tgm
Die Schule der Technik

DIPLOMARBEIT

NOCTUA

Wertpapierhandelsalgorithmus mit Marktzustandsadaption

Ausgeführt in Zuge der Reife und Diplomprüfung
Ausbildungszweig Systemtechnik/Medientechnik

unter der Leitung von
Prof. Mag. Hans Brabenetz
Prof. Dr. Helmut Vana
Abteilung für Informationstechnologie

eingereicht am Technologischen Gewerbemuseum Wien
Höhere Technische Lehr- und Versuchsanstalt
Wexstrasse 19-23, A-1200 Wien

von
Peer Nagy 5CHITI
Gabriel Pawlowsky, 5BHITS
Josef Sochovsky, 5BHITS

Wien, im Oktober 2012

Abteilungsvorständin:	Prof. Dipl.-Ing. Grete Kugler
Tag der Reifeprüfung:	xx. xx xxxx
Prüfungsvorsitzender:	Univ.–Prof. Dipl.-Ing. Dr.techn. xxx
Erster Gutachter:	Dipl.-Ing.(FH) Mag. Dr.techn. Gotti Koppi
Zweiter Gutachter:	Prof. Dr.techn. Wenn Vorhanden

Vorwort

Diese Arbeit wurde im Jahr 2012 im Zuge unserer Ausbildung in der Abteilung für Informationstechnologie am Technologischem Gewerbemuseum (TGM), HTBLVA Wien 20, durchgeführt.

Dankesworte

Wien, im Oktober 2012 Name, Name, Name, Name

Abstract

This is the english abstract.

Kurzfassung

Deutsche Kurzfassung kommt hierher

Inhaltsverzeichnis

Abbildungsverzeichnis

Listings

KAPITEL 1

Einführung

KAPITEL 2

Voruntersuchung

> "The improvement of **understanding** is for two ends: first, our own increase of knowledge; secondly, *to enable us to deliver* that knowledge to others."
>
> (John Locke)

2.1 IST-Erhebung

Der Wertpapierhandel erfährt eine zunehmende Professionalisierung, was dazu führt dass sowohl professionelle Trader in großen Unternehmen, als auch semi-professionelle mit immer größeren Datenmengen, mehr Informationen und durch schnellere Märkte gleichzeitig kürzeren Entscheidungdauern konfrontiert werden. Um dem Trend zu programmatischen Trading-Lösungen entgegenzukommen gibt es eine Reihe von Datenanbietern, die Kursdaten in unterschiedlichen Zeitabständen und mit unterschiedlicher Aktualität anbieten. Je aktueller und öfter ein Tick, also ein Kursstand zu einem Zeitpunkt, desto teurer ist die Datenanbindung im Durchschnitt. Einer der führenden Echtzeit-Marktdaten Anbieter ist eSignal [1], die eine Selektion an Produkten zu verschiedenen Konditionen anbieten. [2] Die Version der Software ëSignal OnDemand"bietet die Möglichkeit Intraday-Kursdaten zu günstigeren Konditionen 15 Minuten verzögert abzurufen. Für einen Produktivbetrieb ist diese Verzögerung vermutlich zu lange, aber sicherlich hinreichend für den Entwicklungs- und Testbetrieb.
Die Anbindung des Datenproviders kann über Dynamic Data Exchange (DDE)

[1] http://www.esignal.com/default.aspx?tc=
[2] Preise s. http://www.esignal.com/esignal/pricing.aspx?tc=

erfolgen.

Um eine individuelle Trading-Strategie zu verwirklichen werden oft Online-Broker verwendet, über die resultierende Orders direkt mit vergleichsweise niedrigen Spesen abgewickelt werden können und algorithmisches Trading überhaupt erst praktisch möglich wird. Interactive Brokers (IB) [3] ist ein erwähnenswertes Beispiel eines solchen. Für IB ist bereits ein Test-Account vorhanden, über den Transaktionen virtuell durchgeführt werden können.
Im Bereich des algorithmischen Tradings sind bei high-performance Anwendungen C, C++ und deren Derivate sehr beliebt, bei Anwendungen, die nicht innerhalb von Sekundenbruchteilen operieren wird die .NET Umgebung unter Windows häufig verwendet.
Hinsichtlich stehen innerhalb des Projektteams mehrere Arbeitsplätze mit .NET-Umgebung zur Verfügung.

2.2 IST-Zustand

Wie bereits erwähnt wurde, ist das algorithmische Handeln eine riesige internationale Industrie. Die meisten Banken haben in der Investment-Abteilung Projekte im Bereich der Entwicklung von Algorithmen zum Wertpapierhandel. Gut funktionierende Algorithmen werden in der Regel aber geheim gehalten, um sich einerseits nicht in die Karten blicken zu lassen und den institutionellen Gewinn zu optimieren. Sollte die Handelsstrategie publik werden, wäre diese leicht auszunutzen und Schwachstellen zu finden.
Große Institutionen, wie Banken sind aber längst nicht die einzigen, die solche Programme nutzen und entwickeln. Mit der zur Verfügung stehenden Technologie kann heutzutage jeder mit Programmierkenntnissen und dem nötigen finanzwirtschaftlichen Wissen Trading-Software entwickeln. Das führt dazu, dass Fonds und selbst private Investoren diese Schiene des Investment nutzen.

Neben der proprietären Software gibt es auch einige wenige Anbiter, die teilweise Handelsentscheidungen im Abo verkaufen, Managed-Account Lösungen, denen der Zugriff auf das Investment-Portfolio gestattet wird und sogar Integrated Development Environments (IDEs), die eine Broker-Schnittstelle zur Verfügung stellen und wo der verwendete Algorithmus selbst programmiert werden kann.

2.3 SOLL-Zustand

[3] http://www.interactivebrokers.com/ibg/main.php

KAPITEL 3

Produktfunktionen

/LF10/
Charts darstellen

Beschreibung	Charts zur Darstellung der Algorithmen werden auf der Website angezeigt. Dabei muss besonders auf die Schnittstelle zur bereits bestehenden Software geachtet werden. Diese kann zum Beispiel über Webservices oder Datenbanken implementiert werden.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF20/
Account-Management bereitstellen

Beschreibung	Über zumindest einen primären Account kann sich ein Benutzer anmelden, um alle sämtliche Controlling-Funktionen der Webseite durchführen zu können.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF30/
News anzeigen

Beschreibung	Wenn möglich können Nachrichten-Headlines, die gehandelte Aktien beeinflussen auf der Webseite angezeigt werden. Ein Beispiel wäre der Bloomberg-Newsfeed, der auch vom Programm benutzt werden könnte.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF40/
Website-Software-Schnittstelle implementieren

Beschreibung	Um die Software von der Website aus kontrollieren zu können wird eine Schnittstelle benötigt, diese kann zum Beispiel über Datenbanken, Streams oder Webservices realisiert werden.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF41/
Investitionskapital einstellen

Beschreibung	Die Höhe des Kapitals, das zur Investition freigegeben ist, kann je nach Aktie eingestellt werden.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF42/
Handeln starten

Beschreibung	Das Handeln des Programms soll mittels einer Usereingabe gestartet werden können.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF43/
Handeln stoppen

Beschreibung	Das Handeln der Software soll mittels einer Usereingabe gestoppt werden können.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF44/
Entscheidungsmodus ändern

Beschreibung	Der User muss zwischen einem automatischen und einem manuellen Modus auswählen können; bei dem automatischen Modus soll die Software automatisiert entscheiden und handeln; bei dem manuellen Modus wird der User mittels einer Benachrichtigung (siehe /LF160/) zum Entscheiden aufgefordert.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF45/
Benachrichtigungen verschicken

Beschreibung	Dem User wird zum Beispiel mittels SMS, E-Mail oder Push-Benachrichtigung der Handlungsvorschlag zugesandt.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF46/
Aktie auswählen

Beschreibung	Der User kann aus einer prädefinierten Liste die gewünschten Aktien zum Handeln selektieren.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF110/
Input speichern

Beschreibung	Die vom Datenprovider ausgesendeten Daten (Bars) sollen gespeichert werden, um später weiter verarbeitet werden zu können.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF120/
Bars weiterleiten

Beschreibung	Der bisherige Kurs in Form von Bars wird an den Rechenkern übergeben.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF130/
Entscheidung berechnen

Beschreibung	Der Rechenkern soll aufgrund des Entscheidungsalgorithmus berechnen, wie das Wertpapier am besten zu behandeln ist (buy, sell, hold).
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF140/
Entscheidung ausgeben

Beschreibung	Die berechnete Entscheidung wird aus dem Rechenkern in den Verarbeitungskern geleitet, wo sie einerseits in der Konsole ausgegeben und andererseits ausgeführt werden.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF150/
Wertpapier kaufen

Beschreibung	Das System kauft über den Online-Broker-Account eine bestimmte Anzahl an Wertpapieren.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF160/
Wertpapier verkaufen

Beschreibung	Das System verkauft über den Online-Broker-Account eine bestimmte Anzahl an Wertpapieren.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF170/
Log erstellen

Beschreibung	Das System legt beim Programmstart lokal ein Log-File an, indem alle Entscheidungen versehen mit Datum und Uhrzeit während dem Programmlauf eingetragen werden.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF180/
Kursdaten beziehen

Beschreibung	Die Software muss eine Schnittstelle zum Beziehen von sowohl aktuellen als auch historischen Bars besitzen. Diese kann zum Beispiel über e-Signal oder Interactive Brokers realisiert werden.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF190/
Datenbankverbindung bereitstellen

Beschreibung	Das Programm muss eine Datenbankschnittstelle erhalten und die berechneten Daten für die Website erreichbar machen.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF200/
Laufzeitparameter ändern

Beschreibung	Um die Controlling-Funktionen zu ermöglichen muss das Programm genauer parametrisierbar sein. Es sollen zum Beispiel die Einstellungen auch während der Laufzeit verändert werden können.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

/LF210/
Multi-Threading ermöglichen

Beschreibung	Die Software muss Multi-Threading unterstützen, um die Entscheidung über automatisches Kaufen und Verkaufen von verschiedenen Aktien gleichzeitig zu treffen.
Aufwand	
Nutzen	
Ziel	
Vorbedingungen	-
Nachbedingungen	-

KAPITEL 4

Technische Machbarkeit

4.1 Variantenbildung

Die Backtesting-Software (BTS) kann man in jeder erdenklichen Programmiersprache schreiben, allerdings ist es wichtig daran zu denken, dass das Programm einerseits effizient arbeiten soll und deswegen hardwarenahe rechnet, und andererseits hat das Projektteam mit manchen Programmiersprachen keinerlei Erfahrung.
Die allgemeine Funktionalität muss das lesen und schreiben von Dateien sein, aber auch das algorithmische Rechnen soll effizient funktionieren. Für das Team kommen daher 3 Möglichkeiten in Frage: Eine Lösung in reinem C++, welches sehr hardwarenahe arbeitet, eine Mischung aus F# und C#, mit der eine parallelisierte Berechnung möglich wäre, und eine Java-Lösung, bei der das Team die größte Erfahrung mitbringt.
Bei der Kombination agiert C# als Handlungs- und Steuerkern und F# als funktionale Programmiersprache, als Rechenkern und Mastermindder Applikation, welches die Entscheidungen trifft. Hierbei wird einerseits eine enorm hohe Arbeitsgeschwindigkeit ermöglicht, da die beiden Sprachen relativ hardwarenah agieren und andererseits besteht der nicht zu unterschätzende Vorteil bzw. die Möglichkeit, den Rechenkern auf ein externes System outzusourcen, welches zum Beispiel enorme Rechenkapazitäten aufweisen könnte und somit viel komplexere und effizientere Algorithmen in annehmbarer Zeit durchrechnen und abhängig davon mehr gewinnbringende Entscheidungen treffen könnte. Dabei sollte es auch bei späteren Erweiterungen des Programms zu keinem signifikanten Geschwindigkeitsabfall kommen.

		Gewichtung	**C++ R*G**		**Java R*G**		**C/F R*G**	
Einfachheit	Aufwand Coding	10%	3	30	1	10	2	20
	Bedienung/ Wartung	6%	3	9	2	6	1	3
	Update	3%	3	9	2	6	1	3
	Integration	5%	3	15	2	10	1	5
	Kenntnisse	6%	3	18	1	6	2	12
	Gesamt	30%	3	90	2	44	1	46
Leistung	Übertragungszeit	6%	1	6	3	18	2	12
	Absturzsicherheit	5%	1	5	2	10	3	15
	Ressourcenverbrauch	3%	1	3	3	9	2	6
	Datenumfang	1%	1	1	3	3	2	2
	Gesamt	15%	1	15	3	50	2	35
Kosten	Lizenzen	10%	1	10	1	10	1	10
	Support	5%	3	15	1	5	2	10
	Betriebskosten	5%	1	5	1	5	1	5
	Dokumentation	5%	1	5	2	20	3	15
	Gesamt	15%	1	30	1	40	2	40
Dokumentation	Verfügbarkeit	10%	3	30	2	20	1	10
	Vollständigkeit	10%	3	30	2	20	1	10
	Qualität	10%	2	20	1	10	1	10
	Gesamt	30%	3	80	2	50	1	30

Kapitel	Gewichtung	**C++**		**Java**		**C#/F#**	
Einfachheit	30%	3	90	2	44	1	46
Leistung	15%	1	15	3	50	2	35
Kosten	15%	2	30	2	40	1	40
Dokumentation	30%	3	80	2	50	1	30

Gesamtbewertung			
Endreihung	3	2	1

Aus der Nutzwertanalyse kann man entnehmen, dass die C#/F# Kombination als die favorisierte Möglichkeit ausgeht, weitere Vorteile die sich aus der Wahl

dieser Mischung ergeben sind: gute Kenntnisse der Programmiersprachen, tolle Community und die Einfachheit, sowie die Erweiterbarkeit. Bei dieser Lösung wird die Steuereinheit vom C# Teil des Programms übernommen und die Rechenaufgaben werden von dem F# Teil bearbeitet. Außerdem ist das .net-Framework sehr beliebt, deswegen kann man damit rechnen das bei einem Problem genügend Helfer gefunden werden können.

KAPITEL 5

Wirtschaftliche Machbarkeit

5.1 Risikoanalyse

5.1.1 Personenausfall

Eintrittswahrscheinlichkeit: gering
Auswirkungen: gering

In dem unerwarteten Fall, dass ein Teammitglied längerfristig ausfällt, muss es möglich sein die Arbeitsaufgaben dementsprechend neu aufteilen zu können. Folgende Fälle könnten auftreten:

- Streit im Team
- Ausfall durch Krankheit oder Tod eines Teammitglieds
- Austritt eines Teammitglieds aus dem Projekt
- Der Auftraggeber könnte aufgrud von Unklarheiten den Projektabbruch initiieren
- Es kann passieren, dass von Seite des Auftragsgebers plötzlich kein Interesse an der Umsetzung des Produktes mehr gegeben ist, und es dadurch zu extremem Zeitverzug kommt, was bis zum Abbruch führen kann

Folgende präventive Maßnahmen werden eingeführt:

- Regeln für den Umgang innerhalb des Projekts

- Ausreichendes Interesse jedes Mitglieds und keine leistungstechnische Probleme
- Gutes Verhältnis mit den Auftraggebern

5.1.2 Zetiliche Risiken

Eintrittswahrscheinlichkeit: gering
Auswirkungen: mittel

Die Aufwands- und Zeitschätzung basiert auf dem derzeitigen Lastenheft des Auftraggebers und stellt eine zeitgerechte Fertigstellung sicher. Sollten sich jedoch die Anforderungen des Kunden während des Projekts ändern, so wird sich das mit großer Wahrscheinlichkeit verzögernd auf den Fertigstellungstermin auswirken. Die mit dem Kunden vereinbarte Funktionsanalyse und die Meilensteine mit gemeinsam festgelegten Qualitätskriterien sollten jedoch diesem Risiko entgegenwirken.

5.1.3 Technische Risiken

Datenverlust
Eintrittswahrscheinlichkeit: gering
Auswirkungen: mittel

Aufgrund der nicht auszuschließenden Gefahr des Datenverlusts, muss dafür gesorgt werden die Sicherheit der Daten, sowie auch die Verfügbarkeit dieser zu garantieren. Dieses Problem wird mithilfe eines GIT-Server (GIT) gelößt, durch diesen Server ist es möglich die Versionen der Software immer zugänglich zu machen und zusätzlich die Daten auf den Computern der Projektmitgliedern zu speichern.

KAPITEL 6

Conclusion and Outlook

"Success and failure are both greatly overrated. But failure gives you a whole lot more to talk about."

(Hildegard Knef)

ANHANG A

Appendix

Literaturverzeichnis

[1] G. Koppensteiner, R. Hametner, R. Paris, A. Passani, and M. Merdan, "Knowledge driven mobile robots applied in the disassembly domain," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, dec. 2011, pp. 52 –56.

[2] M. Merdan, "Knowledge-based multi-agent architecture applied in the assembly domain," Ph.D. dissertation, Vienna University of Technology, 2009.

[3] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE V2*. Wiley, 2007. [Online]. Available: http://www.amazon.ca/Developing-Multi-Agent-Systems-Fabio-Bellifemine/dp/0470057475/sr=8-1/qid=1170365284/ref=sr_1_1/702-0885532-1303250?ie=UTF8&s=books

[4] Wikipedia contributors, "Wissenschaftliche arbeit," Sep. 2012, page Version ID: 107839675. [Online]. Available: http://de.wikipedia.org/w/index.php?title=Wissenschaftliche_Arbeit&oldid=107839675

[5] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley, 2007. [Online]. Available: http://www.amazon.ca/Developing-Multi-Agent-Systems-Fabio-Bellifemine/dp/0470057475/sr=8-1/qid=1170365284/ref=sr_1_1/702-0885532-1303250?ie=UTF8&s=books

Erklärung

Hiermit erklären wir, dass die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Oktober 2012

Name1

Name2

Name3

Name4