

# KI an der Börse: Autonomous Trading Agents

---

---

verfasst von Benjamin Steinert  
- Universität Ulm -

Dieser Aufsatz handelt vom Einsatz künstlicher Intelligenz im Bereich „virtueller Börsenhandel und Spekulationsgeschäft“. Im Einzelnen wird das „Penn-Lehman Automated Trading Project“ der Universität von Pennsylvania betrachtet das als Basis für die Arbeit der darauf operierenden Agenten dient. Die Strategie namens CBR-SOBI wird als solcher Agent erläutert und wie „Case Based Reasoning“ im Detail Anwendung findet. Des weiteren wird der jährliche Wettbewerb zwischen den einzelnen Agenten kurz angerissen, und aufgeführt, wie die einzelnen im Wettstreit des Jahres 2003 abschnitten.

## 1.) Einleitung

Der Strukturwandel, den der Boom der Informationstechnologie mit sich brachte, hat in den letzten Jahren auch vor dem Wirtschafts- und Finanzmarkt nicht halt gemacht. Die Computerisierung hielt Einzug in jedem Börsensaal, in dem vor gut 20 Jahren noch Hunderte Makler um ihren Preis kämpften.

Heute werden der Großteil aller Börsengeschäfte direkt über den Computer erledigt, egal ob durch Menschenhand, oder einen Algorithmus. Sogenannte ECN's (Electronic Crossing Networks) stellen heute eine direkte Verbindung mit dem Markt dar, von jedem Standort auf der Erde. Diese Entwicklung wurde nicht zuletzt durch vollständige Computerisierung des NASDAQ ermöglicht, einem dem wichtigsten Technologie-Index der Welt.

Das „Penn-Lehmann Automated Trading Project“ (kurz: PLAT) ist ein Pilotversuch, einen autonomen Agenten mit realen Börsendaten zu konfrontieren. Dadurch soll nicht dem Menschen der Börsenhandel abgenommen werden, es ist unter anderem als Testplattform angelegt worden, um Strategien zu testen, ohne damit den realen Markt zu berühren. Das Projekt gliedert sich in den Exchange Simulator, der Handelsplattform der sogenannten Klienten, die dort ihr Geschäfte absolut autonom tätigen und analysieren. Zusätzlich zu den Entwicklungstests werden regelmäßig Wettbewerbe abgehalten, um die Agenten unter realen Bedingungen im Dauerbetrieb zu testen.

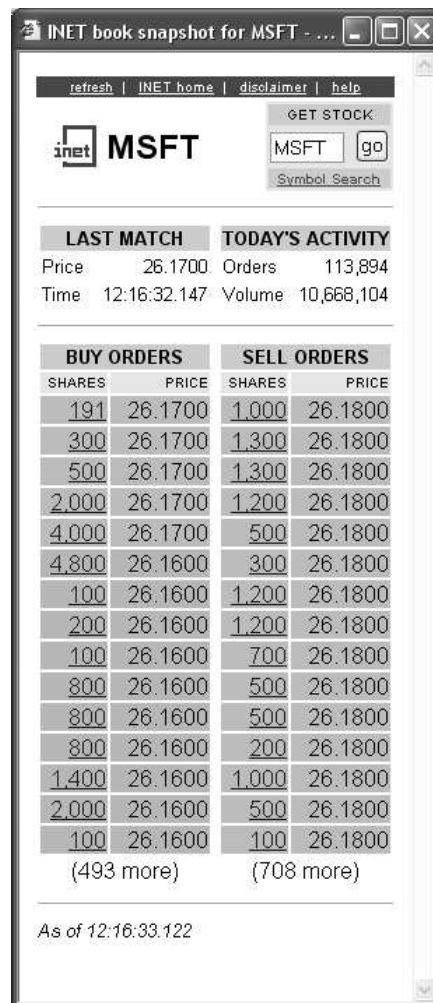
## 2.) strukturelle Grundlagen - ECN's und „Order Books“

Um das Konzept und die Funktionsweise des Projekts zu erklären werden nun grundlegende Aspekte erläutert. Dazu gehören die sogenannten ECN's (Electronic Crossing Networks), die einen direkten Zugriff auf den Markt des NASDAQ ermöglichen, und wie diese ECN's arbeiten.

Electronic Crossing Networks bilden anonyme Handelsplattformen für Aktien die im NASDAQ gelistet sind. Durch sie sind keine Broker mehr nötig, die für Investoren die Aktien auf dem Markt besorgen. Genauso kann ein Makler über einen oder mehrere ECN's seinen Handel tätigen. Der Handel und seine Ausführung bei ECN's funktioniert durch sogenannte Order Books, von denen es zwei verschiedene Arten gibt. Entweder man platziert ein Angebot oder eine Nachfrage als Limit Order oder als Market Order. Interessant für das PLAT-Projekt sind allerdings nur die Limit Orders. Hierbei wird nicht nur die gewünschte Menge der zu handelnden Aktien angegeben sondern auch der maximale Kaufs- oder Verkaufspreis (Limit). Die eingehenden Transaktionsaufträge beim ECN werden nun sortiert und zwar nach ihrer Attraktivität.

Hier ein Beispiel:

1000 Microsoft Aktien (NASDAQ Ticker Symbol ist MSFT) sollen zum Preis von 26.1700\$ gekauft werden. Mit dem Gebot würde der Auftrag direkt an sechster Stelle in der Warteschlange landen (siehe Abb. 1), da unser Preisangebot über 26.1600\$ liegt. Allerdings gab es vor uns schon einige Aufträge zum selben Preis, hier werden wir hinten eingereiht, direkt nach dem Gebot über 4000 Anteile zu je 26.1700\$.



The screenshot shows a web browser window titled "INET book snapshot for MSFT". It features a navigation bar with "refresh", "INET home", "disclaimer", and "help". Below this is a search area with "GET STOCK" and "MSFT" entered, followed by a "go" button and "Symbol Search" text. The main content is divided into sections: "LAST MATCH" (Price: 26.1700, Time: 12:16:32.147) and "TODAY'S ACTIVITY" (Orders: 113,894, Volume: 10,668,104). The core of the page is the "BUY ORDERS" and "SELL ORDERS" tables. The "BUY ORDERS" table lists 19 orders, with the top 6 orders at price 26.1700 and the remaining 13 orders at price 26.1600. The "SELL ORDERS" table lists 15 orders, all at price 26.1800. At the bottom, it indicates "(493 more)" buy orders and "(708 more)" sell orders. The timestamp "As of 12:16:33.122" is shown at the very bottom.

BUY ORDERS		SELL ORDERS	
SHARES	PRICE	SHARES	PRICE
191	26.1700	1,000	26.1800
300	26.1700	1,300	26.1800
500	26.1700	1,300	26.1800
2,000	26.1700	1,200	26.1800
4,000	26.1700	500	26.1800
4,800	26.1600	300	26.1800
100	26.1600	1,200	26.1800
200	26.1600	1,200	26.1800
100	26.1600	700	26.1800
800	26.1600	500	26.1800
800	26.1600	500	26.1800
800	26.1600	200	26.1800
1,400	26.1600	1,000	26.1800
2,000	26.1600	500	26.1800
100	26.1600	100	26.1800

Abb. 1: Typischer Snapshot von Island

Die hier beschriebene Warteschlange ist das sogenannte Buy Order Book, in dem alle „Asks“ vermerkt werden. Dem gegenüber stehen alle Verkäufe (Sell Order Book), die ebenfalls nach Attraktivität und Eingangszeit sortiert werden. Allerdings stehen hier die niedrigsten Angebote an vorderster Stelle.

Kommt es nun vor, dass Angebot und Nachfrage sich trifft, übernimmt das ECN die Transaktion und führt sie aus. Da in den wenigsten Fällen die Menge beider Ordern übereinstimmen, gibt es auch eine teilweise Ausführung. Will zum Beispiel jemand doppelt soviel verkaufen wie momentan zum selben Preis nachgefragt wird, werden sämtliche Transaktionen ausgeführt und die Restmenge weiterhin angeboten, bis weitere Anfragen eingehen.

Eines der bekanntesten und etabliertesten ECN's für NASDAQ Aktien ist „Island“ ([www.island.com](http://www.island.com)), das eine besondere Eigenschaft besitzt. Über das sogenannte „ITCH“-Protokoll, ist es möglich, sämtliche Daten über Aktivitäten von Island abzufragen.

### **3.) Das Projekt**

Das PLAT-Projekt untergliedert sich in zwei wesentliche Teilbereiche – die „Zentrale“, dem Penn Exchange Simulator (PXS), und den darauf operierenden Clients.

#### **a) Penn Exchange Simulator**

Der PXS ist das Herz des ganzen Projektes. Seine Hauptaufgabe besteht darin, als virtuelles ECN zu amtieren. Dabei besorgt es sich die aktuellen realen Order Book TOP15 von Island und integriert diese in den momentanen Status. Darüber hinaus werden sämtliche Daten gespeichert, um die Möglichkeit zu haben, Börsengeschehnisse der Vergangenheit authentisch zu simulieren.

Aufgrund der unzähligen Transaktionen, die Island in nur kurzer Zeit vollbringt, beschränkt sich der PXS auf den Handel mit Microsoft Aktien, und diese deshalb, weil Island einer der größten Märkte für diese Anteile ist.

Der Simulator hat zwei grundlegende Betriebsmodi: der historische Modus und der live Modus. Der live Modus kann immer nur dann aktiv sein, wenn auch Island den Handel eröffnet. Dabei aktualisiert PXS alle drei Sekunden seine Bücher und ist damit auf dem neuesten Informationsstand. Der historische Modus simuliert einen bereits zurückliegenden Börsentag. Hier ist das System nicht auf den Echtzeitfaktor angewiesen, sondern kann direkt die gespeicherten Daten nach dem internen Takt einfügen. Das erhöht zwar die Systemgeschwindigkeit deutlich, verdrängt aber ganz klar den Faktor Realismus. Dem ungeachtet hat der historische Modus natürlich auch seinen Sinn. Mit dessen Hilfe lassen sich Strategien von in Vergangenheit stattgefundenen Ereignissen entwickeln und natürlich darauf testen. Die entwickelten Methoden, bzw. – um gleich einen Schritt weiter zu gehen – Strategienpakete, werden dann als sogenannte Clients implementiert.

#### **b) PXS Clients**

Die Agenten, oder auch Clients bezeichnet führen völlig selbständig Transaktionen mit dem PXS aus. Die Unterprogramme arbeiten über speziell zugewiesene Ports auf dem System, dadurch können sie eindeutig identifiziert werden. Der Handel der Agenten läuft völlig dynamisch, d.h. zu jeder Zeit kann ein Client eingeklinkt und wieder abgeschaltet werden. Wie das Programm handelt und reagiert bestimmt und zugrunde liegende Strategie. Hierbei gibt es natürlich gravierende Unterschiede, wie auch im wirklichen Leben je nach Makler. Sowohl Risikobereitschaft als auch Sicherheit ist den Programmen wohl bekannt und die Entwickler der Agenten haben diese Aspekte mal mehr mal weniger gewichtig implementiert. Als Berater standen den Teilnehmern des Projektes, größtenteils Studenten der University of Pennsylvania, führende Aktienmakler des Finanzhauses „Lehman Brothers“ aus New York zur Seite, die sowohl ihre Erfahrungen einbrachten als auch teils ihre eigenen Strategien in

Agenten verwirklichten. Wie nun Client und PXS genau zusammen arbeiten, und wie der Simulator die gesammelten Daten verarbeitet, wird im folgenden erklärt.

### **c) Die PXS Execution Engine**

In der Execution Engine werden sämtliche Daten verwaltet, die der Simulator den Clients zur Verfügung stellt. Diese Aufgabe lässt sich in vier wesentliche Einzelschritte unterteilen.

- Aktualisierung der PXS Bücher mit aktuellen Daten von Island
- Das sogenannte „Book Cleaning“ das dafür sorgt, dass veraltete und nicht mehr aktuelle Informationen aus den Büchern gelöscht werden.
- Ausführung von offenen Kauf- und Verkaufsaufträgen
- Berechnung aller Agenten-relevanten Kennzahlen wie Gewinn und Verlust, Aktienvolumen und aktuelles Kapital.
- Einfügen von Client-Ordern in das PXS Order Book

Um sich nun ein detailliertes Bild über die Verzahnung von Island, PXS, und den Clients zu machen, wird im folgenden ein kompletter Aktualisierungsablauf beschrieben, wo sich auch die eben erwähnten Punkte wiederfinden.

Die folgenden Schritte werden wiederholt nacheinander ausgeführt bis die Simulation endet.

#### **1.) Abruf des Island Update**

*Die ITCH-Protokoll-Daten beinhalten die TOP 15 des aktuellen Island Order Book. Der Island Last Price wird übertragen und ebenfalls das „Island Total Volume“*

#### **2.) Übertragung der Daten in PXS**

*Jeder Island-Auftrag  $o = (\text{Typ}, \text{Preis}, \text{Vol}, \text{ID})$  durchläuft folgenden Integrationsprozess.*

- a) Überprüfe ob ID von Order xy bereits im PXS gespeichert wurde. Wenn ja, setze die zu handelnde Menge auf *vol* minus das Volumen sämtlicher Aufträge, die für diesen Posten bereits ausgeführt wurden. Ist das neue Volumen nicht positiv, wird der Auftrag aus der PXS Liste entfernt.
- b) Überprüfe ob die ID von Order xy bereits in der PXS exec-Liste auftaucht; wenn ja, ignoriere diesen Auftrag.
- c) Wurde kein Eintrag zu diesem Auftrag in den PXS Büchern gefunden, wird er vom System an die richtige Stelle, im richtigen Buch (Buy oder Sell, ja nach *Typ*) einsortiert. Die *ID* von Island bleibt erhalten.

#### **3.) PSX „Book Cleaning“**

*Da PXS nur die TOP15 aller Ordern von Island erhält, ist folgendes Problem gegeben:*

*Zum Zeitpunkt  $t_1$  erhält der Simulator die TOP15. Kurz darauf gibt es einen enormen Angebotsschub die sich alle auf den vordersten Plätzen einreihen. Dadurch werden einige oder gar alle Aufträge, die wir eben noch als TOP15 erhalten haben, aus der Liste verdrängt. Beim nächsten Update zum Zeitpunkt  $t_2$  stehen diese Ordern dann nicht mehr im Snapshot, obwohl sie natürlich noch Gültigkeit haben. Das sogenannte Book Cleaning ist eine Prozedur, die dieses Problem überwindet.*

*Jeder Auftrag  $o = (\text{Typ}, \text{Preis}, \text{Vol}, \text{ID})$  in den PXS Büchern wird nun geprüft.*

- a) Prüfe, ob Auftrag xy von Island stammt. Dafür betrachtet das System, ob die ID positiv (kommt von Island) oder negativ (von Client generiert) ist.

- b) Stammt die Order von Island, überprüfe, ob sie im aktuellen Island Snapshot erscheinen sollte (also ob der Kaufpreis/Verkaufspreis der 15. Order im Buy/Sell-Book über/unter unserer zu untersuchenden Order liegt.
- c) Wenn der Auftrag in der Island TOP15 auftauchen sollte, das aber in Wirklichkeit nicht tut, dann ist sie entweder zurückgezogen worden, oder seit dem letzten Update ausgeführt worden. Entferne Auftrag aus dem Simulator.

#### **4.) Durchführung von PXS Aufträgen mit dem vorhandenen Aktienvolumen von Island**

- a) Berechne das verfügbare Aktienvolumen aus den Islanddaten der letzten zwei Updates. (Subtraktion des totalen Volumens des letzten Datensatzes vom Aktuellen)
- b) Berechne das Gesamtvolumen aller PXS Kaufaufträge (id<0), deren Preis über oder gleich dem Island Lastprice ist.  
Berechne das Gesamtvolumen aller PXS Verkaufsaufträge, deren Preis unter oder gleich dem Island Lastprice ist.  
Liegt die Summe beider Gesamtvolumina unter den verfügbaren Aktienvolumen bei Island, deklariere alle PXS Ordern als erledigt und überführe sie in die Executed-List.  
Übersteigt das berechnete PXS Handelsvolumen die verfügbare Menge von Island, müssen von oben beginnend alle Aufträge der Reihenfolge nach ausgeführt werden. Dabei kann es natürlich vorkommen, dass Aufträge nur teilweise erfüllt werden. (siehe 5.) Die restliche Anweisung bleibt natürlich erhalten.
- c) Tritt der Fall ein, dass kein Auftrag ausgeführt wurde, dann setze den PXS Lastprice gleich dem Island Lastprice.

#### **5.) Durchführung von PXS Aufträgen**

- a) Durchlaufe alle Aufträge in den PXS Büchern und prüfe auf übereinstimmende Preise.
- b) Überführe jede ausgeführte Order in die Executed-List.
- c) Aktualisiere den PXS Lastprice nach jeder ausgeführten Transaktion.

#### **6.) Bearbeitung von eingegangenen PXS Client Aufträgen**

Prüfe jede neu eingegangene PXS Client Order ob sie direkt ausgeführt werden kann, und aktualisiere ggf. den PXS Lastprice.  
Ist das nicht der Fall, sortiere sie an der richtigen Stelle in der Liste ein.

#### **7.) Aktualisierung des PXS „Public View“ und Durchstellung von PXS Aufträgen**

Aktualisiere den PXS „Public View“, und sende diesen an alle angeschlossenen Agenten. (Der „Public View“ umfasst das gesamte Order Book, den Lastprice usw.)

#### 4.) Funktionsweise eines Agenten am Beispiel „CBR-SOBI“

Das Projekt beschäftigte in der heißen Phase der Strategienentwicklung bis zu 30 Personen, die verteilt an 14 verschiedenen Strategien arbeiteten. Jede Strategie unterscheidet sich wesentlich von der anderen, Aufbau, Intelligenz und Entscheidungsrouitinen wurden entweder auf bereits vorhandenem Wissen aufgebaut oder nach eigenen Ideen entwickelt und implementiert.

Aufgrund der teils sehr hohen Komplexität der Strategien wird an dieser Stelle nur auf CBR-SOBI als Beispiel eingegangen.

##### a) Aufbau

CBR-SOBI besteht, wie der Name schon vermuten lässt, aus zwei wesentlichen Elementen: Der SOBI-Strategie (Static Order Book Imbalance), auf der CBR (Case Based Reasoning) aufsetzt.

SOBI wurde von Micheal Kearns, dem Leiter des Projekts, und seinem Team eigens entwickelt. Es handelt sich hierbei um eine Sammlung von Strategien, die auf der Basis von regelmäßig erstellten Preis-Mengen-Graphen Entscheidungen fällt. Tritt eine große Abweichung zwischen Angebot und Nachfrage auf so ist das ein Zeichen für eine baldige Preisänderung. Darauf angemessen und gewinnbringend zu reagieren hat sich SOBI zur Hauptaufgabe gemacht. Dabei zerlegt sich der Analyseprozess in folgende Schritte:

- 1.) Berechne den Durchschnittspreis  $D_p$  von  $n$  Ordnern unter Einbezug des Verhältnisses der Anteilsmengen  $v$  zum Preis  $p$  der Menge [d.h.  $D_p = (\sum_{i=1..n} v_i * p_i) / (\sum_{i=1..n} v_i)$ ], jeweils im Buy und Sell Order Book. Dabei kann vom Benutzer noch festgelegt werden, welches Quartil  $q$  die Strategie berücksichtigen soll. Das gibt im Endeffekt Einblick darüber, wie stark die erkannte Marktänderung sein wird.
- 2.) Subtrahiere den Marktpreis vom Durchschnittspreis des Sell Order Book. Dieser Wert heißt  $S_q$ .
- 3.) Subtrahiere den Durchschnittspreis des Buy Order Book vom Marktpreis, das Ergebnis heiße  $B_q$ .
- 4.) Berechne den aktuellen Marktpreisunterschied  $\Delta p$  zwischen Angebot und Nachfrage. Ist  $B_q - S_q$  nun größer als die vom Benutzer definierte Variable  $\tau$  (Entscheidungs-größe, ab welcher der Client in Aktion treten soll), dann platziere einen Kaufauftrag,  $x$  Aktien ( $x$  ist vordefiniert) zum Preis von Lastprice  $L - \Delta p$ . Ist  $S_q - B_q$  größer als  $\tau$ , dann platziere einen Verkaufsauftrag mit  $x$  Anteilen zum Preis von  $L + \Delta p$ .

Zur Erklärung: In diesen Schritten macht die SOBI Strategie eigentlich nichts anderes, als abzufragen, ob momentan eine besonders ansteigende Nachfrage vorliegt, oder ob es sich „lohnt“ zu kaufen, da der Preis fallen wird. Und da Gebote bei gleichem Preis nach dem „First Come First Served“ Prinzip abgearbeitet werden, empfiehlt es sich natürlich, so früh wie möglich das Gebot abzugeben, wenn Marktzeichen vorliegen.

Der große Haken an der SOBI-Strategie ist, dass sie nur aufgrund von momentanen Gegebenheiten Entscheidungen fällt, und vor allem kein „Gedächtnis“ besitzt. Es kann also durchaus sein, dass sie bestimmte Marktsituationen falsch interpretiert und das immer wieder.

Effektive Agenten arbeiten mit Börsendaten der Vergangenheit die sie als Anhaltspunkt für die Strategiefindung nutzen. Diese Teilbereiche übernimmt das CBR-Modul, das durch die Projektarbeit von drei Teilnehmern dazugekommen ist.

## b) Das CBR Modul

Case-Based Reasoning, oder CBR, ist ein allgemeiner Begriff aus der künstlichen Intelligenz, unter dem man den Vorgang des maschinellen Lernens durch Erstellen von Fällen (Cases) zum einen versteht, und zum anderen die Auswahl eines Falls für die Reaktion auf ein erkanntes, stattgefundenes Ereignis, das bereits in der „Erfahrungsdatenbank“ vorliegt. CBR-SOBI ist nun in der Lage, Marktsituationen in Form von Cases abzuspeichern, um sie im späteren Verlauf einzusetzen. Dafür besitzt der Agent zwei verschiedene Betriebsmodi: den „Case-Discovery-Modus“ und den „Case-Based-Reasoning-Modus“.

### ❖ Phase 1: „Case-Discovery-Modus“

Bevor der Client seine Arbeit aufnehmen kann, muss er sich genügend Fälle für die spätere Arbeit generieren. Das geschieht dadurch, dass seine aller erste Aktion absolut zufällig<sup>1</sup> gewählt wird. Im selben Moment speichert er die aktuellen Charakteristika des Marktes und die Merkmale der Aktion (Set an SOBI-Variablen) die er ausführt. Danach überprüft der Agent, ob die Handlung ein Erfolg war oder nicht und speichert das Resultat in Form einer Erfolgsquote  $E_q$ . Der Erfolg einer erstellten Aktion wird immer am Portfolio des Clients gemessen. Dies berechnet sich aus der Summe des vorhandenen Kapitals plus dem aktuellen Wert der momentanen Aktien im Besitz. Alle gesammelten Daten bilden dann einen Fall, der in der Client-Datenbank abgelegt wird, die als Baum aufgebaut ist (siehe Abb. 2).

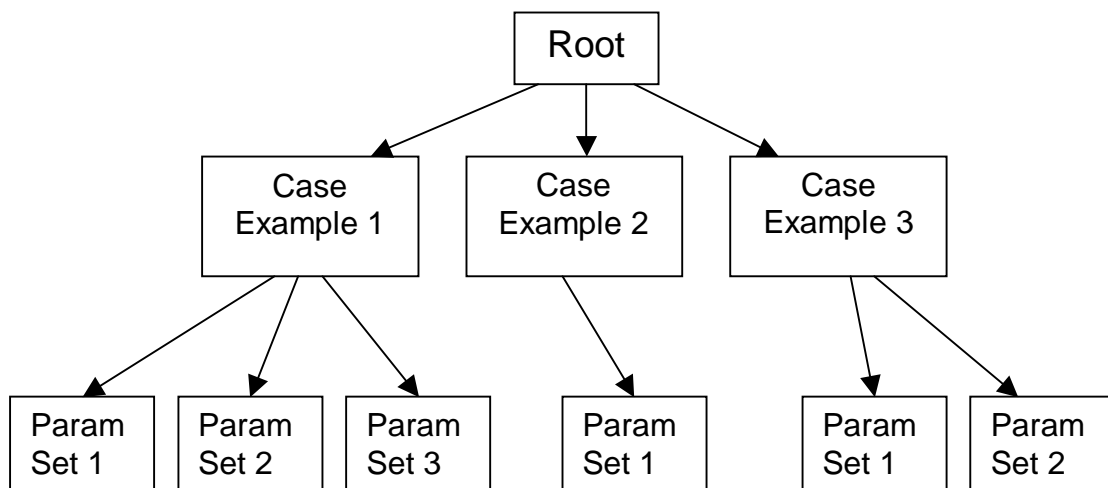


Abb. 2: Root bildet hier lediglich den Startknoten. Jede Abzweigung davon bildet einen Fall. Alle SOBI-Parameter-Sets bilden Blätter, die am jeweiligen Fall hängen

Den Case-Discovery-Prozess wiederholt der Client solange, bis die voreingestellte Anzahl von Fällen  $n$  und Sets  $m$  erreicht ist. Dabei geht der Modus im Detail wie folgt vor:

- 1.) Generiere die Problemstellung, mit anderen Worten die Momentaufnahme des Marktes (Erfassung von  $S_q$ ,  $B_q$  usw.).
- 2.) Überprüfe, ob dieser Fall im groben bereits vorliegt.  
Wenn ja überprüfe, ob für diesen Fall bereits  $m$  verschiedene Sets existieren, wenn ja, wähle den Set mit der höchsten Erfolgsquote  $E_q$ . Ist die Anzahl an Sets  $< m$  dann erzeuge ein neues zufälliges Set zu diesem Fall und initialisiere  $E_q=0$ . Wenn nein, dann gliedere diesen neuen Fall in die Datenbank ein und erzeuge ein zufälliges Set an Variablen.
- 3.) Führe die Aktion mit den eben gewählten, bzw. erzeugten Variablen aus.

<sup>1</sup> Natürlich belaufen sich die Werte in vorher festgelegten Grenzen, um durch die Experimente nicht zu viel Verlust zu machen.

- 4.) Überprüfe, ob die letzte ausgeführte Aktion einen Erfolg gezeigt hat. Wenn ja dann erhöhe die Erfolgsquote der verantwortlichen Aktion um eins, wenn sich ein negativer Effekt zeigt, dann senke Eq um eins. Ist keine Reaktion auf die Aktivität erkennbar, senke Eq um 0,3 Punkte.
- 5.) Ist die maximale Anzahl an Fällen erreicht? Wenn ja dann wechsele den Modus. Wenn nein, dann beginne von vorne beim nächsten relevanten System-Tick.

Wie unschwer zu erkennen geht die Strategie ganz simpel nach dem Trial & Error Verfahren vor, um Erfahrungen zu sammeln. Im weiteren Verlauf sammelt der Agent immer mehr Momentaufnahmen des Marktes und viele dazugehörige Lösungen, die nach Erfolgsgrad sortiert in der Datenbank eingetragen sind.

❖ Phase 2: „Case-Based-Reasoning“

Hat der Agent alle Voraussetzungen an Wissen erfüllt, wechselt er den Betriebsmodus. Ab sofort kommen keine neuen Fälle hinzu.

Nun sucht der Client bei jedem System-Tick nach einem passenden Fall in der Datenbank für die momentane Marktlage. Findet er einen exakten Fall (minimale Abweichungen sind natürlich vernachlässigbar), entscheidet er sich für das dazugehörige Set an Parametern mit der höchsten Erfolgsquote und veranlasst eine Order.

Tritt der Fall ein, dass es keinen exakt passenden Fall in der Wissensdatenbank gibt, werden die der Problemstellung naheliegendsten Fälle betrachtet. Nun verwendet der Agent entweder den Fall, der am nächsten an der Marktsituation liegt, oder wählt, bei mehreren relevanten Fällen, das Set mit der höchsten Eq aus allen Sets der relevanten Fälle aus. Die dritte Möglichkeit ist, dass der CBR-SOBI ein Hybrid-Set erstellt, das heißt er entscheidet sich für den Durchschnitt aller relevanten Sets, um das Risiko einer Fehleinschätzung gering zu halten. Den letzten Fall verwendet das System allerdings nur, wenn zu viele verschiedene Parameter-Sets zur Auswahl stehen.

Natürlich hat CBR-SOBI auch während der 2. Phase noch die Möglichkeit „dazuzulernen“, da im Regelfalle während der Strategiefindung zwar viele Fälle gesammelt werden können, aber aufgrund der Zufallserzeugung der Sets diese meistens nicht viel Wert sind, und nur durch Zufallstreffer gute Strategien gefunden werden. Deshalb sortiert das System nach jeder Aktionsbewertung unbrauchbare und selten bis gar nicht verwendete Sets aus. Dazu gehören solche, die entweder einen  $Eq < -5$  haben, oder die Differenz vom Besten zum schlechtesten Set  $> 10$  ist. Diese werden durch neue Zufallszahlen ersetzt und der Eq auf 0 gesetzt.

Da ein Set mit  $Eq=0$  in einem ausgebauten Fall auch keine Chance hat, gibt es noch die Möglichkeit für den CBR-SOBI, ein rausfallendes Set durch eine Interpretation von zwei weiteren sehr guten Sets zu ersetzen. Dabei wird zum Beispiel festgestellt, dass Set1 insgesamt höhere Variablen besitzt als Set2, Set2 aber die bessere Erfolgsquote hat. Darauf basierend erstellt dann der Client ein Set3, das noch etwas kleinere Variablen besitzt als Set2, um so vielleicht die Strategie zu verbessern. Das neue Set erhält dann auch eine höhere Erfolgsquote, und zwar den Durchschnitt von allen beteiligten Sets.

Es wurde nun die grundlegende Funktionsweise des Agenten CBR-SOBI, sowie der SOBI-Strategie selbst, vorgestellt. Darüber hinaus besitzt der Client noch eine weitere Analyseverfahren namens VWAP, die allerdings aufgrund des Umfangs an dieser Stelle keine Erwähnung findet. Es ergibt sich deshalb aber keine Einschränkung der Aussagen, da mithilfe von VWAP die Fallstellung bzw. -findung spezialisiert und damit eindeutiger bzw. sicherer gemacht wird, sie jedoch keinerlei Einfluss auf die grundlegende Funktionsweise hat. Es fanden des weiteren natürlich auch etliche kleine Erweiterungen statt, sodass man mit Sicherheit davon ausgehen kann, dass der Faktor „künstliche Intelligenz“ im Falle von CBR-SOBI noch weitaus größer ist als hier angedeutet.

## 5.) Der Wettbewerb

Damit die Entwickler die endgültige Effizienz ihrer Agenten untereinander messen konnten, wurden in regelmäßigen Abständen Wettbewerbe abgehalten. Beim dritten Wettbewerb im Frühjahr 2003 hatten Entwickler und Betreuer ein komplettes Regelwerk aufgestellt, das über Gewinner und Verlierer entschied. Aufgrund der Tatsache, dass die ersten Wettbewerbe aufgrund des Fehlens dieser Regeln wenig Aussage hatten, werden sie hier nicht erwähnt.

Der komplette Wettbewerb wurde, wie schon alle Simulationen zuvor, mit Microsoft-Aktien ausgetragen. Der Grund dafür war, dass sämtliche Agenten speziell auf diese Anteile ausgelegt waren, und das musste geschehen, weil jede Aktie auf dem Markt anderen Regeln des Handels und Marktes folgt. So wäre es zum Beispiel nicht möglich, das PLAT-Projekt kurzerhand mit einer anderen Aktie zu betreiben, da einfach die Strategien nicht passen würden, und Marktverhalten vielleicht falsch interpretiert würde.

Einer der Hauptaspekte des Wettbewerbs war natürlich Gewinn und Verlust, wie im wirklichen Leben müssen die Agenten mit dem Risiko und den damit verbundenen Profit kalkulieren.

Eines der Hauptrisiken der Clients waren Aktiengeschäfte mit gewaltigem Volumen. Zum einen weil durch Transaktionen mit sehr großen Mengen der Markt nachhaltig negativ beeinflusst wird, zum anderen verliert der Agent natürlich mehr Geld, wenn es zu einem Preisrutsch kommt, und er sehr viele Anteile besitzt. Darum wurde entschieden, dass das maximale Anteilsvolumen eines jeden Klienten nicht höher als 100000 sein darf. Somit wird unterbunden, dass Agenten einfach hohe Mengen zu günstigen Preisen ankaufen und danach wieder abstoßen, was den Marktpreis rapide drückt.

Die Kriterien des Wettbewerbs hier in Kurzform:

- **Kriterien zur Profitabilität:**
  - *Gewinn und Verlust (G&V) pro Tag (G):*  
Täglich werden dem Agenten mit dem höchsten Tagesprofit drei Punkte gutgeschrieben, dem zweitbesten zwei und dem Drittplatzierten ein Punkt.
  - *G&V über den gesamten Wettbewerb (GgW):*  
Am Ende des Wettbewerbs erhält jeder Client einmalig 15 Punkte, der an nicht mehr als 3 Tagen einen negativen G&V Stand hatte, und die insgesamt über den Wettbewerb kumulierten G&V Stände positiv ausfielen.
- **Kriterien zu Robustheit und „Intraday Trading“**
  - *Umgang mit Marktschwankungen (M):*  
Fünf Punkte Belohnung erhält jeder Agent, der an zwei aufeinanderfolgenden Tagen, bei denen es am ersten eine Preisexplosion gab und am zweiten einen ganztägigen Preisrutsch, ein positives G&V aufweisen kann.
  - *Konzentration auf Intraday Trading (I):*  
Täglich erhält jeder Client 2 Punkte, der einen positiven G&V Stand aufweisen kann, und die Aspekte des Intraday Trading beachtet hat. (Beim Intraday Trading handelt es sich um eine Strategie, die sich auf den Kauf und Verkauf von Anteilen am selben Tag konzentriert. Dabei soll erreicht werden, dass man den Tag mit null oder wenig Anteilen beginnt und auch wieder beendet.) Allerdings soll auch an mindestens einem Punkt des Tages das Anteilsvolumen mindestens 10000 betragen, um Tatenlosigkeit nicht zu belohnen,
- **Kriterien zur Bewertung der Handelsstrategien, ohne Beachtung des G&V**
  - *Tägliche Risikosättigung (R):*  
Pro Tag erhält jeder Client zwei Punkte, dessen gehaltene Aktienmenge an einem Punkt des Tages 50000 überschreitet, aber unter der Höchstgrenze von 100000 Anteilen bleibt.
  - *Tägliche Depotleerung (U):*  
Täglich hat jeder Client der Punkte für die Risikosättigung erhielt, auch noch die Chance zwei Punkte für das sog. „Position Unwinding“ zu erhalten wo es darum geht, dass das Depot des Clients am Ende des Tages nicht mehr als 5000 Aktien enthält. Die Risikosättigung ist deshalb Voraussetzung, um Agenten mit ohnehin

schon niedrigem Aktienanteil über den gesamten Tag hinweg von der Punktvergabe auszuschließen.

Um den PXS und CBR-SOBI selbst im Vordergrund zu behalten begnüge ich mich an dieser Stelle mit der Angabe des Endstandes nach dem Wettbewerb, da eine genaue Beschreibung des Verlaufs für alle Agenten wohl den Rahmen dieser Arbeit sprengen würde.

Strategie	Pool	Rang	Tot	G	R	U	I	GgW	M	Ø G&V
<b>CBR-SOBI</b>	B	1	74	9	18	2	10	15	20	4187
<b>MoneyFlow</b>	B	2	69	15	20	0	4	15	15	2007
<b>OBMM</b>	B	3	46	8	20	0	8	0	10	258
<b>CreaTiv</b>	B	4	42	7	20	0	10	0	5	-2410
<b>OBCrossover</b>	B	5	33	6	6	0	6	0	15	3242
<b>OBBreakout</b>	B	5	33	10	18	0	0	0	5	3680
<b>RaSTa</b>	B	7	21	5	2	0	4	0	10	1182
<b>DAMM-STAT</b>	R	1	65	6	20	14	10	0	15	685
<b>Contrarian</b>	R	2	55	6	20	2	12	0	15	2022
<b>OBSigma</b>	R	3	54	8	20	6	10	0	10	1649
<b>OBVol</b>	R	4	53	14	0	0	4	15	20	4037
<b>RapidMM</b>	R	5	50	10	20	0	10	0	10	3649
<b>CIA</b>	R	6	30	13	12	0	0	0	5	-1451
<b>SimpleTrend</b>	R	7	27	3	20	2	2	0	0	-24467

## 6.) Ausblick und Schlusswort

Während diese Ausarbeitung erstellt wurde, entschied sich der Wettbewerb des Jahres 2004 unter den Klienten. Es bleibt abzuwarten, ob eines Tages wirklich eine „marktreife“ Strategie zum Vorschein kommen wird, oder ob der Computer so schnell nicht in der Lage sein wird, sich mit dem Mensch in Sachen Intuition und Risikobereitschaft zu messen. Gezeigt hat diese Ausarbeitung eines: Die KI hat bewiesen, dass sie in der Lage ist, komplexe wirtschaftliche Prozesse zu erkennen und auf sie zu reagieren, und das natürlich je nach Strategie unterschiedlich gut. CBR-SOBI ist vielleicht die ausgereifteste Strategie des Wettbewerbs gewesen. Allerdings wurden die Ziele dieses Jahr noch höher gesteckt, was definitiv einen noch besseren und ausgereifteren Agenten hervor gebracht hat.

Literaturverzeichnis:

- Michael Kearns und Luis Ortiz: „The Penn-Lehman Automated Trading Project“, Datum nicht bekannt  
<http://www.cis.upenn.edu/~mkearns/projects/plat.html>
- Gruppe 46D (Sean Lao, Hoi Yee Leung, Tan Fei Wong), Teilnehmer am PLAT Project 2003: „Penn-Lehman Automated Trading Project: Case Based Reasoning“ Final Report (5/9/03), unveröffentlicht, persönliche Kommunikation