<TRADEKNOWLEDGE/>

Konzeption, Entwicklung, Test und Betrieb von Handelssystemen auf Basis eines wissensbasierten Modells

DETLEV MATTHES, DMATTHES@WEB.DE

OKTOBER 2022

WORUM GEHT ES UND WAS IST DAS ZIEL?

Vorstellung eines Entwicklungsprozesses für automatisierte Handelssysteme mit dem Ziel:

- Zugriff auf gesammeltes Wissen und Erfahrungen von Experten, Tradern und Handelssystementwicklern in Form einer Wissensdatenbank
- Öffentlichkeit und Erweiterbarkeit der Wissensdatenbank
- Zielgerichtete Interpretation von Wissen und den Erhalt von Hinweisen, so z.B. „hier ist ein anderer Indikator besser geeignet“ oder „es fehlt noch eine Stopp-Order für die Verlustbegrenzung“
- Verbergen von umgebungsspezifischen Besonderheiten, wie die zu verwendende Programmiersprache
- Wiederverwendbarkeit von Bestandteilen, wie z.B. Algorithmen, Indikatoren und Setups

LUST AUF EIN KLEINES EXPERIMENT?

Wir möchten ein einfaches Handelssystem mit den folgenden Handelsregeln automatisiert umsetzen:

Einstiegsregel:
Kaufe 100 Aktien einer Firma wenn am Ende des Tages der einfach gleitende Durchschnitt (20 Perioden und basierend auf dem Tages-Schlusskurs) den einfach gleitenden Durchschnitt (20 Perioden und basierend auf dem aktuellen Wochen-Schlusskurs) von Unten nach Oben kreuzt.

Ausstiegsregel:
5%-Trailing-Stopp

Probieren Sie es doch einmal selbst aus und realisieren das System für einen Backtest in einer Umgebung ihrer Wahl.

(Nachfolgend erfolgt eine beispielhafte Umsetzung in ProBuilder für ProRealtime)

UMSETZUNG IN PROREALTIME WWW.PROREALTIME.COM/DE

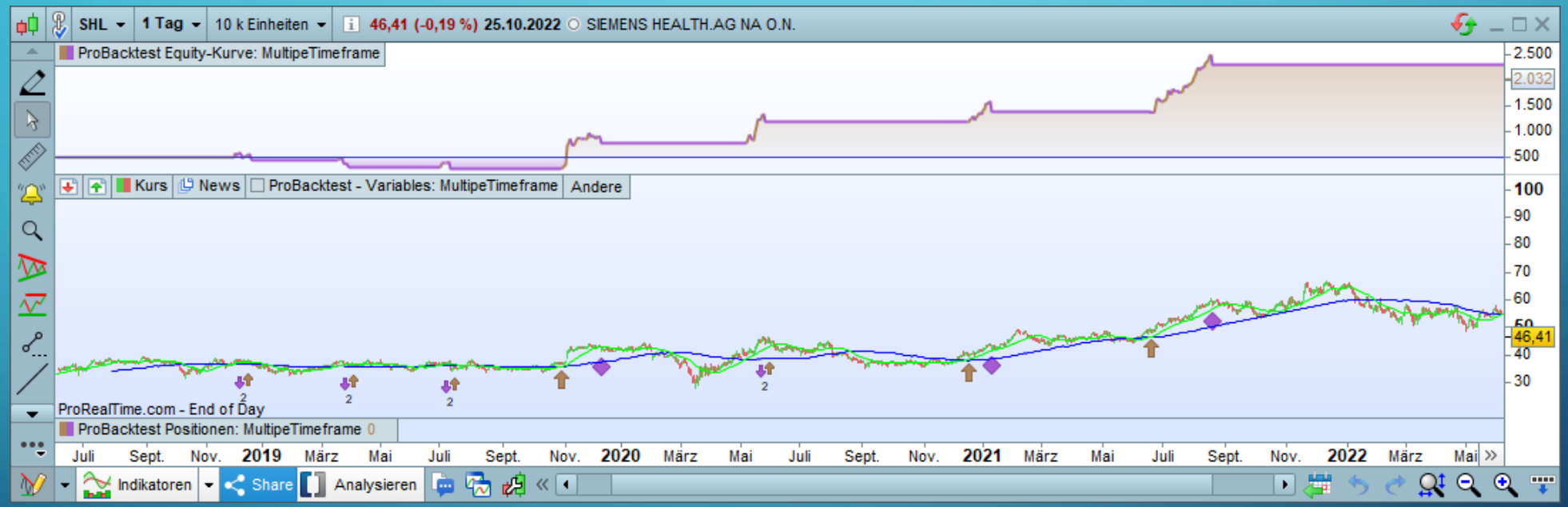
Für ProRealtime schreiben wir den folgenden Programmcode in der Sprache ProBuilder:

```
TIMEFRAME(WEEKLY)
sma20OfWeek = AVERAGE[20](close)
GRAPHONPRICE sma20OfWeek COLOURED (0,0,255) as "SMA(20) of weekly close"

TIMEFRAME(DEFAULT)
sma20OfDay = AVERAGE[20](close)
GRAPHONPRICE sma20OfDay COLOURED (0,255,0) as "SMA(20) of daily close"

buySignal = sma20OfDay CROSSES OVER sma20OfWeek
IF buySignal AND NOT LONGONMARKET THEN
   BUY 100 SHARES AT MARKET
ENDIF

SET STOP %TRAILING 5
```

NACH DER REALISIERUNG

Es wird ein Backtest ausgeführt und es wird ein Equity-Verlauf angezeigt.

(Wir wollen vorerst die Performance des Handelssystems nicht beachten)



Beispielhafter Tages-Chart und Equity-Verlauf der Aktie SHL in ProRealtime

Bitte nehmen Sie sich Zeit und überlegen:

- Was ist an dieser Umsetzung nicht optimal oder auch falsch?
- Ist diese Umsetzung für einen praktischen Handel geeignet?

DOCH NICHT SO OPTIMAL!

Sind Ihnen u.a. die folgenden Punkte aufgefallen?

- Der Schlusskurs des Tages und der Schlusskurs der Woche werden für die Signalgenerierung miteinander kombiniert.
  Haben Sie beachtet, dass der Schlusskurs der aktuellen Woche erst mit Ende der Woche gültig ist?
  Vor Abschluss der aktuellen Woche können im realen Handel Fehlsignale entstehen.
- Es ist kein echtes Risiko- und Money-Management realisiert. Unabhängig vom aktuellen Kurs einer Aktie und dem jeweils verfügbaren Kapital werden mit jedem Einstiegssignal einfach 100 Aktien gekauft.
- Im realen Handel sind kaum Ausführungen zu denen, im Backtest angenommenen, Schluss- und Stopp-Kursen möglich.
- Ein praktischer Handel ist nur umsetzbar, wenn das Handelssignal täglich ausgewertet und eine Order aufgegeben werden kann.

- Und viele andere Kleinigkeiten …

WAS SOLL UNS DAS ZEIGEN?

Bei der Realisierung von automatisierten Handelssystemen kann viel falsch gemacht werden. Wir denken z.B. nicht an das Risiko- und Moneymanagement oder nicht an die Machbarkeit und die Umstände für einen praktischen Handel.

Es sind Kenntnisse und Fähigkeiten in der jeweiligen Programmiersprache und Entwicklungsumgebung notwendig.

Es sind Kenntnisse zur Börse, der Verwendung technischer Indikatoren, von Fall-Stricken, von Best-Practice-Ansätzen und vieles mehr notwendig.

WAS SOLL UNS DAS ZEIGEN?

Die Entwicklung von automatisierten Handelssystemen ist ein komplexer Vorgang und setzt ein hohes Maß an Wissen und an praktischen Erfahrungen voraus.

Die Möglichkeiten eines praktischen Einsatzes sind bereits bei der Entwicklung zu beachten und zu überprüfen.

WIE WÄRE ES DAMIT?

- Was wäre, das komplette Wissen über die Struktur und den Aufbau eines Handelssystems, bekannter Fall-Stricke, Best-Praxis-Ansätze und der korrekten Verwendung von technischen Indikatoren und Orderarten zu besitzen und anwenden zu können?

- Was wäre, Jemanden „hinter unseren Rücken“ zu haben,
  - der uns gleich mitteilt, wenn wir etwas falsch machen?
  - der uns zeitnah und zielgerichtet Tipps, Hinweise und Vorschläge geben kann?

- Was wäre, Jemanden zu haben, der das Verhalten des Handelssystems im praktischen Einsatz überwacht?

DAS GESAMMELTE WISSEN

Stellen wir uns ein Regal voll mit Büchern mit den folgenden Titeln vor:

„Der Aufbau und die Struktur von Handelssystemen“

„Indikatoren der technischen Analyse und deren Verwendungen“

„Die Programmiersprache Equilla“

„Die besten Tipps- und Tricks von Profis“

„Bekannte Fehler bei der Handelssystementwicklung“

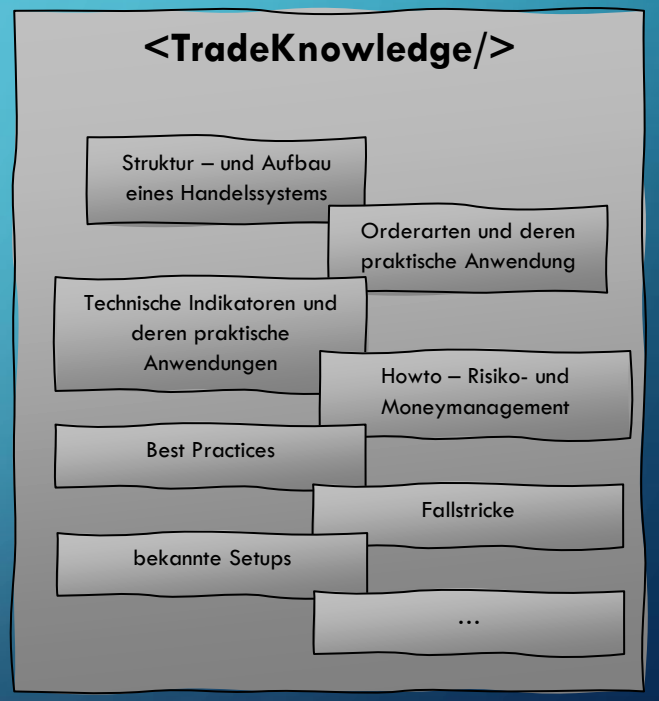
„Orderarten und deren Verhalten an der Börse“

… und viele weitere Bücher und Artikel

Lassen wir all diese Bücher mit <TradeKnowledge/> abbilden.

WAS IST <TRADEKNOWLEDGE/>?

- <TradeKnowledge/> ist eine Wissensdatenbank.
- Das abgebildete Wissen ist für jeden einsehbar.
- Im aktuellen TradeKnowledge sind die Basis an Grundwissen und die Struktur eines einfachen Handelssystems mit den einfachsten Orderarten abgebildet.
- Das abgebildete Wissen kann jederzeit um neue Kenntnisse erweitert werden und Irrtümer korrigiert werden. Es kann kontinuierlich mit dem Wissen von Experten und Entwicklern erweitert werden.



WIE WIRD <TRADEKNOLEDGE/> VERWENDET?

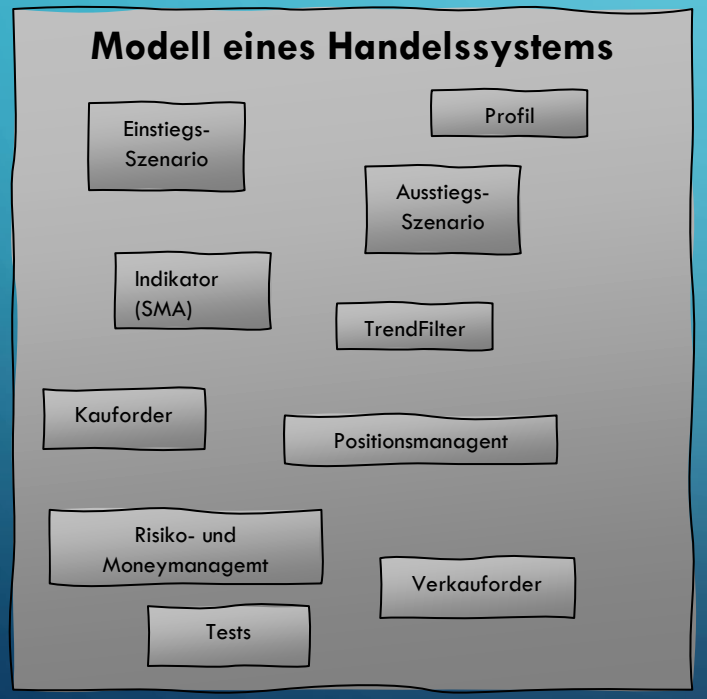
- Das abgebildete Wissen liegt mit <TradeKnowledge/> in einem maschinen-lesbaren Format vor (XML-Format) und ist somit durch Software interpretierbar.

- Während der Entwicklung eines Handelssystems kann somit auf <TradeKnowlege/>, und damit auf das abgebildete Wissen, durch entsprechende Entwicklerwerkzeuge zugegriffen werden.

- Um zu entwickelnde Handelssysteme softwaretechnisch und automatisiert mit <TradeKnowledge/> auswerten zu können, werden diese als Modelle abgebildet.

DAS HANDELSSYSTEM ALS MODELL

Mit dem Modell eines Handelssystems werden dessen Struktur, verwendete Algorithmen, Indikatoren, Orderanweisungen u.a. abgebildet:



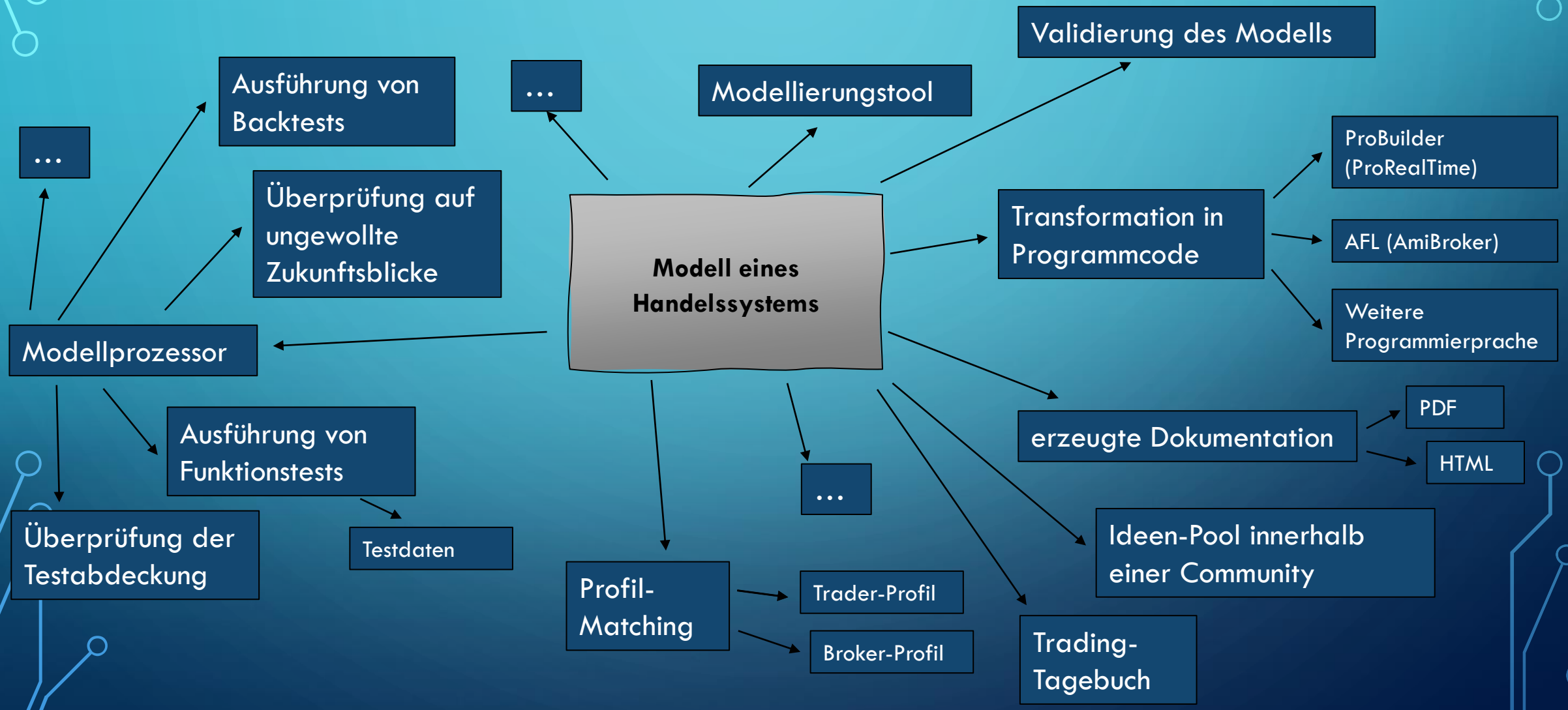
Bestandteile:

- Szenarien für Einstieg und Ausstieg
- Verwendete Algorithmen und Indikatoren
- Trendfilter
- Orderanweisungen
- Positionsmanagement
- Risiko- und Money-Management
- ggf. Funktionstests und Profile

DAS HANDELSSYSTEM ALS MODELL

- Das Modell eines Handelssystems ist durch seine Beschreibung (z.B. das XML-Format) maschinell interpretierbar.
- Durch diese Interpretierbarkeit kann das Modell durch verschiedenste Werkzeuge ausgewertet und verarbeitet werden.
- Damit ergibt sich eine hohe Vielzahl an Anwendungsmöglichkeiten:

VIELFÄLTIGE ANWENDUNGSMÖGLICHKEITEN FÜR EIN MODELL:



MODELLBASIERTER ENTWICKLUNGSPROZESS (VEREINFACHT)

Modellierung des Handelssystems

Modell eines Handelssystems

Validierung des Modells gegenüber TradeKnowledge

Gegenüber <TradeKnowledge/> gültiges Modell

Transformation des Modells in Programmcode

Programmcode des Handelssystems

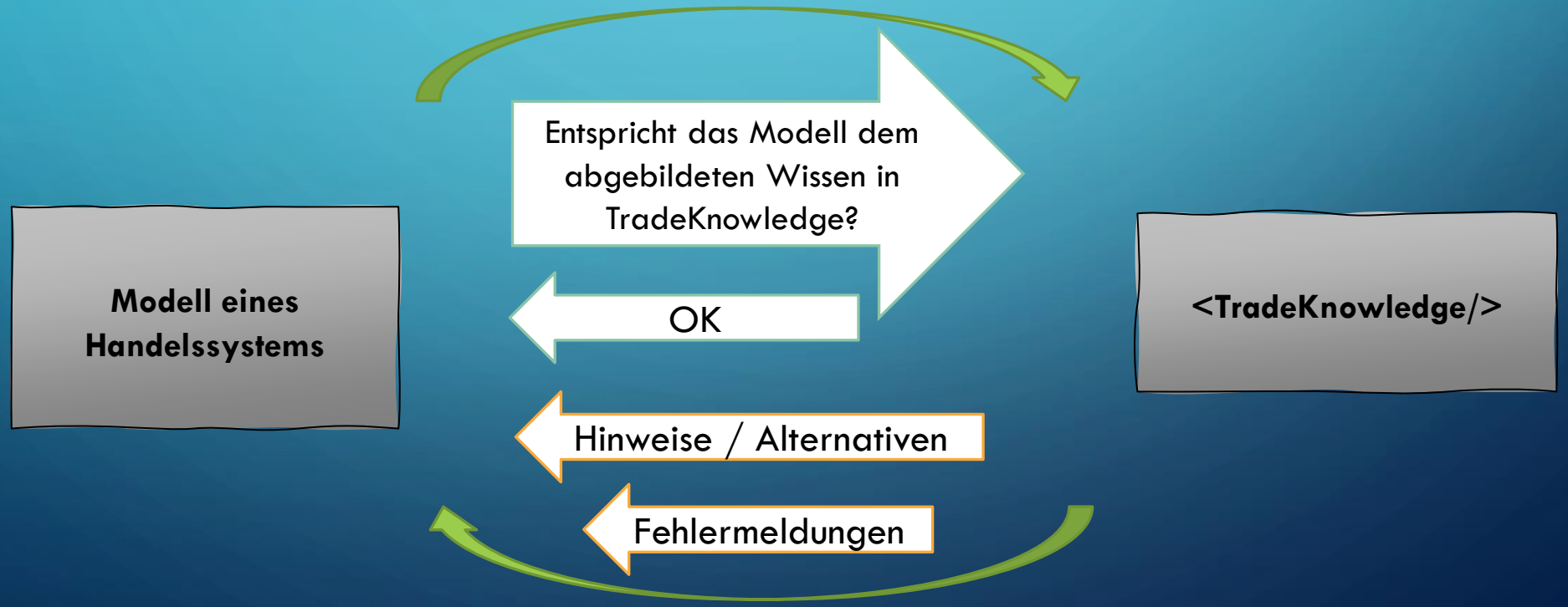
ENTWICKLUNGSPROZESS: MODELLIERUNG EINES HANDELSSYSTEMS

- Der Handelssystementwickler bildet die Bestandteile (und deren Beziehungen untereinander) des Handelssystems innerhalb eines Modells ab.
- Bestandteile können sein: Szenarien für Ein- und Ausstieg, Orderanweisungen, Indikatoren, Algorithmen, Filter, …

- Diese Abbildung (Modellierung) kann mit einem entsprechende Modellierungstool unterstützt werden.

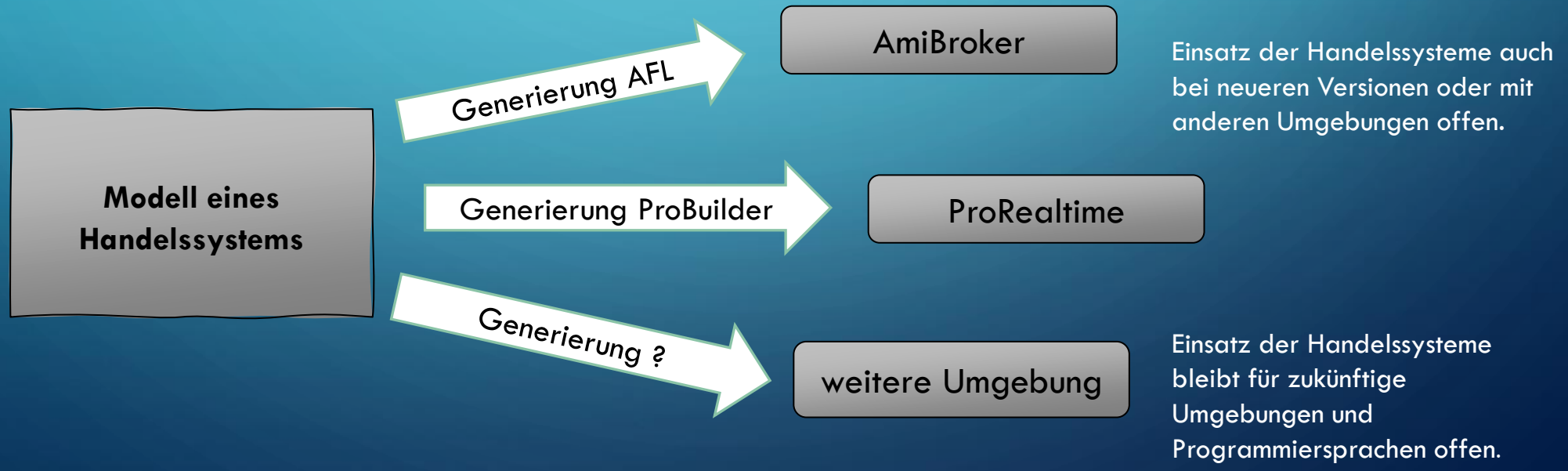
ENTWICKLUNGSPROZESS: VALIDIERUNG DES MODELLS

Durch eine maschinelle Überprüfung (Validierung) des Modells gegenüber <TradeKnowledge/> entspricht das modellierte Handelssystem stets dem abgebildeten Wissen.



ENTWICKLUNGSPROZESS: TRANSFORMATION DES MODELLS IN PROGRAMMCODE

- Aus dem Modell eines Handelssystems lassen sich Programmcodes für unterschiedliche Umgebungen generieren.
- Somit ist das, mit dem Modell abgebildete, Handelssystem in unterschiedlichen Umgebungen einsatz- und austausch- und wiederverwendbar.



LOHNT SICH DIE ANWENDUNG DES MODELLBASIERTEN ENTWICKLUNGSPROZESSES?

Vorteile

- Möglichkeit des Zugriffs und der Verwendung von gesammelten Wissen aus der Allgemeinheit.
- Zielgerichtete Interpretation von Wissen.
- Öffentlichkeit und Erweiterbarkeit.
- Vielfältige Möglichkeiten für den modellbasierten Ansatz.
- Unabhängig von der Programmiersprache und Umgebung.

Nachteile

- Einarbeitung in das Konzept und in die anzuwendenden Tools notwendig.
- Struktur von <Knowledge/> (noch) zu kompliziert. Um neues Wissen abzubilden, ist entsprechendes Spezialwissen oder unterstützende Tools erforderlich.
- Anwendung des Modellierungstools (aus der prototypischen Realisierung) noch viel zu aufwendig und zu kompliziert.
- Für jede Programmiersprache ist die Realisierung eines eigenen Transformators notwendig.

Was denken Sie?

WAS WURDE BIS JETZT ERREICHT?

Realisierung einer prototypischen Implementierung

(Sourcecode und benötigte Dateien werden noch im GIT veröffentlicht)

- Entwicklung der Wissensbeschreibungssprache <Knowledge/>
- Erste Version von <TradeKnowledge/>
- Erste Version eines Modellierungstools
- Code-Transformator für ProBuilder (ProRealtime) und erste Ansätze für Equilla (TradeSignal) und AFL (AmiBroker)
- Alles noch in den Kinderschuhen („Wackelig“ und Anwendung noch viel zu umständlich)

Dokumente zu diesem Projekt:

- TradeKnowledge.pdf
  u.a. Beschreibung von TradeKnowledge
- TradeKnowledge_Anhang.pdf
  u.a. Konzept von Knowledge, durchgehendes Beispiel einer Modellierung und weitere Ideen

WEITERE ZIELE

- Veröffentlichung und Darstellung von TradeKnowledge in Form einer Internetpräsens
- Weitere Entwicklung und Erweiterungen von <Knowledge/>
- Weitere Entwicklung und Erweiterungen von <TradeKnowledge/>
- Stabilisierung und Weiterentwicklung der vorhandenen prototypischen Implementierung
- Vereinfachung des Modellierungsprozesses
- Realisierung weiterer Transformatoren zur Codegenerierung
- Realisierung weiterer Ideen und Ansätze
- Und und und …

WAS WILL ICH?

- <TradeKnowledge/> ist als Freizeitprojekt entstanden und hat sich als MEGA-Projekt herausgestellt.
- Ich bin vom Ansatz und Nutzen überzeugt.
- Einsatz als Community-Version wünschenswert.
- Es soll als profitloses und öffentliches Projekt weiterbetrieben werden.
- Ich suche interessierte Mitstreiter und Unterstützer.