajú eminenti a investori. Cieľom eminentov je získať finančné prostriedky pre svoje podnikanie a investori majú možnosť svoje voľné finančné prostriedky zhodnotiť. Burza má teda podobu obojstrannej aukcie, kde o cene obchodovaného inštrumentu rozhoduje stav ponuky a dopytu [10].

Táto kapitola obsahuje charakteristiku kapitálového trhu, ktorá predstavuje základ potrebný pre pochopenie konceptu obchodovania. Ďalej sa zameriava na technickú analýzu, ako na jeden z hlavných mechanizmov predvídania vývoju cien aktív. V tretej časti je vysvetlený pojem automatický obchodný systém a poskytnutý pohľad na problémy, ktoré v dnešnej dobe nastávajú pri snahe o vytvorenie ziskových automatických obchodných systémov. V poslednej časti tejto kapitoly je popis obchodnej platformy NinjaTrader a jej častí, ktoré budú využité pri tvorbe nástroju pre analýzu grafu.

2.1 Kapitálový trh

Kapitálový trh je jedným z druhov finančných trhov a umožňuje výmenu tovaru. Obchodujú sa tu teda peniaze a najrôznejšie aktíva vyjadrujúce peniaze. Na trhu kapitálu má všetko svoju presne danú cenu a je veľmi rýchlo zameniteľné späť za peniaze. Z hľadiska obchodovania a investovania je dôležité si ujasniť niekoľko vecí. Kapitál má jeden jediný základný cieľ a tým je jeho rast. Na trhu nie je nikto, kto by zámerne nechal zarobiť niekoho iného na svoj úkor. Zisk vyplývajúci z nejakého obchodu je úmerný strate niekoho iného. To v praxi znamená, že sa obchodník vždy obohatí na úkor niekoho iného a nie je možné jednoducho zarobiť na ozdravení ekonomiky alebo rastom nejakej spoločnosti.

Výhodou kapitálového trhu je jeho likvidita. Pri obchodovaní na štandardných trhoch nie je problém behom pár sekúnd predať svoje aktívum a mať svoje finančné prostriedky k dispozícii pre iné využitie. Ďalšia výhoda je to, že na kapitálovom trhu nie je problém ne-

platičov. Vďaka elektronickému systému a účtom u brokera sa nemôže stať, že si protistrana kúpi od obchodníka nejaké aktíva, na ktoré nemá dostatok peňazí [9].

2.1.1 Účastníci kapitálového trhu

Na kapitálovom trhu sa pohybuje rada „hráčov“ a ich predstavenie je jedným z hlavných pilierov úspešného obchodovania na trhoch. Účastníkov kapitálového trhu možno podľa [9] rozdeliť do troch kategórii: veľkí hráči, malí hráči a brokeri.

Veľkí hráči sú nadnárodné spoločnosti, ktoré obchodujú na trhu miliardy každý deň. Sú to predovšetkým veľké investičné spoločnosti, ako napr. JP Morgan, Deutsche Bank a ďalší. V súvislosti s nimi je možné sa stretnúť s pojmom „Market Maker“ – tvorca trhu. Cieľ týchto inštitúcii je rovnaký ako každého iného účastníka trhu – zisk. Obchody veľkých hráčov obvykle znamenajú určenie trendu trhu.

Malí hráči sú samostatní a začínajúci obchodníci a menšie obchodné spoločnosti. Táto skupina sa vyznačuje tým, že nie je schopná pohnúť trhom, určiť jeho smer a vývoj podľa svojej vôle. Malí hráči musia zisťovať, ako sa trh pohybuje a vymýšľať svoju stratégiu podľa tohto pohybu. Veľmi častá chyba začínajúcich obchodníkov je chcieť alebo očakávať, že trh pôjde tým smerom, ktorým chcú aby išiel, prípadne sa dokonca snažiť trh „pretlačiť“.

Poslednou kategóriou účastníkov kapitálového trhu sú brokeri. Broker je spojovací článok medzi obchodníkmi a burzou, kde je možné nakúpiť aktívum. Jednoducho povedané, broker za určitý poplatok umožní obchodníkom pohodlne obchodovať na burze. Broker však v súvislosti s týmito poplatkami spôsobuje istú nerovnováhu na kapitálovom trhu. V situácii, kedy je kapitálový trh v určitom čase uzatvorený a peniaze do neho neprichádzajú ani z neho neodchádzajú, vzájomnými obchodmi niektorí účastníci získajú, niektorí stratia, peniaze však nevznikajú, ani úplne nemiznú. Keď to tejto schémy vstúpi broker, táto rovnováha bude narušená, pretože z trhu odčerpáva peniaze, ktoré sa musia odniekiaľ objaviť [9].

2.1.2 Rozdelenie burzy

V literatúre [9] sú burzy delené podľa zamerania na peňažné – akciové, komoditné a menové. Taktiež sú členené aj územne, pričom každý štát má väčšinou svoju burzu. Drobní investori a obchodníci však nemajú záujem obchodovať na týchto malých burzách a vystačia si s veľkými najznámejšími burzami v zahraničí.

Peňažné – akciové burzy sú zrejme najznámejší typ búrz. Je na nich možné nakupovať a predávať akcie a iné finančné deriváty. Z pohľadu spoločností je to miesto, kde je možné upísať svoje akcie a získať za ne kapitál. Prvé burzy tohto typu vznikali už v dvanástom storočí. Dôvodom existencie týchto búrz je obchod so spoločnosťami na nich vedenými, predaj a nákup dlhopisov, operácie s finančnými derivátmi typu ETF [12] atď. Tieto burzy sú väčšinou viazané na hlavné mestá štátov, keďže v nich prebieha aj úpis domácich spoločností. Veľa spoločností sa však necháva vypisovať aj na iných než domácich burzách, napríklad na rozvinutejších burzách v Londýne alebo v USA.

Komoditné burzy sú burzy, na ktorých sa obchoduje s komoditami. Komodity je najjednoduchšie si predstaviť ako poľnohospodárske produkty, indexy [13], energie, kovy a ďalšie finančné deriváty. História komoditných búrz sa datuje do dôb, kedy ľudia po prvý krát potrebovali vymieňať tovar. Tento význam majú tieto burzy dodnes, pretože sa na nich obchodujú aktíva s reálnym podkladom, na rozdiel od peňažných búrz. Komodity sa obchodujú v tzv. kontraktoch (lotoch). Kontrakt má špecifikované množstvo základnej komodity, pričom je to vždy určitý násobok ceny, za ktorú prebiehajú obchody. Kontrakty sa vypisujú

na konkrétny dátum, kedy dôjde k ich vypršaniu a fyzickému dodaniu komodity vlastníkovi kontraktnej listiny.

Menová burza (Forex – Foreign exchange) je trh s menami, ktoré na svete používajú jednotlivé štáty ako svoje platidlá. Účel tohto trhu je zmena jednej meny za inú. Forexový trh je jedným z najväčších a najrozšírenejších trhov a objem zobchodovaných prostriedkov za deň sa tu pohybuje rádovo v triliónoch dolárov. Forex nemá jednu centrálnu burzu, ale obchodovanie prebieha na subtrhoch v rámci veľkých obchodných domov. Subtrhy sú medzi sebou prepojené pomocou internetu a ponuka sa stretáva s dopytom. Obchodovanie na Forexe prebieha celý deň, keďže neexistuje jedna centrálna burza, a aktívum je v podstate jedno pre celý svet. Počas denného cyklu majú rôzne mestá aktívnu časť dňa v rôzny čas a preto rastie a klesá aktivita na tomto trhu. Medzi najaktívnejšie centrá patrí Londýn, New York, Tokio a Hong Kong. Na tomto trhu sa zúčastňujú jednotlivci, ktorí si chcú zameniť peniaze, väčšie obchodné spoločnosti aj veľké banky, ktoré tu chcú zarobiť. Pri vývoji nástroja pre analýzu grafu bude využívaný ako referenčný práve Forexový trh, kvôli jeho dynamickému charakteru a zaujímavosti z hľadiska technickej analýzy a algoritmického obchodovania [8].

2.1.3 Základné aktíva

Medzi najbežnejšie a najviac obchodované aktíva patria akcie, indexy, komodity, meny a opcie. Avšak toto ani zďaleka nie sú všetky, keďže niektoré banky a brokeri poskytujú špeciálne aktíva, ktoré nie sú bežne obchodované.

Akcie sú základný investičný nástroj a predstavujú hodnotu spoločnosti. Akcia potvrdzuje, že jej držiteľ vložil svoj majetok do spoločnosti a stáva sa jej spoluvlastníkom. Držiteľ akcie získava rôzne práva, ako napríklad právo na dividendu, hlasovanie na valnej hromade spoločnosti, prípadne aj riadenie spoločnosti. Pre začínajúcich obchodníkov nie sú akcie najvhodnejším aktívom pre obchodovanie napríklad preto, že dividendu je potrebné zdaniť, výplata výnosu je nepravidelná a minimálna možná investícia je relatívne vysoká. Z toho plynie, že akcie sú skôr dlhodobým investičným nástrojom.

Indexy v sebe zahŕňajú vždy priemer hodnôt niekoľkých iných aktív. Najznámejšie sú indexy akciové, ktoré predstavujú priemer akcií spoločností obchodovaných na niektorej burze. Indexy sú pre začiatočníkov jedným z najvhodnejších nástrojom práve preto, že ich obchodovanie eliminuje niekoľko rizík, ako napríklad to, že pád jednej akcie zapríčiní masívnu stratu. Ďalej je dôležité to, že likvidita indexov je veľmi vysoká a vyhľadanie ponuky k dopytu je otázkou milisekúnd. Indexy môžu byť sektorové (zhrňujúce bankové a finančné inštitúcie kótované na burzách) alebo stanovené. Medzi najznámejšie indexy patrí: S&P 500, Dow Jones, NASDAQ Composite, Nikkei alebo Index PX (oficiálny cenový index Burzy cenných papierov Praha) [13].

Pod pojmom komodity (futures) rozumieme tovar, ktorý má fyzický podklad, ako napríklad ropa, zlato, meď, kukurica, mäso a ďalšie. Určité komodity ako ropa a zlato majú veľmi dobrú likviditu a obchodujú sa vo veľkom. Komodity sú avšak rizikovejším aktívom než indexy a akcie a obchodujú sa na páku. Ako už bolo spomínané pri komoditných burzách, komodity sa obchodujú v lotoch. V praxi to znamená, že ak má napríklad ropa lot o veľkosti 1000 a cena ropy je 80 USD za barel, jeden lot má hodnotu 80000 USD. Obchodník však nemusí zaplatiť 80000 USD ale zaplatí iba zálohu, napríklad 1/12 ceny lotu, v tomto prípade 6666 USD. Kedysi platilo, že obchodovanie komodít bolo vhodné skôr pre skúsenejších obchodníkov s väčším účtom, ale dnes je vďaka tzv. komoditným mini kontraktom, alebo už spomínaných ETF dostupné aj pre menších obchodníkov a začiatočníkov.

Menové páry predstavujú najviac obchodovaný trh na svete a ako už bolo spomenuté, každý deň sa na tomto trhu otočí rádovo trilióny dolárov. Najviac obchodovaným menovým párom na svete je EURUSD. V prípade mien nekupuje nikdy obchodník iba jednu konkrétnu menu ale vždy menový pár. Napríklad EURUSD znamená menový pár euro voči doláru, pričom hodnota páru je vždy podiel aktuálnych cien jednotlivých mien. Ak je v spomínanom páre aktuálny kurz 1,12 znamená to, že za jedno euro je možné kúpiť 1,12 doláru. K základným menám, ktoré tvoria hlavné menové páry patria: euro, americký dolár, japonský jen, švajčiarsky frank a anglická libra. Prvá mena v menovom páre sa nazýva základná a druhá sa nazýva krížová. Z toho vyplýva, že menové páry sú vzájomne previazané a vzniká medzi nimi korelácia. Pohyb na jednom trhu, ktorý obsahuje EUR ovplyvní všetky ostatné trhy, ktoré tiež obsahujú menu EUR. Hlavné výhody menových párov sú: obchodovanie na vysokú páku, už spomínaná vysoká likvidita a nízke poplatky spolu s jednoduchým vstupom na forexový trh [8].

Opcia je dohoda medzi dvomi stranami, predávajúcim a kupujúcim, kde kupujúci získava právo nakúpiť alebo predať dohodnuté množstvo podkladového aktíva za predom zjednanú cenu. Táto predom zjednaná cena sa nazýva realizačná cena alebo strike cena. Opcie sa delia na call a put opcie. Call opcia, čiže kúpna opcia, je právo na nákup podkladového aktíva v stanovenom termíne za stanovenú cenu. Put opcia, čiže predajná opcia, je naopak právo na predaj podkladového aktíva. Veľmi zjednodušene povedané, obchodník, ktorý obchoduje opcie vsádza na jednu z dvoch možností, akým sa trh bude vyvíjať a či sa trh za určitý čas bude nachádzať nad, alebo pod aktuálnou cenou [7].

2.2 Technická analýza

Technická analýza je jedným zo základných spôsobov analýzy aktív s cieľom predikcie ich budúceho vývoja pomocou rôznych technických metód. Na rozdiel od fundamentálnej alebo psychologickej analýzy pristupuje technická analýza k trhom ako k číslam a nezameriava sa na politické a ekonomické správy. Technický analytik sústreďuje svoju pozornosť na cenový graf, jeho tvar a rôzne indikátory odvodené z cien podkladového aktíva a objemu uskutočnených obchodov. Základnou myšlienkou tohto typu analýzy je to, že všetky informácie, ktoré by mohli mať vplyv na budúci vývoj ceny aktíva sa v danom cenovom grafe už prejavili. Aj pre obchodníkov, ktorí sa venujú fundamentálnej analýze a snažia sa čo najrýchlejšie zachytiť novinky zo zdrojov ako sú noviny, správy a rôzne internetové portály je dôležité pochopiť aspoň základy technickej analýzy a vedieť sa orientovať v grafoch.

Existuje niekoľko základných predpokladov technickej analýzy:

- Všetky udalosti a informácie ovplyvňujúce trh sú už zahrnuté v cene.
- Ceny sa nepohybujú náhodne, ale v trendoch (prúdoch).
- História vývoja ceny má tendenciu sa opakovať.
- Čo sa odohráva s výslednou cenou je dôležitejšie ako prečo sa to odohráva.

Matematici, štatistici, analytici a programátori zakladajú technickú analýzu na týchto očakávaniach. Technická analýza môže byť zdĺhavá, zložitá a časovo náročná a práve preto obchodníci vyhľadávajú programy podobné nástroju pre analýzu grafu, ktorého návrh je jedným z cieľov tejto práce – programy, ktoré za obchodníka vypočítajú a na grafe znázornia všetky informácie, ktoré sú zaujímavé.

Veľké množstvo súčasných metód technickej analýzy pochádza z princípov analýzy časových radov. Časový rad je jednoducho povedané chronologicky usporiadaná postupnosť hodnôt určitého štatistického ukazovateľa [5].

Pri technickej analýze tržných dát skúmajú analytici dve základné časti, ktorými sú obrazce (formácie, patterns) a indikátory. Väčšina informácií o tejto téme pochádza z literatúry [6] a [9].

2.2.1 Grafové formácie

Grafové formácie (patterns) sú obrazce v grafe, ktoré predznačujú pravdepodobný budúci vývoj ceny – jej nárast alebo pokles. Samozrejme, že po ich výskyte nie je isté, že sa cena skutočne bude pohybovať tak, ako to formácia naznačuje, ale určite zvyšujú pravdepodobnosť vývoju ceny tým smerom. Patterns značia, že cena bude pokračovať vo svojom trende alebo naopak predznačujú, že cena tento trend poruší a trend sa zmení na opačný. Najznámejšie grafové formácie a útvary sú: trendové čiary, supporty (podpora) a rezistencie (odpor), trojuholníkové formácie, dvojité dno, dvojitý vrchol, hlava a ramená, kanály atď. Existujú taktiež formácie, ktoré sú viditeľné iba na sviečkových (candlestick) grafoch. Najznámejšie candlestick patterns sú: doji, dragonfly doji, gravestone doji, hammer, inverted hammer, shooting star, engulfing candle atď. Viac o sviečkových grafoch a iných typoch grafov je možné nájsť v podkapitole 2.4.1, kde je vysvetlený ich princíp a použitie priamo v obchodnej platforme NinjaTrader. Grafové formácie reflektujú skutočné kroky obchodníkov na trhu a ich správna interpretácia môže taktiež pomôcť k uskutočneniu výnosného obchodu. Existujú teórie, že formácie sa vytvárajú a „fungujú“ iba preto, že sa ich ľudia naučili používať a veria v ich naplnenie.

Na obrázku 2.1 je zobrazená ukážka grafovej formácie trojuholník, ktorej výskyt úspešne predpovedal pokles ceny na forexovom trhu EURUSD.

Obr. 2.1: Grafová formácia trojuholník

2.2.2 Indikátory

Indikátory sú rôzne vzorce, ktorých výpočtom je možné získať informácie, ktoré sú na prvý pohľad ťažko viditeľné alebo z grafu nejasné. Väčšinou zobrazujú dáta z trhov v krivkách, bodoch alebo ako číselný výstup. Indikátory je možné rozdeliť na trendové ukazovatele a oscilátory [9].

Trendové ukazovatele vyjadrujú smer trhu a zobrazujú, či je trh v trende, kde nastal zlom v trende a kde je možné očakávať ďalšiu zmenu. Najpoužívanejšími indikátormi v tejto skupine sú rôzne druhy priemerov a ich používanie sa riadi myšlienkou jednoduchosti. Priemery sa dajú vytvárať z ľubovoľných hodnôt, z ďalších ukazovateľov, no najčastejšie sú vytvárané priamo z cien za určité obdobie. Keď cena vybočí nad alebo pod sledovaný priemer, môže to znamenať zmenu trhu daného aktíva. Najčastejšie používanými priemermi sú aritmetické (prostý priemer sčítaných hodnôt) a exponenciálny (odmocnina súčtu hodnôt). Drvivá väčšina trendových ukazovateľov pracuje s priemermi, štandardnými odchýlkami a ich rôznymi kombináciami. Medzi najznámejšie trendové ukazovatele patria kĺzavé priemery, MACD alebo Bollingerove pásma. Na obrázku 2.2 je zobrazený trendový indikátor Bollinger Bands.

Obr. 2.2: Indikátor Bollinger Bands

Oscilátory merajú silu trhu a s ich pomocou je možné sledovať aktuálnu náladu a možný smer trhu. Sila trhu môže znamenať práve prekúpený alebo prepredaný trh. Oscilátory, ako už ich názov napovedá, oscilujú v rozmedzí nejakých hodnôt, napríklad 0 až 1 alebo -1 až 1 apod. Najrozšírenejším oscilátorom je RSI, hlavne kvôli jeho jednoduchosti. Nadobúda hodnôt od 0 do 100, pričom čím bližšie je k nule, tým je trh viac prepredaný a obchodník môže očakávať príchod nákupcov na trh. Naopak, čím je hodnota oscilátoru bližšie ku hranici sto, tým je trh viacej prekúpený a je možné očakávať pokles ceny na trhu. V praxi sa oscilátor nikdy nepriblíži okrajovým hodnotám, ale pohybuje sa v rozmedzí 30 až 70, ktoré na vyššie uvedenom princípe efektívne znázorňujú informáciu o prekúpenom alebo prepredanom trhu. Okrem RSI sú využívané ďalšie oscilátory ako napríklad Stochastic alebo CCI, ktoré fungujú na podobnom princípe ako RSI [9]. Na obrázku 2.3 je znázornený oscilátor RSI.

Obr. 2.3: Indikátor RSI

2.3 Automatické obchodné systémy

Automatický (automatizovaný) obchodný systém (AOS) je akákoľvek stratégia, ktorá sa vykonáva nezávisle na rozhodnutí obchodníka, ale podľa pravidiel ním zadaných. V dnešnej dobe je odhadované, že viac ako 80% všetkých obchodov vykonávajú tieto systémy [9]. Na používanie AOS je nutné mať program, ktorý je pripojený na burzu a umožňuje definovať pravidlá stratégie. Tieto pravidlá v podstate definujú vstupy na trh a výstupy z neho. Hlavnou výhodou AOS je samozrejme možnosť súbežne sledovať desiatky trhov 24 hodín denne, čo človek len ťažko dokáže. Ďalšou veľkou výhodou je eliminácia psychologických faktorov. To znamená, že počítačový program nemá emócie, nepociťuje únavu, nestráca koncentráciu a jeho rozhodnutia teda nemôžu byť týmto spôsobom ovplyvnené, naopak jeho rozhodnutia sú založené na presne definovaných pravidlách. Na druhej strane je treba podotknúť, že používane AOS môže mať aj svoje nevýhody. Hlavné riziká jeho používania sú spojené so stabilitou prostredia, v ktorom program beží. Preto nie je vhodné, aby program bežal na počítači, ktorý používajú iný ľudia, nemá záložný zdroj energie a stabilné pripojenie k internetu. Ďalším veľkým rizikom je podcenenie backtestingu (overovanie funkčnosti AOS na historických dátach). Ak obchodník túto časť podcení, program v niektorých situáciách môže prestať správne fungovať a najhoršie je, že takto vzniknutá chyba sa nemusí prejaviť ihneď a počas istého časového intervalu sa môže javiť, že program stále pracuje správne.

Obrovským problémom pri tvorbe AOS je to, že obchodníci dostanú nápad na stratégiu, ktorá by mohla byť výnosná, ale podcenia backtesting tejto stratégie. Na prvý pohľad sa môže zdať, že je stratégia výnosná a funguje bez problémov, avšak môže nastať situácia, na ktorú nebol program pripravený a rýchlo premení zisk na stratu.

V mojej práci si dovolím mierne upraviť klasickú definíciu AOS a budem predpokladať, že pojem AOS znamená súbor všetkých nástrojov a častí, ktoré sa podieľajú na vyhľadaní a identifikácii stratégie, definovaní vstupných a výstupných signálov, spravovaní objednávok a vytváraní štatistiky úspešnosti stratégie. Tento prístup umožňuje efektívne rozdeliť AOS na tri časti, ktoré môžu, ale aj nemusia byť na sebe závislé. Prvou časťou je nástroj, ktorý dokáže spracovať historické dáta (nástroj, ktorého tvorba je hlavným bodom tejto práce).

Ďalšia časť je samotné spracovanie aktuálnych dát na trhu a vyhodnocovanie vstupných a výstupných signálov a poslednou časťou je systém na správu objednávok a zobrazovanie štatistík o vykonaných obchodoch.

2.3.1 Spracovanie historických dát

Činnosť nástroju, ktorý spracováva historické dáta, spočíva v technickej analýze grafu na základe pokynov užívateľa. Spracovaním výsledkov, ktoré tento nástroj ponúka je možné vytvoriť novú obchodnú stratégiu, otestovať existujúcu stratégiu, alebo vizualizovať priebeh vývoja ceny na určitom trhu za určitý čas. Mnoho obchodníkov, ktorý veria vo výsledok technickej analýzy, analyzuje grafy „ručne“. Tento prístup je nevýhodný, pretože do hry opäť vstupuje ľudský faktor a ľudská chyba. Samozrejme je možné výsledky technickej analýzy kresliť manuálne a presne, ale časová náročnosť takéhoto úkonu je extrémne vysoká.

Kvalitných nástrojov na analýzu grafov je však nedostatok a preto obchodníci volia prístup taký, že po náročnej manuálnej technickej analýze istého trhu dospejú k stratégii, ktorú chcú zautomatizovať a preto zaplatia nemalé peniaze na jej vývoj, prípadne obetujú veľa času na jej vývoj. Až po vytvorení tejto stratégie ju otestujú na historických dátach a zistia, že nie je tak zisková ako čakali. Presne z tohto dôvodu je potrebné ešte pred vytváraním samotnej stratégie analyzovať graf pokiaľ možno na čo najväčšom objeme čo najpresnejších dát. Keďže túto analýzu vykoná počítačový program, je možné získať technickú analýzu veľkého časového intervalu presných dát v reálnom čase, čo by pri manuálnom prístupe nebolo možné.

2.3.2 Spracovanie aktuálnych dát

Časť AOS, ktorá spracuje aktuálne (live) dáta z nejakého trhu, priamo odpovedá pôvodnej, klasickej definícii AOS. Činnosť tejto časti spočíva v konštantnom sledovaní jedného, alebo viacerých trhov a analýze dát. Každú novú informáciu o zmene ceny aktíva spracováva a vyhodnotí všetky podmienky, ktoré určujú vstup alebo výstup z trhu. Vytváranie tejto časti je časovo náročné a najviac dôležité je nepodceniť spätné testovanie stratégie. Táto časť môže, ale aj nemusí, byť prepojená s nástrojom pre analýzu grafu a s nástrojom, ktorý sa stará o správu objednávok.

2.3.3 Správa objednávok

Systém správy objednávok nie je nevyhnutný pre základné fungovanie AOS, avšak v niektorých prípadoch dokáže veľmi pomôcť. Príkladom môže byť obchodná firma, ktorá používa niekoľko typov AOS, ktoré môžu byť založené na diametrálne odlišných stratégiách a pôsobiť na rozličných trhoch. V takomto prípade je v záujme firmy zariadiť si systém, ktorý bude združovať obchodné príkazy z týchto AOS, vyhodnocovať viaceré rovnaké objednávky a následne ich vykonávať. Zavedením tohto systému je možné všetky objednávky spojiť s jedným (firemným) účtom.

Existujú obchodníci, ktorí veria, že brokeri môžu zneužívať ich stratégie. Napríklad objednávky, ktoré sa majú vykonávať až za určitý čas (pending orders) sú umiestňované na server brokera. V takomto prípade nevie obchodník zistiť, či objednávky nebudú zneužité, pretože pre brokera je výhodné ho z obchodu vyradiť. Overovanie, či je táto teória pravdivá je mimo rozsahu tejto práce a každý obchodník si na túto problematiku musí vytvoriť svoj vlastný názor a podľa neho sa zariadiť. Systém správy objednávok by tento problém mohol

riešiť tak, že združené pending orders by uchovával na vlastnom úložisku a signál pre nákup a predaj by vykonával až vtedy, keď nastane správny čas.

2.4 Obchodná platforma NinjaTrader

Obchodná platforma je elektronický systém, ktorý umožňuje vykonávať rôzne operácie, nakupovať a predávať aktíva alebo meny. Existujú platformy, ktoré sa zameriavajú napríklad iba na komodity, akcie, forex alebo na všetky aktíva. Obchodné platformy v dnešnej dobe poskytujú jednotlivcom prístup na trh v rozsahu, ktorý bol kedysi dostupný iba pre špecializované obchodné spoločnosti. Každý broker má väčšinou svoju vlastnú platformu, ktorú sa snaží obchodníkom vnútiť. Existujú ale platformy, ktoré sú podporované viacerými brokermi. Kvalitná obchodná platforma by mala fungovať nonstop, mala by mať vysokú rýchlosť exekúcie objednávok, nízke požiadavky na rýchlosť pripojenia a systémové nároky. Ďalej by mala umožňovať vykresľovať grafy, vykonávať analýzu týchto grafov pomocou indikátorov, ukazovateľov, grafických znázornení a ďalších pomôcok. Dôležitou súčasťou dobrej platformy je aj možnosť vytvárania vlastných obchodných stratégii, ktoré sú samozrejme automatizované, a taktiež umožňuje tieto stratégie naprogramovať v niektorom zo známych programovacích jazykov.

NinjaTrader je jedna z najrozšírenejších obchodných platforiem na svete a poskytuje všetky dôležité funkcie potrebné pre pohodlné obchodovanie, vizualizáciu trhu, testovanie a optimalizáciu obchodných nápadov a mnoho ďalších. NinjaTrader spolupracuje s rôznymi brokermi a umožňuje rôzne možnosti získavania trhových dát. Zaujímavou funkciou NinjaTraderu je možnosť prehrávať, pozastavovať a pretáčať historické dáta. Táto simulácia je veľmi mocným nástrojom na overovanie a testovanie správnej funkčnosti stratégií [4].

V nasledujúcich podkapitolách budú predstavené a popísané niektoré dôležité časti obchodnej platformy NinjaTrader, ktoré budú využívané vyvíjaným nástrojom pre analýzu grafu.

2.4.1 Grafy

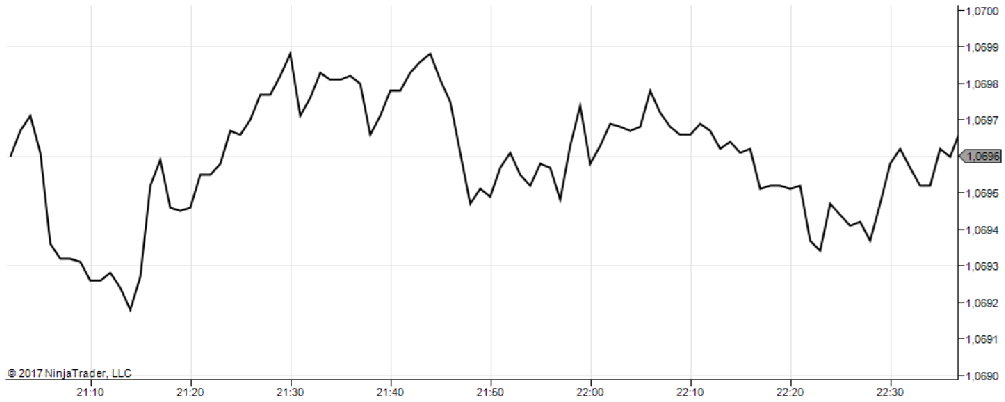
Pri technickej analýze a obchodovaní samotnom sa využíva veľké množstvo kombinácii jednotlivých ukazovateľov a indikátorov. Najzákladnejším prvkom sú ale grafy samotné. Pri zobrazovaní vývoja ceny určitého aktíva na nejakom trhu platí, že na vodorovnej ose grafu je zobrazený čas a na zvislej ose to môže byť kurz, body, percentá alebo iná hodnota, ktorá je na danom trhu skúmaná a je závislá na čase. Existuje obrovské množstvo typov grafov, pričom najpoužívanejšie sú tieto dva:

- **Líniový graf:**

  Tento typ grafu je tým najjednoduchším a vo finančníctve najpoužívanejším. Jednotlivé minulé dáta sú spojené čiarou, pričom väčšinou sa používajú uzavieracie (close) ceny, teda tie, ktoré nastali na konci nejakého časového úseku. Ukážku líniového grafu je možné vidieť na obrázku 2.4.

- **Sviečkový graf:**

  Základom sviečkových grafov (candlestick) je jedna sviečka, ktorá reprezentuje základné hodnoty daného časového rozpätia. Sviečky majú rôznu farbu pre prípad, kedy je výsledný pohyb kladný a kedy je záporný. Sú tvorené hrubým telom, ktoré ohraničuje hodnoty open a close a tenkými čiarami, ktoré znázorňujú hladiny low a high.



Obr. 2.4: Líniový graf

Na obrázku 2.5 je ukážka sviečkového grafu, pričom na porovnanie a vizualizáciu rozdielov sú použité rovnaké dáta ako pri líniovom grafe. Grafy sú vytvorené priamo pomocou obchodnej platformy NinjaTrader.

Obr. 2.5: Sviečkový graf

2.4.2 Stratégie

Stratégie sú v platforme NinjaTrader nástroje, pomocou ktorých je možné definovať pravidlá automatizovaného obchodného systému, tým pádom takýto AOS vytvoriť. NinjaTrader umožňuje vytvoriť tieto pravidlá (resp. celú stratégiu) pomocou pomocníka (Wizard) alebo pomocou programovacieho jazyka NinjaScript. Stratégie je možné podľa potreby zapínať, vypínať a upravovať a môžu mať rôzny význam, teda nemusia slúžiť iba na definovanie pravidiel vstupu, výstupu, hraníc profitu a straty. V nástroji pre analýzu grafu budú stratégie použité na prevádzanie dát medzi NinjaTraderom a vlastnou aplikáciou a na vykresľovanie objektov podľa zadaných parametrov.

2.4.3 Jazyk NinjaScript

NinjaScript je programovací jazyk, ktorý je nadstavbou nad programovacím jazykom C#. Jazyk NinjaScript teda dokáže všetko to, čo aj C# a naviac obsahuje rôzne mechanizmy a funkcie pre jednoduché zadávanie obchodných príkazov, prácu s dátami a indikátormi, vyznačovanie rôznych informácií v grafe a mnoho ďalšieho. Môže byť použitý pre vývoj vlastných indikátorov ako aj pre vývoj celých obchodných stratégii.

Platforma NinjaTrader obsahuje vývojové prostredie, ktoré je veľmi jednoduché, a tým neodradí ani začiatočníkov a obchodníkov, ktorí nemajú skúsenosti s programovaním.

2.5 Vymedzenie základných pojmov

V tejto kapitole budú vysvetlené niektoré pojmy, ktorých pochopenie je nevyhnutné pre pochopenie konceptu obchodovania a pre pochopenie ďalších kapitol, v ktorých budú tieto pojmy používané. Niektoré výrazy boli už vyššie v texte spomenuté, ale je vhodné, aby boli združené na jednom mieste a vysvetlené. Zdrojom pre objasnenie väčšiny pojmov je literatúra [9].

- **Broker** – spoločnosť, ktorá zaisťuje spojenie obchodníkov s burzou.
- **Long pozícia** – označenie tzv. dlhej pozície, čiže keď obchodník očakáva nárast na trhu.
- **Short pozícia** – označenie tzv. krátkej pozície, čiže keď obchodník očakáva pokles na trhu.
- **Leverage (páka)** – vyjadrená pomerom dvoch čísiel, napríklad 1:100. Daný pomer znamená, že obchodníkovi postačí 1% z požadovanej ceny na to, aby mohol kontrolovať celý objem, teda 100%. Napríklad pri nákupe jedného lotu menového páru EUR/USD, ktorý predstavuje 100 000 EUR, s pákou 1:100 znamená, že obchodníkovi stačí 1000 EUR na to, aby mohol kontrolovať celú objem. Zvyšok, 99 000 EUR za obchodníka doplní broker, nie však zadarmo.
- **Market príkaz** – príkaz na vstup do trhu za aktuálnu najlepšiu možnú cenu čo najrýchlejšie.
- **Limit príkaz** – príkaz na vstup do trhu za cenu, ktorú obchodník určí, a za ktorú chce požadované aktívum získať. Vstup sa udeje vždy za najlepšiu možnú cenu pod limitom.
- **Stop príkaz** – príkaz, pri ktorom sa vstup, prípadne výstup z trhu udeje pri prekročení istej hladiny, ktorú obchodník určí.
- **Stop limit** – príkaz, ktorý pri prekročení istej hranice aktivuje stop príkaz a vytvorí limit príkaz, ktorý ďalej čaká na vyplnenie. Tento typ príkazu je vhodné používať pri nelikvidných trhoch.
- **Breakeven** – bod zvratu, inak povedané je to okamih, kedy príslušná pozícia na trhu nie je zisková ani stratová.

- **Spread** – rozdiel medzi cenami ponuky a dopytu. V praxi sa používajú pojmy bid a ask. Na forexovom trhu je spread spôsob, akým si brokeri vyberajú svoje provízie za sprostredkovanie obchodov.
- **Stop loss** – hodnota, ktorú obchodník nastaví pre výstup z obchodu s istou stratou. Používa sa ako ochranný prvok pred obrovskými stratami.
- **Profit target** – hodnota, ktorú obchodník nastaví pre výstup z obchodu s istým ziskom. Používa sa pre zaistenie zisku z obchodu a zamedzenie prípadného zvratu a následnej strate.
- **Pip** – najmenšia zmena kurzu. Pri väčšine menových párov je to štvrté desatinné miesto a pri pároch obsahujúcich japonský jen je to druhé desatinné miesto.

Kapitola 3

Špecifikácia a analýza požiadaviek

Pred vytvorením návrhu systému je dôležité ujasniť si, aké požiadavky sú na systém kladené a čo sa od neho očakáva. Požiadavky na nástroj pre analýzu grafu sú výsledkom vzájomnej dohody medzi mnou a vedúcim mojej práce.

Jednotlivé komponenty systému musia byť rozdelené, teda na sebe nezávislé, v zmysle softvérovej architektúry MVC (Model-view-controller) alebo MVP (Model-view-presenter). Systém musí byť jednoducho rozšíriteľný, pripravený na vylúčenie platformy NinjaTrader, prípadne zmenu na inú platformu a jeho návrh musí podporovať možnosť jednoduchého pridávania nových analytických modulov.

Každý analytický modul bude vykonávať špecifickú analýzu nad grafom. Z týchto modulov musí byť možné vytvoriť postupnosť takú, že vstup do modulu aktuálneho bude odpovedať výstupu z modulu prechádzajúceho a výstup modulu aktuálneho bude možné pripojiť na vstup modulu nasledujúceho. Táto postupnosť bude samozrejme vytváraná iba pre moduly, pre ktoré to bude mať praktický význam a nebude možné spojiť dva moduly, ktoré oba poskytujú výsledok taký, že nad ním nie je potrebné vykonávať žiadnu ďalšiu analýzu.

Výsledkom analýzy s využitím jedného modulu alebo postupnosti modulov bude skupina záznamov. Jeden záznam predstavuje všetky údaje o cene v určitom čase spolu s výsledkami všetkých analytických modulov, ktorých účasť bola na danej analýze vyžadovaná.

Ďalšou požiadavkou na vyvíjaný systém je možnosť vzniknuté skupiny záznamov vhodným spôsobom ukladať, uložené analýzy prezerať a nechať znova vykresliť, prípadne odstrániť.

Výsledky analýzy budú po načítaní zakresľované priamo do grafu, ktorý predstavoval zdroj dát pre danú analýzu. Užívateľovi bude umožnené jednotlivé skupiny záznamov exportovať a tým mu bude umožnené jednoducho zobraziť tieto údaje napríklad v nejakom tabuľkovom kalkulátori.

V tejto kapitole sú postupne popísané jednotlivé prípady použitia systému a detailnejší pohľad na požiadavky týkajúce sa uchovávania dát.

3.1 Prípady použitia

Na základe požiadaviek na systém je vhodné vypracovať diagram prípadov použitia (angl. use case diagram). Ten slúži na zobrazenie funkcionality systému z pohľadu užívateľa. Pri jeho tvorbe je dôležité ujasniť si, koľko typov užívateľov bude systém obsluhovať a aké jednotlivé prípady použitia a väzby bude obsahovať.

Vyvíjaný systém bude podporovať iba jeden typ užívateľa, ktorému budú dostupné všetky možnosti, ktoré systém poskytuje. Užívateľ bude pracovať v dvoch aplikáciach, z ktorých jednou je obchodná platforma NinjaTrader. Akcie, ktoré môže užívateľ vykonávať prostredníctvom tejto platformy a súvisia s nástrojom pre analýzu grafu sú:

- **Zobraziť graf pre určité obdobie, určitý trh:** Systém umožňuje užívateľovi zobraziť graf vývoju ceny pre určitý inštrument. Obdobie a typ grafu, ktorý je možné zobraziť, je limitované dátami, ktoré má užívateľ k dispozícii a boli importované do NinjaTraderu.
- **Pridať stratégiu:** Užívateľ môže na zobrazený graf pridať ľubovoľnú stratégiu. Stratégia, ktorá vykonáva analýzu nad vybraným grafom a výsledok ukladá a taktiež stratégia, ktorá výsledky vykresľuje budú predpripravené.
- **Odobrať stratégiu:** Po úspešnej analýze má užívateľ možnosť stratégiu odstrániť, alebo nahradiť inou v prípade, že už nechce vykonávať analýzu, ale chce výsledky vykresliť.

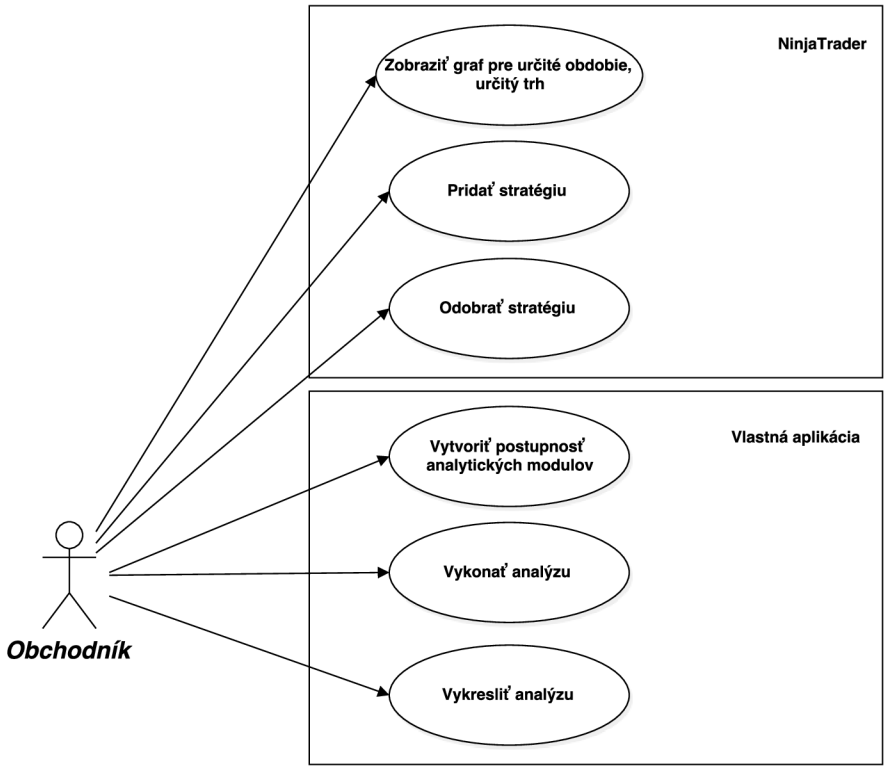
Nasledujúce akcie sú pre užívateľa dostupné prostredníctvom vyvíjanej časti automatického obchodného systému, konkrétne prostredníctvom užívateľského rozhrania, ktoré poskytuje:

- **Vytvoriť postupnosť analytických modulov:** Z jednotlivých analytických modulov umožňuje aplikácia užívateľovi vytvoriť postupnosť, za cieľom získania požadovaného výsledku.
- **Vykonať analýzu:** Po úspešnej analýze vznikne výsledok v podobe skupiny záznamov, ktorý bude vhodne uložený tak, aby bolo možné túto analýzu spätne dohľadať, vykresliť, upraviť, prípadne odstrániť. Pred týmto krokom musí byť v obchodnej platforme NinjaTrader aktívna predpripravená stratégia určená pre vykonanie novej analýzy.
- **Vykresliť analýzu:** Po vyhľadaní konkrétnej analýzy má užívateľ možnosť výsledky zakresliť do grafu, ktorý musí obsahovať rovnaké dáta ako tie, ktoré boli použité pre vytvorenie analýzy a jej uloženie. Pred týmto krokom musí byť v obchodnej platforme NinjaTrader aktívna predpripravená stratégia určená pre vykresľovanie výsledkov.

Na obrázku 3.1 je zobrazený diagram, ktorý vizualizuje vyššie popísané prípady použitia (angl. use case diagram).

3.2 Uchovávanie dát

Vyvíjaná časť automatického obchodného systému musí byť schopná ukladať skupiny záznamov, ktoré vznikli analýzou nad určitým grafom. Požiadavkou je, aby dáta týchto záznamov boli uchovávané takým spôsobom, aby prístup k nim bol rýchly a bolo možné dáta triediť a vyhľadávať v nich na základe istých kritérií. Pod pojmom skupina záznamov (predstavuje jednu analýzu) sa myslí celá časť časovej rady, ktorá bola skúmaná. To znamená všetky hodnoty ceny daného inštrumentu spolu s presným časom a dátumom, hodnotami všetkých použitých analytických modulov a v prípade, že sú k dispozícií, aj dáta reprezentujúce objem obchodov (indikátor vyjadrujúci celkové množstvo kontraktov zobchodovaných v rámci



Obr. 3.1: Diagram prípadov použitia

špecifického časového úseku). K analýze bude užívateľovi umožnené uložiť dátum a čas vykonania analýzy, rozsah vstupných dát, úroveň agregácie vstupných dát a trh, na ktorom bola analýza vykonávaná (EUR/USD, USD/JPY atď.).

Kapitola 4

Návrh aplikácie

Po dôkladnej analýze požiadaviek nástroju pre analýzu grafu nasleduje vytvorenie detailného návrhu tohto systému. Prvým krokom je dekompozícia problému na podproblémy, ktorých riešenie bude podrobne popísané v tejto kapitole.

Prvým podproblémom je samotné rozdelenie funkčnosti do dvoch aplikácii – NinjaTraderu a vyvíjanej časti automatického obchodného systému. Tieto dve aplikácie si budú musieť dokázať predávať medzi sebou dáta, takže ďalším riešeným problémom je komunikácia aplikácií v systéme. Ďalej je potrebné vyriešiť problém ukladania záznamov efektívnym spôsobom. Keď už aplikácia dokáže komunikovať s obchodnou platformou, ukladať záznamy a zobrazovať výsledky, stále jej chýba spôsob, akým bude komunikovať s užívateľom, čiže vhodné užívateľské rozhranie.

4.1 Architektúra systému

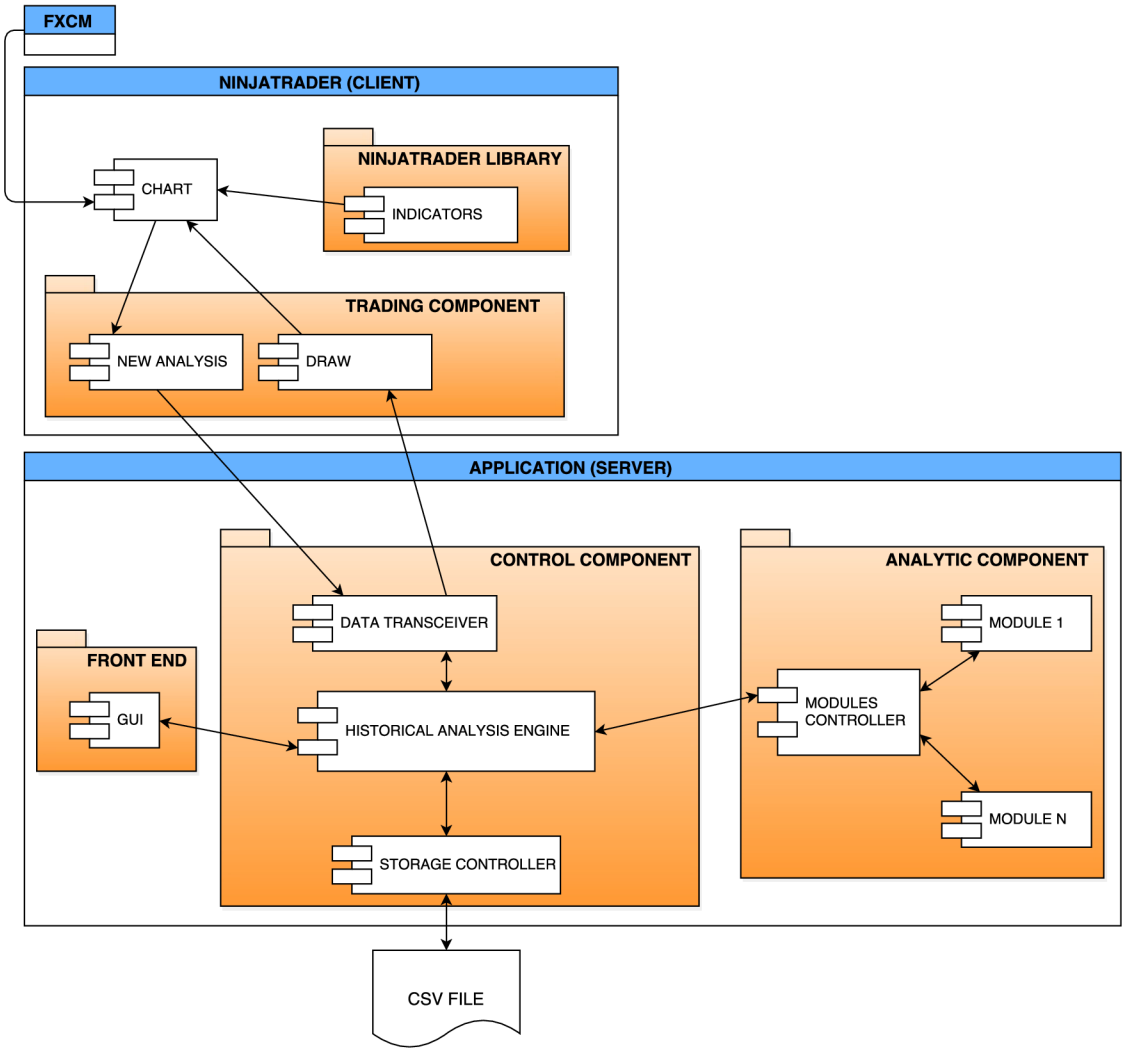
Keďže nástroj pre analýzu grafu bude využívať funkcie obchodnej platformy NinjaTrader, je nutné si ujasniť, ako bude systém rozdelený, ktoré funkcie poskytne NinjaTrader, a ktoré naopak budú obsahom vyvíjanej aplikácie.

NinjaTrader bude v systéme predstavovať zdroj dát pre analýzu a taktiež nástroj, ktorý bude slúžiť na vizualizáciu výsledkov analýzy. Činnosť tejto časti bude teda spočívať v efektívnom presune dát do vyvíjanej aplikácie a prijímaní vykresľovacích inštrukcií.

Za ďalšiu časť systému by sa dal považovať originálny zdroj dát, ktorým bude broker FXCM [1], ktorý podporuje prepojenie s NinjaTraderom a pre ďalší výklad bude táto časť považovaná za spojenú s podsystémom v obchodnej platforme.

Aplikácia vyvíjaného nástroja bude stredobodom systému, keďže bude obsahovať celú logiku spracovania dát. Zo zadaných požiadaviek na systém vyplýva, že systém musí byť pripravený na úplne vyčlenenie obchodnej platformy alebo jej zmenu. Táto podmienka je splnená tým, že všetky analytické moduly a ostatné moduly aplikácie, ktoré sa podieľajú na analýze sú umiestnené v tejto časti a nie v stratégii obchodnej platformy. Pri potrebe vyčleniť obchodnú platformu zo systému stačí zvoliť iný zdroj dát a zvoliť iný spôsob vizualizácie výsledkov analýzy, napríklad iný softvér, prípadne by mohla postačiť aj možnosť exportu výsledkov do formátu csv, ktorý je podporovaný väčšinou kvalitných tabuľkových kalkulátorov.

Obrázok 4.1 znázorňuje architektúru systému, jeho rozdelenie na podsystémy, komponenty a moduly.



Obr. 4.1: Architektúra systému

4.1.1 Podsystém v obchodnej platforme

Podľa môjho názoru, najjednoduchším spôsobom, ako vysvetliť princíp navrhnutého systému je pohľadom na tok dát v systéme. Tok dát v obchodnej platforme je nutné rozdeliť na dve časti. Za prvú časť je považovaná doba od zaslania dát brokerom do odoslania potrebných dát do vyvíjanej aplikácie a za druhú časť je považovaná doba od prijatia inštrukcií, ktoré udávajú body v čase a spôsob zakreslenia výsledku analýzy pre ten konkrétny čas, po kompletnú vizualizáciu vyplývajúcu z vytvorenej postupnosti analytických modulov. Tieto dve časti toku sú na sebe nezávislé a ich vykonanie je súčasťou dvoch cyklov, čiže nenasledujú sekvenčne za sebou.

Tok dát začína momentom, keď NinjaTrader vyžiada istý objem historických dát od brokera. Tieto dáta vstupujú do platformy a sú ihneď premietané do grafu, ktorého typ si užívateľ určí. V tento moment má užívateľ k dispozícii ním zvolený rozsah dát historického vývoja ceny nejakého inštrumentu premietnutého do grafu a je pripravený zadať systému pokyn na analýzu tohto grafu alebo na vykreslenie analýzy, ktorá je už uložená.

Tieto pokyny sú rozlíšiteľné na úrovni stratégií NinjaTraderu. Pre každý z týchto pokynov existuje jedna stratégia, ktorej úlohou je buď prenos dát určených na analýzu, alebo zobrazenie výsledkov postupnosti analytických modulov. V prípade vykreslovacieho cyklu je možné do grafu dokresliť indikátory, ktoré platforma NinjaTrader poskytuje, prípadne vlastné indikátory, ktoré sú v platforme nahraté a pripravené na používanie.

Po zadaní príslušného pokynu sú dáta, prípadne inštrukcie pre výber, zhromaždené a prípravné na prenos do vyvíjanej aplikácie. Konkrétny spôsob prenosu dát a komunikácie medzi týmito podsystémami je popísaný v podkapitole 4.3.

4.1.2 Podsystém vo vyvíjanej aplikácii

Aplikácia rozlišuje dva hlavné cykly programu. Prvým je požiadavka na analýzu dát a druhým požiadavka na znázornenie situácii v už analyzovaných dátach. Pri pokyne užívateľa na novú analýzu prídu do aplikácie dáta s určitou úrovňou agregácie. Tieto dáta sú hneď predávané do modulu data transceiver, ktorého úlohou je prijímanie, združovanie a odosielanie dát, ktoré sú prijímané z NinjaTraderu alebo sú pripravované na odoslanie. Ďalšou dôležitou úlohou tohto modulu je prevedenie dát do takej dátovej štruktúry, ktorá umožňuje jednoduché spracovanie, analýzu, štatistiku dát.

Po prijatí a následnom vložení všetkých cenových dát do vhodnej dátovej štruktúry postupujú dáta do modulu historical analysis engine. Tento modul je jadrom celej aplikácie a jeho najdôležitejšími úlohami sú: riadiť celý proces analýzy, prijímať a vyhodnocovať vstupy od užívateľa, komunikovať s modulom pre ukladanie dát a modulom, ktorý má na starosti vykonávanie samotného výpočtu.

Užívateľov vstup, ktorý tvorí definícia postupnosti analytických modulov s určitými parametrami alebo požiadavka na vykreslenie výsledkov určitej analýzy je zaznamenaný pomocou grafického užívateľského rozhrania (GUI), ktorého výstup je naviazaný na modul historical analysis engine, ktorý si dáta od užívateľa z GUI vyberie v prípade potreby.

V tomto momente má modul engine všetky potrebné informácie aby vyhodnotil, čo chce vlastne užívateľ vykonať. Pokiaľ sa jedná o novú analýzu a jej následné uloženie, sú dáta s inštrukciami okamžite posielané do časti modules controller. Pokiaľ sa jedná o potrebu vizualizovať uloženú analýzu, potrebné inštrukcie sú zaslané modulu storage controller, ktorý je určený pre ukladanie a načítanie výsledkov analýzy.

V jednoduchšom prípade, teda v takom, kde užívateľ požaduje vizualizáciu výsledkov uloženej analýzy reaguje storage controller tak, že zo zvoleného súboru načíta všetky dáta, ktoré postupne premení na vykreslovacie inštrukcie. Tie sú predané modulu data transceiver a odoslané do NinjaTraderu, kde sú následne premenené na konkrétne geometrické útvary a iné zvolené prostriedky vyznačenia a vykreslovania zaujímavých údajov. Týmito krokmi je uzavretý cyklus vykresľovania uložených výsledkov.

Cyklus novej analýzy však skončil poslaním dátovej štruktúry spolu s užívateľovými inštrukciami do modulu modules controller. Význam jeho zavedenia spočíva v rozložení záťaže riadenia programu a odľahčení modulu engine, ktorý slúži ako rozhranie pre pracovanie so všetkými ostatnými časťami systému. Dáta sú v tomto momente pripravené na analýzu, zatiaľ čo controller rozošle inštrukcie analytickým modulom, ktorých činnosť je v požadovanej analýze žiadaná. Na základe postupnosti, ktorú určil užívateľ, sú dáta postupne spracovávané jednotlivými analytickými modulmi a sú vytvárané nové dátové štruktúry, obsahujúce výsledky každého analytického modulu. Všetky novo vytvorené štruktúry budú združené aj s pôvodnou štruktúrou predstavujúcou vstup do analýzy a zaslané späť do modulu engine, ktorý do tohto toku zasahuje tak, že pridá určité informácie o analýze, teda

dátum a čas, rozsah vstupných tržných dát a užívateľov komentár v podobe označenia trhu, úrovne agregácie vstupných dát a ľubovoľnej správy, ktorá napríklad stručne popíše význam zavedenia danej analýzy. Takto vytvorená štruktúra je presunutá do modulu storage controller, ktorý dáta uloží do súboru vo formáte CSV (Comma separated values) [11]. Formát uloženia týchto dát je vysvetlený v nasledujúcej kapitole. V tomto momente končí cyklus novej analýzy.

4.2 Formát súboru s výsledkami analýzy

Z analýzy požiadaviek na systém vyplýva, že aplikácia musí umožňovať výsledky analýz nejakým spôsobom ukladať a taktiež ich exportovať do súboru vo formáte CSV, aby ich bolo možné zobraziť v tabuľkovom kalkulátori. Tieto dve požiadavky je výhodné spojiť do jedného úkonu, čiže ukladanie výsledkov analýzy priamo do súboru CSV. Tento spôsob ukladania má oproti využitiu relačnej databázy niekoľko výhod: dáta sú radené priamo za sebou podľa času, súbory sú ľahko prenositeľné a môžu byť poskytnuté iným spolupracovníkom a môžu byť ľubovoľne usporiadané a organizované na disku, prípadne spoločnom úložisku. Klasický formát CSV používa znak čiarka na oddelovanie jednotlivých buniek, ale keďže čiarky môžu byť v tržných dátach používané na oddelenie desatinnej časti, aplikácia bude používať bodkočiarku ako separátor.

Súbor bude zložený z komentárovej časti, hlavičky pre jednotlivé stĺpce dát, príznakov upravujúcich využitie stĺpca pri náčítaní (napr. vykresliť alebo nevykresliť) a dát samotných.

Tento koncept je vizualizovaný pomocou tabuľky 4.1.

COMMENTS...							
TIME	**CLOSE**	**OPEN**	**HIGH**	**LOW**	**VOLUME**	**MOD$_1$**	**MOD$_N$**
						draw	draw
16.1.2017 10:00	1,05829	1,05936	1,06016	1,05795	7808	UPT	D
16.1.2017 10:15	1,05899	1,05829	1,05919	1,05826	5362	UPT	D
16.1.2017 10:30	1,05812	1,05899	1,05899	1,05803	4782	UPT	N

Tabuľka 4.1: Formát výsledného súboru

4.3 Komunikačné rozhranie podsystémov

Keďže je systém rozdelený na dve časti, je dôležité medzi nimi zaistiť správnu komunikáciu a teda výmenu informácii. Obe aplikácie budú spúšťané na jednom počítači, a preto stačí zabezpečiť takúto lokálnu komunikáciu.

Komunikácia pripomína svojim charakterom model klient-server, kde sa NinjaTrader správa ako klient a vlastná aplikácia ako server. NinjaTrader vždy iniciuje komunikáciu tým, že užívateľ spustí v grafe, ktorého analýzu žiada, príslušnú stratégiu. V prípade novej analýzy grafu táto stratégia začína na začiatku časového intervalu grafu a postupne zberá do vhodnej dátovej štruktúry všetky dáta. Tie sú následne naraz zaslané spolu s inštrukciami na analýzu, ktoré môžu byť nastavené v platforme ako parametre stratégie. Keďže nástroj pracuje s historickými dátami, bolo by nevýhodné posielať každú informáciu o cene v danom čase zvlášť, ale tento prístup je nutný napríklad pre druhú časť AOS – spracovanie aktuálnych (live) dát.

V prípade, kedy užívateľ žiada o vizualizáciu výsledkov uloženej analýzy sa postupuje tak, že užívateľ jednoducho zašle všetky potrebné inštrukcie do NinjaTraderu, ktorého stratégia sa v nich dokáže orientovať. Taktiež sa použije iná stratégia, ako v predchádzajúcom prípade. Postupne prechádza graf od začiatku časového rozsahu. Ak sa dostane na bod, ktorý je v inštrukciach prítomný (budú radené chronologicky, ale nie je to nutné), vyhodnotí, aký grafický objekt sa má vykresliť a bod si zapamätá, prípadne objekt vykreslí, ak má na to dostatok údajov.

4.4 Užívateľské rozhranie

Pri návrhu užívateľského rozhrania (UI) aplikácie je nutné analyzovať požiadavky na toto rozhranie, vykonať štúdium cieľovej skupiny, ktorá bude s aplikáciou pracovať, a na základe týchto dvoch krokov navrhnúť vhodné grafické užívateľské rozhranie (GUI). Obchodná platforma NinjaTrader už svoje užívateľské rozhranie obsahuje a je nemenné, preto nemá zmysel sa mu venovať a snažiť sa ho upravovať. Medzi hlavné požiadavky UI vyvíjanej aplikácie patrí to, že musí umožňovať jednoducho vytvárať postupnosť analytických modulov pre novú analýzu. Každý modul bude graficky zvlášť znázornený a prepojený s ďalšími modulmi. Ďalej musí obsahovať sekciu pre pridávanie komentárov a informácii. Tiež musí užívateľovi poskytnúť spôsob pre ukladanie/načítavanie súborov z disku.

Cieľová skupina, ktorá bude využívať UI vyvíjaného nástroja sú matematici, štatistici, obchodníci a programátori. Táto cieľová skupina je významná tým, že dokáže bez problémov pracovať s počítačom a rýchlo sa naučiť používať novú aplikáciu. Predpokladaný užívateľ je teda vzdelaný muž/žena stredného veku.

Na základe analýzy cieľovej skupiny a požiadaviek na UI bude rozhranie časti pre tvorbu postupnosti analytických modulov založené na princípe stromových dátových štruktúr. Každý modul má vstup a výstup, ktorý môže byť počiatočný, ukončujúci alebo pripojený na ďalší modul. Celý takýto orientovaný graf, zložený z jednotlivých uzlov, predstavuje tok dát, ktorý je postupne spracovávaný modulmi, začínajúci z počiatočného predpripraveného modulu značiaceho celé vstupné dáta, ktoré užívateľ vybral v NinjaTraderi a končiaci poslednými analytickými modulmi. Pri každom module má užívateľ možnosť nastaviť jeho parametre (časové údaje, hodnotu v pipoch, smer analýzy, atď.). Toto nastavenie je pri každom type modulu iné.

Kapitola 5

Implementácia

Po analýze požiadaviek a návrhu nástroja pre analýzu grafu prichádza na rad jeho implementácia. Aplikácia je vytvorená na platforme .NET v programovaciom jazyku C# [3]. Dôvod pre výber tejto kombinácie technológii je kompatibilita s jazykom NinjaScript, ktorý je nadstavbou nad jazykom C#. Keďže nie je potrebné zavádzať do systému prvok, ktorý prevádza informácie medzi dvomi rozličnými programovacími jazykmi, zvýši sa rýchlosť prenosu tržných dát medzi NinjaTraderom a aplikáciou.

V rámci implementácie je potrebné vytvoriť dve stratégie v jazyku NinjaScript, správu pre prenos dát medzi dvomi podsystémami, moduly jednotlivých komponentov aplikácie a päť analytických modulov, ktoré slúžia ako ukážka možností, ktoré nástroj ponúka.

V tejto kapitole sú postupne popísané stratégie, niektoré zaujímavé moduly aplikácie, ktoré tvoria základ tohoto nástroja a vytvorené analytické moduly s ukážkami ich grafického výstupu. Väčšina týchto prvkov je doplnená o ukážky kódu pre vizualizáciu konkrétnych programových štruktúr.

5.1 Komponent v obchodnej platforme

Všetky stratégie v NinjaTraderi sú umiestnené v jednom priestori mien a pozostávajú z triedy, ktorá povoluje poskytnúť implementáciu štyrom, pre systém dôležitým, metódam.

Prvou je metóda `Initialize`, ktorá sa podľa dokumentácie volá vždy raz pred spustením stratégie v grafe. Táto informácia je však mierne zavádzajúca, pretože metóda je volaná viackrát, a to pri načítaní zoznamu stratégii, pri výbere konkrétnej stratégie a pri jej spustení. Z tohto dôvodu nie je vhodné túto metódu využiť na vytvorenie prostriedkov pre prenos dát do aplikácie a je použitá iba pre nastavenie parametrov stratégií, ako napríklad: používanie agregovaných dát, maximálny objem tržných dát, ktoré stratégia spracuje a počet barov, ktoré sú potrebné pre začiatok vykonávania stratégie.

Druhá metóda má názov `OnStartUp` a je volaná iba raz po metóde Initialize, čiže tesne pred spustením spracovania tržných dát stratégiou. Práve táto metóda je vhodná pre inicializáciu prostriedkov na prenos dát.

Ďalšia metóda, `OnBarUpdate`, je volaná po každom jednom bare tržných dát (prípadne po každom ticku), teda poskytuje dáta pre istý časový úsek a možnosť vykresliť alebo vyznačiť tento bar.

Poslednou významnou metódou je `OnTermination` a je volaná raz, ihneď po odstránení danej stratégie z grafu. Je vhodná pre uvoľnenie zdrojov, ktoré stratégia využívala, prípadne niečo, čo je potrebné vykonať iba raz, a to po definitívnom ukončení stratégie.

5.1.1 Stratégia novej analýzy

Stratégia novej analýzy používa agregáciu tržných dát, ktoré prechádza v rozsahu grafu, na ktorý je stratégia zavolaná, a na svoju činnosť nepotrebuje žiadne predom načítané bary.

Pred začiatkom jej činnosti je inicializovaný objekt správy, ktorý je zodpovedný za serializáciu správy do binárneho formátu, a klientská časť mechanizmu nazývaného Named Pipe, ktorý slúži na prenos dát medzi časťami systému. Všetky tieto zdroje sú umiestnené ako statické členy vlastnej statickej triedy `Holder`, ktorá je súčasťou priestoru mien `Strategy`. Ďalej sa klientská časť Named Pipe pripojí na serverovú časť umiestnenú v aplikácii.

Pri prechádzaní tržných dát stratégia nerobí nič iné ako zber jednotlivých dát pre každý bar. Tieto informácie, ktoré definujú jeden bar sú: presný čas, volume, close, open, high a low a sú pridávané priamo do správy.

Pretože neexistuje vhodný spôsob ako dopredu určiť presný počet barov (prípadne tickov) v grafe, všetky zozbierané dáta sa serializujú a do aplikácie pošlú po odstránení tejto stratégie z grafu.

Nasledujúca ukážka kódu zobrazuje implementáciu metódy `OnBarUpdate`.

```
protected override void OnBarUpdate()
{
    //pridanie času, volume, close, open, high a low hodnôt
    Holder.messageToSend.dateVal.Add(Time[0].ToString());
    Holder.messageToSend.volumeVal.Add(Volume[0]);
    Holder.messageToSend.closeVal.Add(Close[0]);
    Holder.messageToSend.openVal.Add(Open[0]);
    Holder.messageToSend.highVal.Add(High[0]);
    Holder.messageToSend.lowVal.Add(Low[0]);
}
```

5.1.2 Stratégia vykreslenia výsledkov

Stratégia na vykreslovanie výsledkov má rovnaké počiatočné nastavenia ako vyššie popísaná. Rozdielom je využitie inej statickej triedy (`HolderD`) na prechovávanie prostriedkov na komunikáciu a využitie XML serializéru namiesto binárneho.

Pred začiatkom vykreslovania samotného je inicializovaný serializér, klientská časť Named Pipe, ktorá je pripájaná na serverovú časť, a do objektu správy sú deserializované vykreslovacie inštrukcie, ktoré sú z aplikácie poslané hneď po pripojení.

Čas každého baru je porovnávaný s časmi inštrukcií. Ak sa zhoduje, je vykonaná príslušná akcia, čiže buď priame vykreslenie objektu alebo napríklad zapamätanie určitej hodnoty apod. Ukážky vizualizácie implementovaných modulov sa nachádzajú v podkapitole 5.3.

Nasledujúca ukážka kódu zobrazuje vykreslovanie niektorých objektov v závislosti na danom príznaku v metóde `OnBarUpdate`.

```
...
else if (HolderD.messageToD.drawResult[i][k] == "DNTSTOP")
{
    BackColorAll = Color.DarkRed;
}
else if (HolderD.messageToD.drawResult[i][k] == "T")
```

```
{
    DrawDiamond(Time[0].ToString() + "time", true, 0,
                Low[0] - TickSize*25, Color.Silver);
}
else if (HolderD.messageToD.drawResult[i][k] == "G")
{
    DrawVerticalLine(Time[0].ToString() + "gap", 0,
                     Color.Teal, DashStyle.Dot, 4);
}
...
```

5.2 Kontrolný komponent vyvíjanej aplikácie

Najväčšou výzvou pri implementácii jednotlivých modulov aplikácie je správne odhadnutie rozloženia záťaže medzi tieto moduly. Musia byť ľahko upravitelné a navzájom nezávislé. Každý modul má na starosti jednu konkrétnu úlohu (riadenie, komunikácia, ukladanie, atď.).

Nasledujúce podkapitoly ponúkajú popis troch najvýznamnejších modulov aplikácie.

5.2.1 Data Transceiver

Hlavnou úlohou tohoto modulu je prenos dát medzi aplikáciou a obchodnou platformou NinjaTrader. Je reprezentovaný pomocou triedy, ktorá pri vytovorení nového objektu inicializuje v tomto objekte serverovú časť Named Pipe, XML serializér a binárny serializér.

Dôležitým aspektom tohto modulu je jednoduchá zmena, prípadne vylúčenie obchodnej platformy. V prípade jej nahradenia za iný zdroj tržných dát, napíklad súbory, stačí vymeniť serverovú časť Named Pipe za mechanizmus práce so súbormi, ktorý tento súbor načíta a v príslušnej metóde sa konvertuje do tvaru správy, ktorá sa využíva na komunikáciu.

Data Transceiver obsahuje dve hlavné metódy:

- `ReceiveMessage` je využívaná pre prijatie dát zo stratégie novej analýzy. Čaká na pripojenie klientskej časti Named Pipe a prijaté binárne dáta deserializuje do objektu správy. Ak všetko prebehlo v poriadku, odpojí klientskú časť.
- `SendMessage` sa používa na odoslanie vykreslovacích inštrukcií do stratégie na to určenej. Čaká na pripojenie a dáta serializované do formátu XML naraz zašle do NinjaTraderu. Po úspešnom odoslaní odpojí klientskú časť.

Ukážku týchto dvoch metód je možné vidieť na nasledujúcom príklade.

```
public void ReceiveMessage()
{
    //čaká na pripojenie, deserializuje prijatú správu a uloží ju
    serverStream.WaitForConnection();
    message = (MessageDataNT)formatter.Deserialize(serverStream);
    serverStream.Disconnect();
}

public void SendMessage()
```

```
{
    //čaká na pripojenie, serializuje správu a odošle ju
    serverStream.WaitForConnection();
    xmlSerializer.Serialize(serverStream, message);
    serverStream.Disconnect();
}
```

5.2.2 Historical analysis engine

Dôvodom zavedenia tohoto modulu je poskytnutie jedného rozhrania pre riadenie celej aplikácie. Vďaka tomuto modulu je veľmi jednoduché zmeniť užívateľské rozhranie, keďže sa na užívateľove vstupy dotazuje priamo on.

Engine je reprezentovaný pomocou triedy, ktorá pri vytvorení objektu inicializuje novú sekvenciu analytických modulov (ktorú užívateľ neskôr naplní) a tri ďalšie moduly aplikácie: Data Transceiver, Storage Controller a Modules Controller.

Implementácia niektorých metód v engine nie je nevyhnutná, ale poskytuje vhodný level abstrakcie a z ich postupného volania je na prvý pohľad jasné, ako v systéme prúdia dáta (tiež pri pohľade na obrázok 4.1 je jednoznačné, ktoré metódy vykonávajú aký presun a akciu).

Na nasledujúcej ukážke kódu sú zobrazené dve hlavné situácie: nová analýza a vykreslenie existujúcej analýzy. Menej zaujímavé časti týchto situácií sú vynechané a nahradené komentárom.

```
/***Nová Analýza***/
historicalAnalysisEngine.LoadDataFromNT();
historicalAnalysisEngine.PassDataToEngine();
historicalAnalysisEngine.PassDataToModController();
historicalAnalysisEngine.PassModSeqToModController();
historicalAnalysisEngine.Analyse();
historicalAnalysisEngine.PassResultToEngine();

//pridanie informácií k analýze a komentárov
//pridanie cesty k výslednému súboru do storage controlleru

historicalAnalysisEngine.PassResultToStorageController();
historicalAnalysisEngine.StoreAnalysis();

/***Vykreslenie Analýzy***/
//nastavenie cesty k súboru s výsledkami analýzy
historicalAnalysisEngine.LoadAnalysis();
historicalAnalysisEngine.PassDrawIToEngine();
historicalAnalysisEngine.PassDrawIToDataTransceiver();
historicalAnalysisEngine.SendDataToNT();
```

5.2.3 Storage controller

Tento modul je do systému zaradená ako prvok, ktorý sa stará o ukladanie výsledkov analýzy do CSV súborov, ich spätné načítanie, filtráciu a prevod na vykreslovacie inštrukcie, ktorým rozumie určená stratégia v NinjaTraderi.

Metódy, ktoré zabezpečujú ukladanie a načítanie analýz sú:

- `StoreAnalysis` - Metóda začína výpisom obecných informácií a komentárov k analýze do súboru, ktorý užívateľ určil. Nasleduje tisk hlavičky tržných dát a názvov jednotlivých analytických modulov, ktoré sa na analýze podielali. Ďalší riadok obsahuje prípadné príznaky ku každému an. modulu. V aktuálnej verzii je to iba príznak draw (je ale možné pridať ľubovoľné príznaky), ktorý značí, či majú byť výsledky daného modulu vykreslované, alebo boli určené len na export. Po úvodných výpisoch nasledujú samotné tržné dáta a výsledky modulov, ktoré sú v cykle vypísané priamo do súboru.
- `LoadAnalysis` - Metóda začína inicializáciou novej správy, ktorá v tomto prípade slúži na ukladanie času a príslušných výsledkov modulov a bude neskôr modulom Data Transceiver zaslaná stratégii určenej na vykreslovanie. Nasleduje ignorovanie prvých troch riadkov súboru, ktoré predstavujú obecné informácie a komentáre. Ďalej je zistený počet modulov, ktoré sa podielali na analýze, a pre každý je v správe inicializovaný nový zoznam. Nasleduje pole príznakov a dáta samotné, ktoré sú prechádzané po riadkoch. Ak tento riadok neobsahuje iba prázdne výsledky všetkých modulov (nič zaujímavé pre vykreslenie) a modul zaviedol príznak draw, je pridaný aj s časom do správy. Tento postup sa opakuje pre každý výsledok modulu na riadku a pre každý riadok súboru. Po prechode celým súborom obsahuje správa tzv. vykreslovacie inštrukcie a je pripravená na ďalšie spracovanie (odoslanie).

5.3 Analytické moduly

Súčasťou požiadaviek na implementáciu nástroja pre analýzu grafu bolo aj päť analytických modulov, ktorých význam nie je poskytnúť bezchybnú a stopercentne správnu technickú analýzu, ale ukázať, o aké typy modulov môže obchodník (užívateľ) systém rozširovať a čo prostredníctvom nich môže dokázať.

Analytický modul je reprezentovaný triedou, ktorá dedí z triedy `ModuleBase` položky, ktoré obsahuje každý jeden modul. Tieto sú: ukazateľ na iný modul, ktorého výsledok signalizuje, ktoré riadky tržných dát majú byť použité ako vstup do modulu, meno modulu, výsledok modulu, výskyt príznaku draw a číslo, ktoré určute prioritu tohto modulu pri vykonávaní analýzy (čím skôr sa modul v sekvencii nachádza, tým má vyššiu prioritu). Ďalšou dôležitou položkou triedy `ModuleBase` je abstraktná metóda `Analyse`, ktorej implementáciu musí povinne poskytnúť každý z modulov a ako parameter berie správu obsahujúcu všetky tržné dáta.

Nastavenie modulov sa vykonáva pomocou vlastností (properties), ktorých počet je ľubovoľný. Každá vlastnosť, ktorá je v triede modulu uvedená, musí slúžiť na konfiguráciu. Aktuálne podporované dátové typy vlastností sú: `string`, `boolean`, `integer` a `double`.

Samotné výsledky modulov sú reprezentované cez príznak (dátového typu `string`), ktorý môže signalizovať významnú udalosť, ktorá bola vyhodnotená, alebo môže obsahovať nejakú hodnotu. Formát týchto príznakov nie je nijak dopredu obmedzený, ale užívateľ musí dávať pozor, aby nepoužil jeden príznak viackrát, pretože pomocou nich NinjaTrader vykresluje určené objekty. Pre všetky moduly platí, že príznak signalizujúci žiadnu zmenu, nič nenájdené, neaktivitu alebo nezaujímavý riadok, má označenie `N`.

Nasledujúca ukážka kódu zobrazuje minimálny kód potrebný pre vytvorenie nového analytického modulu s ignorovaním bezvýznamných riadkov vstupu a inicializáciu jeho výsledku.

```
public class newMod : ModuleBase
{
    public int vlastnost { get; set; }
    public override void Analyse(MessageDataNT baseData)
    {
        //inicializácia výsledku
        moduleResult = new ModuleResult(moduleName, isDrawn,
                                        baseData.dateVal.Count);

        //prechod cez všetky tržné dáta
        for (int i = 0; i < baseData.dateVal.Count; i++)
        {
            //ignorovanie bezvýznamných riadkov vstupu
            if (input != null)
            {
                if (input.moduleResult.result[i] == "N")
                {
                    moduleResult.result[i] = "N";
                    continue;
                }
            }

            //vlastná logika modulu
        }
    }
}
```

5.3.1 Modul 1

`PipOverTimeMod` je modul, ktorý vyhľadáva zmeny v tržných dátach, počas určeného časového okna, ktoré sú rovné alebo väčšie ako špecifikovaný počet pipov.

Modul má niekoľko vlastností, pomocou ktorých je konfigurovaný:

- `pipRestriction` - Počet pipov (výška tržných dát), ktorý je použitý ako minimum pre vyhľadávané zmeny.
- `timeRestriction` - Počet barov (šírka tržných dát), ktoré sú použité ako časové okno pre vyhľadávanie zmien.
- `pipSize` - Veľkosť jedného pipu na trhu (väčšinou je táto hodnota rovná jednej desaťtisícine).

Práca tohto modulu spočíva v prechádzaní vstupných dát v dvoch cykloch. Prvý iteruje cez všetky bary a druhý kontroluje `N` nasledujúcich barov od baru aktuálneho, pričom `N` je dané vlastnosťou `timeRestriction`. V tomto rozsahu vyhľadáva minimum a maximum (konkrétne v hodnotách high a low) a ich rozdiel porovná s minimálnou veľkosťou dát, ktorá je daná súčinom vlastností `pipRestriction` a `pipSize`. Ak je toto porovnanie pravdivé, modul úspešne našiel požadovanú situáciu, index prvého cyklu sa posúva na jej posledný bar a do výsledku je zaznačený príslušný príznak. Po prvom priechode dát nasleduje priechod druhý, v ktorom sú vyplnené medzery vo výsledku príznakmi.

Príznaky, ktoré predstavujú výstup tohoto modulu sú:

- `U` - Bar je súčasťou rastúcej postupnosti, ktorá spĺňa vyššie spomenuté požiadavky na situáciu.
- `D` - Podobne ako pri predchádzajúcom príznaku, ale postupnosť je klesajúca.
- `US` - Bar samotný tvorí stúpajúcu postupnosť o dĺžke jedna, čiže situácia nastala počas tohto jediného časového úseku.
- `DS` - Podobne ako v prechádzajúcom prípade s rozdielom takým, že smer situácie je klesajúci.

Na obrázku 5.1 je zobrazená ukážka grafického výstupu tohto analytického modulu.

Obr. 5.1: Grafický výstup `PipOverTimeMod` modulu

5.3.2 Modul 2

`TimeMod` ako analytický modul vykonáva restrikciu tržných dát podľa určeného časového intervalu. Obmedzenie sa môže týkať rokov, mesiacov, dní, hodín alebo minút. Tento modul by mal slúžiť hlavne na označenie vstupných dát pre iné moduly, ale jeho výsledok môže byť taktiež graficky zobrazený.

Vlastnosti, pomocou ktorých je modul konfigurovateľný sú nižšie vymenované, pričom tie, ktoré nie sú využité, je potrebné pomocou GUI nastaviť na hodnotu –1. `X` značí postupne všetky prvky z množiny {year, month, day, hour, minute}.

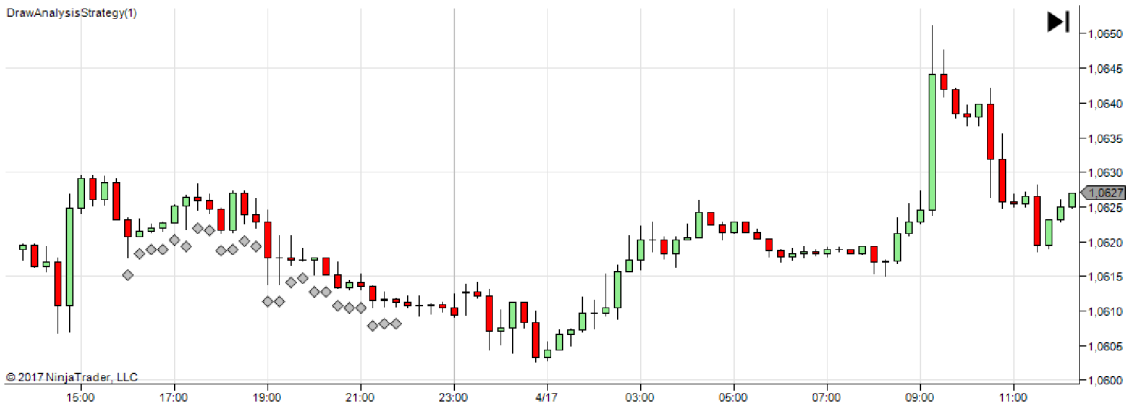
- `XRestrictionStart` - Číslo označujúce začiatok časového intervalu pre obmedzenie.
- `XRestrictionEnd` - Koniec tohoto intervalu. Tieto dve vlastnosti môžu nadobúdať iba hodnoty, ktoré podporuje daný typ obmedzenia (napr. minúty 0-59).

Činnosť tohoto modulu spočíva v prechádzaní vstupných dát v jednom cykle a konvertovaní času každého baru do formátu, ktorý umožňuje porovnávanie časových údajov. Každý časový údaj je porovnaný s príslušným časovým intervalom. Pokiaľ spĺňa toto obmedzenie, je posupne porovnávaný s ostatnými restrikcami, ktoré užívateľ špecifikoval.

Výstup modulu je daný pomocou jediného príznaku:

- `T` - Bar spĺňa všetky užívateľom dané časové obmedzenia.

Obrázok 5.2 slúži ako ukážka grafického výstupu modulu `TimeMod`.



Obr. 5.2: Grafický výstup `TimeMod` modulu

5.3.3 Modul 3

`SMAMod` je analytický modul, ktorý slúži ako ukážka toho, že nástroj je schopný ukladať výsledky aj v podobe číslených hodnôt a takýmto spôsobom definovať napríklad vlastné indikátory. SMA je skratka pre Simple Moving Average, čo v preklade znamená jednoduchý kĺzavý priemer.

Princíp výpočtu kĺzavých priemerov spočíva v zvolení časovej periódy, počas ktorej sú ceny priemerované. SMA prikladá všetkým časovým údajom zahrnutým do výpočtu rovnakú dôležitosť. Vzorec pre jeho výpočet zobrazuje rovnica 5.1, pričom n značí veľkosť periódy a pM jednotlivé ceny [6].

$$SMA = \frac{1}{n}\sum_{i=0}^{n-1} pM - i \tag{5.1}$$

Modul má jednu vlastnosť, pomocou ktorej je konfigurovateľný:

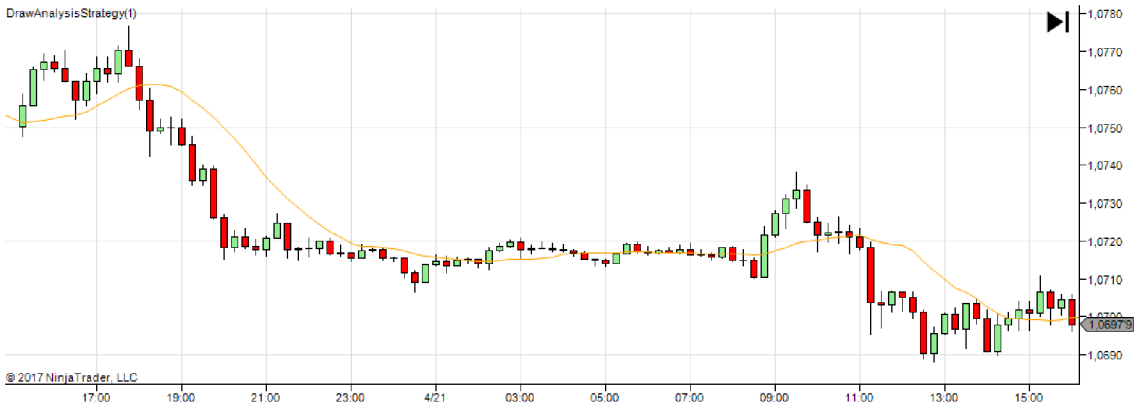
- `period` - Časová perióda, počas ktorej sú ceny priemerované.

Činnosť tohoto modulu priamo odpovedá vyššie uvedenému vzorcu, a teda pre každý bar sa vypočíta SMA za posledných `period` barov. Prvá perióda sa samozrejme ignoruje, pretože v nej nie je dostatok údajov na výpočet SMA.

Príznak predstavujúci výsledok analýzy týmto modulom má tvar:

- `SMA:$VALUE` - Určuje hodnotu SMA pre aktuálny bar, pričom `$VALUE` je samotná vypočítaná hodnota.

Na obrázku 5.3 sa nachádza ukážka grafického výstupu modulu.



Obr. 5.3: Grafický výstup `SMAMod` modulu

5.3.4 Modul 4

`GapMod` ako analytický modul vyhľadáva v tržných dátach tzv. „Gapy“. Gap je jednoducho povedané oblasť grafu, kde nedošlo pri danej cene (za určitý čas) k žiadnym obchodom.

Tento modul neobsahuje žiadne vlastnosti na jeho konfiguráciu, aj keď by bolo možné využiť napríklad minimálnu/maximálnu veľkosť gapu alebo obmedzenie časového intervalu, kedy ku gapu mohlo dôjsť.

V hlavnom cykle modul prechádza všetky tržné dáta a pre aktuálny a nasledujúci bar vypočíta skutočný časový rozdiel. Ten si uchová a postup opakuje pre ďalšie bary. Ak je rozdiel cien dvoch susedných barov odlišný od prechádzajúceho rozdielu, tieto dva bary tvoria gap.

Príznak, ktorý udáva úspech tohoto modulu je nasledujúci:

- `G` - Bar je súčasťou gapu, pričom gap je vždy tvorený minimálne dvomi barmi.

Obrázok 5.4 zobrazuje spôsob vykreslenia úspechu modulu `GapMod`.

Obr. 5.4: Grafický výstup `GapMod` modulu

5.3.5 Modul 5

`TrailingTrendMod` uzatvára päticu implementovaných analytických modulov. Identifikuje stúpajúci a klesajúci trend v tržných dátach. Keďže identifikácia trendu je veľmi rozsiahla kapitola a je vhodná pre rozsiahly výskum, tento modul považuje za koniec aktuálneho trendu moment, kedy rozdiel medzi posledným minimom/maximom a aktuálnou cenou je väčší ako špecifikovaný počet pipov. Týmto prístupom simuluje príkaz (mechanizmus) trailing stop, ktorý je veľmi využívaný a obľúbený medzi obchodníkmi. Výsledok je teda taký, že celý graf je rozdelený na časti stúpajúce a klesajúce, aj keď to nutne nemusí znamenať, že sa tak striedal trend.

Dve vlastnosti, pomocou ktorých sa vykonáva konfigurácia modulu sú:

- `maximumTrailingStop` - Maximálny počet pipov, ktorý je považovaný za trailing stop.
- `pipSize` - Veľkosť jedného pipu na danom trhu.

Modul začína analýzu na prvom bare vstupných dát, podľa orientácie ktorého určí, či začína stúpajúci alebo klesajúci trend a určí aktuálne minimum a maximum. Pre každý ďalší bar zisťuje, či nenastalo nové minimum alebo maximum a vypočíta rozdiel medzi poslednou aktuálnou cenou a posledným minimom/maximom. Pokiaľ je tento rozdiel prepočítaný na veľkosť v pipoch väčší ako hodnota vlastnosti `maximumTrailingStop`, nastáva zmena v smere trendu a aktuálny bar je označený ako koniec predchádzajúceho trendu.

Príznaky, ktoré modul produkuje sú nasledujúce:

- `UPT` - Bar je súčasťou stúpajúceho trendu.
- `UPTSTOP` - Bar značí ukončenie aktuálneho stúpajúceho trendu a prechod na klesajúci.
- `DNT` - Bar je súčasťou klesajúceho trendu.
- `DNTSTOP` - Bar označuje ukončenie klesajúceho trendu a zmenu na stúpajúci trend.

Na obrázku 5.5 je znázornený grafický výstup modulu na grafe v NinjaTraderi.

Obr. 5.5: Grafický výstup `TrailingTrendMod` modulu

Kapitola 6

Testovanie

Po implementácii samotného programu nasleduje jeho testovanie. Každý z implementovaných modulov aplikácie bol otestovaný samostatne ako časť a nasledovalo systémové testovanie, čiže všetkých komponentov ako celku.

Zaujímavejšie je však testovanie, ktoré má zmysel pre obchodníka, čiže overovanie nejakej hypotézy, ktorú obchodník vymyslel, a myslí si, že by v konečnom dôsledku mohla byť stratégia na nej založená zisková.

Ak je potencionálna stratégia založená na technickej analýze, teda predpokladá sa, že všetky informácie, ktoré majú zmysel pre predikciu vývoja smeru trhu sa už prejavili na grafe, je vhodné takúto analýzu vykonať a jej výsledky vizualizovať. Keď má obchodník k dispozícii graf s vyznačenými údajmi, dôležitými pre jeho hypotézu, sleduje, v akých miestach vývoj ceny odpovedá jeho očakávaniam a kedy by naopak obchodná stratégia zlyhala. Po opakovanom upravovaní parametrov analýzy, sledovaní a zhodnocovaní výsledkov je možné vyrobiť obchodnú stratégiu, ktorá je profitabilná.

V tejto kapitole je zobrazený pohľad na dve prípadové štúdie, teda rozobratie dvoch hypotéz. Po ich formulácii nasleduje predstavenie postupností analytických modulov, ktorých grafický výstup je vhodný na pozorovanie. Výber týchto modulov je však obmedzený na päť implementovaných. Posledným krokom je zhodnotenie výsledkov a ukážka grafického výstupu. Nejdená sa o celkový rozbor potenciálnej obchodnej stratégie a vytváranie štatistík, ale o ukážku práce vyvíjaného nástroja na základe nejakého nápadu.

6.1 Prípadová štúdia 1

Jednou z najznámejších obchodných stratégií je stratégia založená na SMA crossover. Ako už názov napovedá, jedná sa o prekríženie dvoch jednoduchých kĺzavých priemerov. Teória je taká, že na graf sú zavedené dva kĺzavé priemery s rozličnou periódou a obchodník sleduje ich prekríženie. Ak SMA s menším číslom periódy (rýchlejší) prekríži druhý SMA (pomalší) a rýchlejší pri tomto strete stúpa (prekríženie zdola nahor) je to signál pre nákup. Pri prekrížení v opačnom smere sa jedná o signál pre predaj. Toto je iba rýchly pohlad na tento mechanizmus a viac je možné dohľadať napríklad v [2] v sekcii o stratégiach.

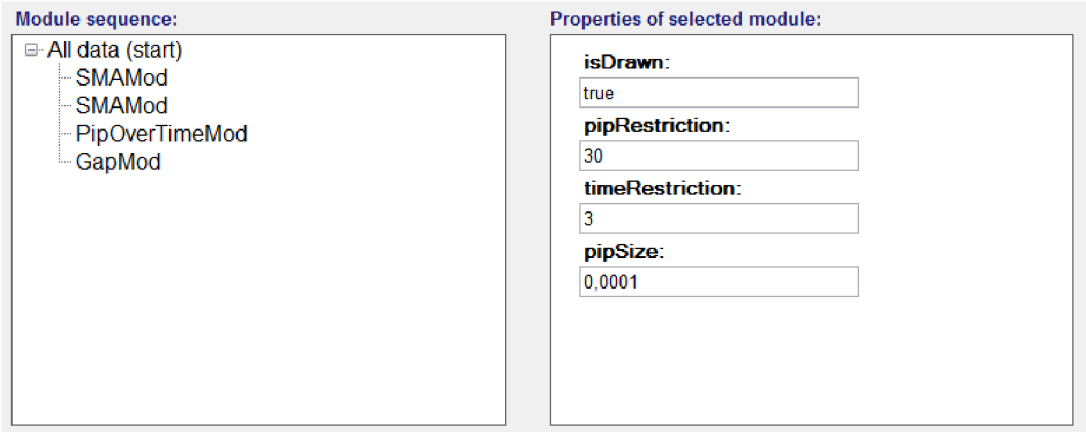
Vo veľa prípadoch je SMA crossover považovaný za príklad stratégie, ktorá nie je ani zisková, ani strátová, teda po opakovaných nákupoch a predajoch by sa mal stav účtu držať na približne rovnakej hladine. Z tohoto dôvodu sa obchodníci snažia túto stratégiu vylepšovať o nové prvky, ktoré by ju premenili na aspoň trochu ziskovú. Táto prípadová štúdia simuluje takúto snahu obchodníka o vylepšenie. Do grafu budú znázornené miesta,

ktoré predstavujú relatívne prudký nárast alebo pokles ceny. Predpoklad je, že ak nastane prudký nárast ceny v mieste, kde podľa SMA crossover stratégie má cena stúpať, trh bude mať tendenciu čo najskôr tento prudký nárast vyrovnať a nastane pokles ceny. Príkaz vstupu by sa teda vykonal ihneď po dosiahnutí prudkého nárastu a výstup by nastal pri najbližšom prekrížení SMA, prípadne ešte nižšie so zavedením vhodného trailing stop loss príkazu. Podobne by to bolo aj pre opačný prípad, teda s prudkým poklesom ceny.

6.1.1 Postupnosť analytických modulov

Základom pre túto jednoduchú hypotézu sú dva moduly `SMAMod`, každý s rozdielnou periódou. Tento test bude prebiehať na pätnásťminútovom grafe, periódy modulov budú zvolené experimentálne na jedenásť a tridsaťštyri. Pre zobrazenie prudkých nárastov a poklesov ceny bude využitý modul `PipOverTimeMod`, ktorý bude konfigurovaný na vyhľadávanie situácií, kde v priebehu troch barov nastala zmena ceny o viac ako tridsať pipov, čo je na pätnásťminútovom grafe relatívne prudký nárast/pokles. Posledným zavedeným modulom bude `GapMod`, ktorý vyhľadáva na trhu gapy. Tento modul bude slúžiť ako varovanie pred obchodovaním v blízkosti nájdených situácií, pretože SMA v týchto miestach nemá dostatok údajov na správny výpočet a môže zaznamenávať veľmi prudké nárasty/poklesy.

Na obrázku 6.1 je zobrazený pohľad na vyššie spomínané nastavenia priamo v aplikácii.



Obr. 6.1: Nastavenie postupnosti analytických modulov

6.1.2 Zhodnotenie výsledkov

Po prechádzaní grafického výstupu analýzy obchodník zistí, či má jeho navrhnutá stratégia zmysel, spočíta pomer úspešných situácii k neúspešným, počet získaných pipov k strateným. Ak má na prvý pohľad stratégia zmysel, je vhodné ju ďalej študovať, vytvoriť rozsiahlu štatistiku a vymyslieť ďalšie podmienky, ktoré by zaistili jej profitabilitu.

Tento test bol vykonávaný na pätnásťminútovom grafe za obdobie päťdesiatich dní. Očakávane nastali situácie, kedy bola stratégia úspešná a situácie, kedy bola neúspešná. Ako bod vstupu do obchodu bola zvolená presná hranica tridsať pipov prudkého nárastu/poklesu a výstupný bod bola vždy open cena nasledujúceho baru po prekrížení dvoch zavedených SMA. Vždy bol aktívny (prebiehajúci) maximálne jeden obchod, v tesnej blízkosti gapov obchody neboli spúšťané a zvolený trh bol EUR/USD. Nasledujúce výsledky by sa určite

dali vylepšiť (zmierniť straty) zavedením vhodných príkazov trailing stop loss, prípadne take profit.

- **Počet všetkých obchodov:** 19
- **Počet ziskových obchodov:** 12
- **Počet strátových obchodov:** 7
- **Celkový počet získaných pipov:** 255.7
- **Celkový počet stratených pipov:** 143.8
- **Výsledný pomer získaných pipov ku strateným pipom:** 1.78:1

Uvedené výsledky napriek jednoduchosti vstupných a výstupných pravidiel preukazujú, že danú stratégiu je vhodné študovať a venovať sa jej.

Obrázok 6.2 zobrazuje ukážku grafického výstupu analýzy. Pre zvýšenie prehľadnosti bol SMA s periódou 34 prekrytý modrou farbou. Šipky znázorňujú jednotlivé obchody, ktoré boli zmerané nástrojom pravítko (ruler).

Obr. 6.2: Grafický výstup analýzy

6.2 Prípadová štúdia 2

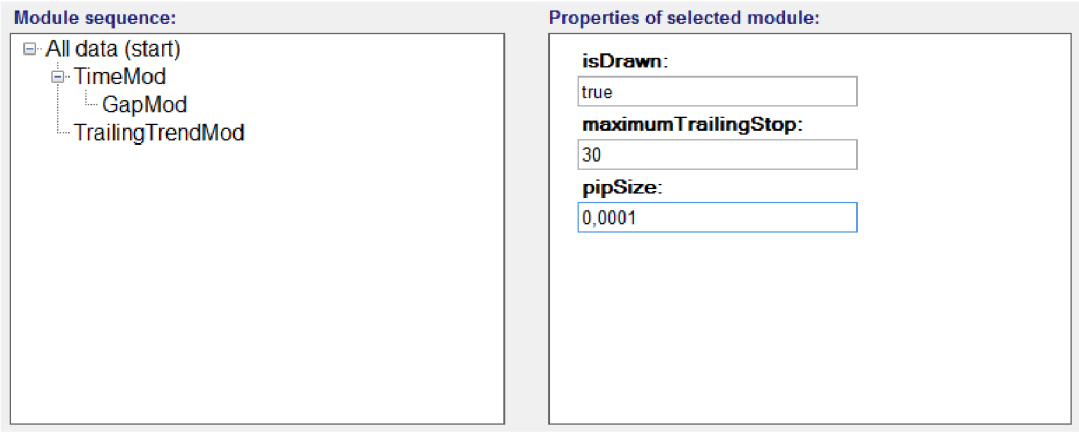
Medzi ďalšiu populárnu stratégiu na forexovom trhu patrí obchodovanie nedeľných gapov. Obchodníci sa snažia vymyslieť spôsob, ako zistiť, aká bude cena po obnovení možnosti obchodovať po víkende. Svojim charakterom tieto stratégie pripomínajú princíp binárnych opcií. Vyhodnotenie tejto ceny je zložité a často súvisí s fundamentálnou analýzou.

Tento test simuluje snahu obchodníka o vytvorenie novej stratégie založenej na viac-menej náhodnom nápade. Teória je taká, že gap sa bude obchodovať v smere aktuálneho trendu. Za moment vstúpenia do obchodu sa bude mierne nepresne považovať sekunda pred ukončením obchodovania a za vystúpenie z obchodu sa bude považovať sekunda po obnovení obchodovania.

6.2.1 Postupnosť analytických modulov

Na skúmanie tohoto nápadu budú stačiť tri analytické moduly. Prvým modulom je `TimeMod`, ktorý je konfigurovaný na obmedzovanie času na interval od dvadsiatej druhej hodiny do druhej hodiny. Grafický výstup tohoto modulu nie je potrebný, preto bude tak aj konfigurovaný. Na jeho výstup bude pripojený `GapMod`, ktorý bude vyhľadávať gapy. Táto postupnosť modulov zabezpečí, že budú vyhľadané iba gapy, ktoré nastali v spomenutom časovom intervale. Ďalším modulom bude `TrailingTrendMod`, ktorý bude simulovať detekciu stúpajúceho a klesajúceho trendu, ktorý sa otočí vždy v prípade, keď rozdiel medzi posledným minimom/maximom a niektorou nasledujúcou cenou bude väčší ako tridsať pipov.

Na obrázku 6.3 je zobrazený pohľad na vyššie spomínané nastavenia priamo v aplikácii.



Obr. 6.3: Nastavenie postupnosti analytických modulov

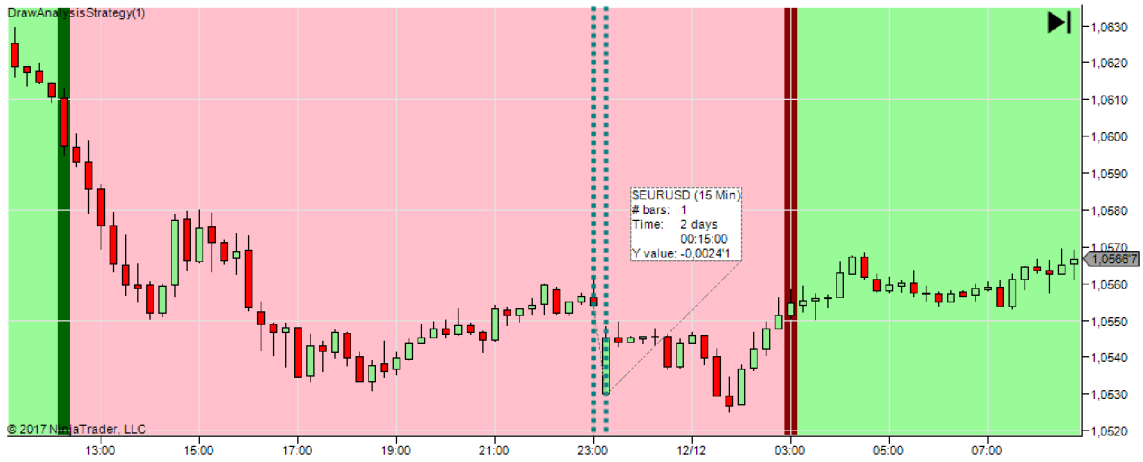
6.2.2 Zhodnotenie výsledkov

Test bol vykonávaný na pätnásťminútovom grafe za obdobie stopäťdesiat dní. Vždy bol aktívny maximálne jeden obchod a zvolený forexový trh bol EUR/USD. Keďže obchody vykonané v rámci tejto stratégie trvajú teoreticky dve sekundy, jej straty nie je možné zmierniť použitím podporných obchodných príkazov. Následujúce výsledky boli získané pozorovaním grafického výstupu analýzy.

- **Počet všetkých obchodov:** 22
- **Počet ziskových obchodov:** 12
- **Počet strátových obchodov:** 9
- **Celkový počet získaných pipov:** 412.4
- **Celkový počet stratených pipov:** 272,6
- **Výsledný pomer získaných pipov ku strateným pipom:** 1.51:1

Je možné, že výsledky tejto stratégie boli značne ovplyvnené náhodou. Na jej presnejšie otestovanie by bolo potrebné zaviesť modul na identifikáciu trendu, ktorého základ by tvoril veľmi zložitý algoritmus.

Obrázok 6.4 zobrazuje grafický výstup analýzy doplnený o zmeranie obchodov nástrojom ruler.



Obr. 6.4: Grafický výstup analýzy

Kapitola 7

Záver

Technická analýza je súčasťou obchodného plánu mnohých obchodníkov, a preto hľadajú nástroje, ktoré im umožnia analýzu vykonávať rýchlejšie a presnejšie. Kvalitných počítačových programov, ktoré toto dokážu je nedostatok, a preto je veľakrát technická analýza vykonávaná manuálne. Tento prístup je nepresný, zdĺhavý, náchylný k ľudskej chybe a mnohokrát nie je obchodníkovi umožnené si ťažko získané výsledky a nákresy uložiť.

Z týchto dôvodov som ako hlavný cieľ tejto práce zvolil vytvorenie nástroja, ktorý by vyššie uvedené problémy čiastočne vyriešil a bol by ľahko používateľný. Pri jeho vývoji bolo nutné naštudovať koncepty finančných trhov a obchodovania, zoznámiť sa s obchodnou platformou NinjaTrader, analyzovať všetky požiadavky na tento systému, vytvoriť jeho detailný návrh, nástroj implementovať a otestovať jeho funkčnosť. Súčasťou implementácie bolo aj vytvorenie piatich analytických modulov, pomocou ktorých boli otestované dve hypotézy (potencionálne obchodné stratégie).

Výsledný nástroj dokáže komunikovať s obchodnou platformou NinjaTrader, umožňuje užívateľovi vytvoriť viacero postupností analytických modulov, pomocou ktorých vykonáva analýzu tržných dát, výsledky analýzy ukladá a dokáže ich zobraziť na grafe v NinjaTraderi. Zdrojový kód aplikácie je ľahko modifikovateľný a udržiavateľný vďaka architektúre MVC a vzájomnej nezávislosti jednotlivých komponentov.

Okrem pridávania ďalších analytických modulov a optimalizácie existujúcich je aj na aplikácii samotnej čo pridávať a vylepšovať. Predmetom ďalšieho vývoja tohoto projektu by mohlo byť pridanie prepojenia aplikácie s ďalšími známymi obchodnými platformami ako je MetaTrader alebo TradeStation. Ďalším veľkým rozšírením by bolo načítanie tržných dát priamo zo súborov a vytvorenie vlastného vykreslovacieho jadra, ktoré by dokázalo zobrazovať výsledky analýz priamo v aplikácii.

Literatúra

[1] *FXCM: Forex Trading.* [Online], [cit. 2017-05-7].
URL https://www.fxcm.com

[2] *Moving Averages: Strategies.* [Online], [cit. 2017-05-8].
URL http://www.investopedia.com/university/movingaverage/

[3] *.NET Documentation.* [Online], [cit. 2017-05-7].
URL https://docs.microsoft.com/en-us/dotnet/

[4] *NinjaTrader: Trading Software and Brokerage.* [Online], [cit. 2017-05-7].
URL https://ninjatrader.com

[5] Arlt, J.; Arltová, M.: *Ekonomické časové řady.* Professional Publishing, 2009, ISBN 978-80-86946-85-6.

[6] Edwards, R. D.; Bassetti, W.; Magee, J.: *Technical Analysis of Stock Trends.* Taylor & Francis Inc, 2013, ISBN 978-1439898185.

[7] Ižip, R.: *Forexové opcie.* TRIM Broker, a.s., 2014, ISBN 978-80-971568-0-0.

[8] Polouček, S.: *Peniaze, banky, finančné trhy.* Wolters Kluwer (Iura Edition), 2010, ISBN 978-80-8078-305-1.

[9] Štýbr, D.; Klepetko, P.; Ondráčková, P.: *Začínáme investovat a obchodovat na kapitálových trzích.* Grada, 2011, ISBN 978-80-247-3648-8.

[10] Wikipedia: *Burza.* [Online], [rev. 2017-05-4], [cit. 2017-05-7].
URL https://cs.wikipedia.org/wiki/Burza

[11] Wikipedia: *Comma-separated values.* [Online], [rev. 2015-06-19], [cit. 2017-05-7].
URL https://sk.wikipedia.org/wiki/Comma-separated_values

[12] Wikipedia: *Exchange-traded fund.* [Online], [rev. 2017-04-19], [cit. 2017-05-7].
URL https://en.wikipedia.org/wiki/Exchange-traded_fund

[13] Wikipedia: *Stock market index.* [Online], [rev. 2017-03-24], [cit. 2017-05-7].
URL https://en.wikipedia.org/wiki/Stock_market_index

Prílohy

Príloha A

Obsah CD

K tejto práci je priložené CD s touto adresárovou štruktúrou:

- `install/` - Inštalátor aplikácie.
- `doc/` - Príručky na vytvorenie nového analytického modulu, inštaláciu, používanie aplikácie a pdf súbor tejto práce.
- `NTinclude/` - Zdrojové súbory určené na kopírovanie do NinjaTraderu.
- `latex/` - Zdrojové súbory tohoto dokumentu.
- `src/` - Zdrojové súbory aplikácie spolu s projektom do Microsoft Visual Studia.
- `caseStudies/` - CSV súbory ku dvom popísaným prípadovým štúdiám.