Ein Mehrphasen–Protokoll für wissensbasierte Dienstvermittlung

A. Puder, F. Gudermann und S. Markwitz
{puder,florian,markwitz}@informatik.uni-frankfurt.de

Fachbereich Informatik
Johann Wolfgang Goethe Universität Frankfurt
Postfach 11 19 32
D–60054 Frankfurt

Kurzfassung: In offenen verteilten Umgebungen erlangt die Vermittlung von Diensten eine besondere Bedeutung. Bisherige Ansätze konzentrieren sich jedoch auf die Vermittlung von *Instanzen* dieser Dienste zur Laufzeit. Dabei wird vernachlässigt, wie zur Übersetzungszeit Kenntnis über einen *Diensttyp* erlangt werden kann. Eine bisher vorausgesetzte Lösung dieses Problems sind *a priori* Absprachen aller Teilnehmer einer offenen verteilten Umgebung. Um einen solchen Vorgang zu automatisieren, wird in dieser Arbeit ein wissensbasierter Vermittlungsdienst vorgeschlagen, der sich insbesondere durch eine auf Konzeptgraphen beruhende Dienstbeschreibungsmethodik und einen maschinellen Lernalgorithmus für Import– und Export–Operationen auszeichnet. Charakteristisch für diesen Lösungsansatz ist die Tatsache, daß sowohl für einen Import, als auch einen Export Vorgang, u.U. mehrere Interaktionen mit einem Dienstnutzer und Dienstanbieter notwendig sind. In dieser Arbeit werden die Voraussetzungen für dieses mehrere Phasen umfassende Vermittlungsprotokoll erarbeitet.
Schlüsselwörter: Meta–Trading, Maschinelles Lernen, ODP.

1 Einleitung

Die Rolle eines Dienstvermittlers spielt in einer offenen verteilten Umgebung eine zentrale Rolle. Durch diese Komponente finden Dienstnutzer passende Anbieter und realisieren so erst einen Dienstmarkt, in dem Dienste frei angeboten und genutzt werden können. Eine Interaktion zwischen Anbietern und Nutzern setzt eine präzise Schnittstellendefinition voraus, die häufig durch eine *Interface Definition Language* (IDL) festgelegt wird.

Der Vermittlungsvorgang zur Laufzeit vernachlässigt die wichtige Fragestellung, wie die Programmierer zur Übersetzungszeit ihrer Applikationen Kenntnis über Diensttypen erlangen. Bisherige Ansätze setzten eine *a priori* Absprache aller Beteiligten voraus, durch den globaler Konsens über Diensttypen erreicht wird. Diese Absprachen werden angesichts einer zunehmenden Vielfalt von Diensten immer komplizierter und es ist nahezu unmöglich, ein vollständiges Bild über das aktuell existierende Dienstangebot zu erlangen.

Soll dieser Prozeß automatisiert werden, ist ein Vermittlungsdienst von Nöten, der nicht Instanzen, sondern deren Typen vermittelt. Eine starre, IDL–basierte Dienstbeschreibung ist jedoch nicht geeignet, um a priori Absprachen zwischen den Teilnehmern eines offenen Systems zu vermeiden. Dies kann nur durch eine geeignete semantische Dienstbeschreibung geschehen, die die kognitive Ebene der Teilnehmer unterstützt.

Von einer semantischen Dienstbeschreibung, die zudem einer Erwartungshaltung eines Teilnehmers entspringt, können nicht streng deterministische Regeln für die Vermittlung erwartet werden. In diesem Szenario ist es wegen abweichenden Sichten auf einen Sachverhalt durchaus möglich, daß nur eine geringe Übereinstimmung der Dienstbeschreibungen festzustellen ist.

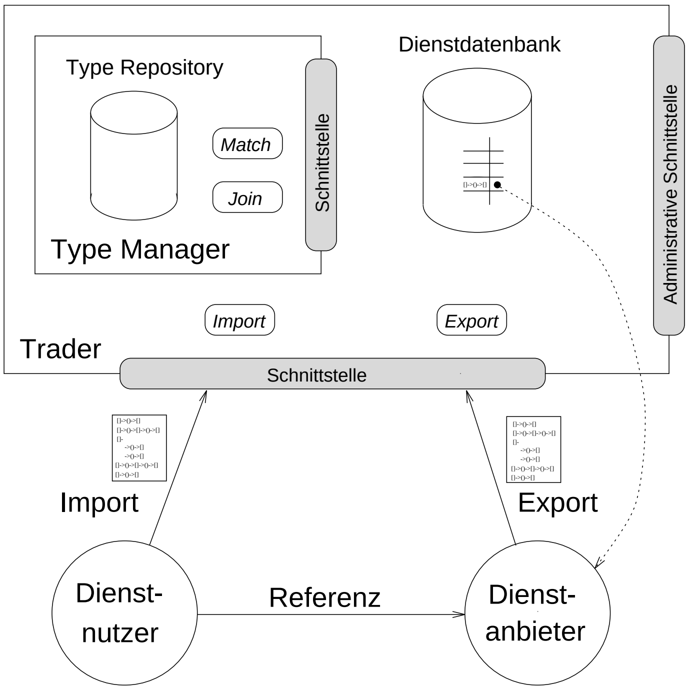

Abbildung 1: Architektur eines Traders.

In dieser Arbeit wird ein Ansatz für die wissensbasierte Vermittlung von Diensten vorgeschlagen. Da sich der Vermittlungsdienst durch eine Lernkomponente auszeichnet, die sukzessive unterschiedliche Sichtweisen der Teilnehmer zu einer Approximation eines Diensttyps erstellt, muß ein spezielles Protokoll entwickelt werden, das den auftretenden Fallunterscheidungen Rechnung trägt.

Diese Arbeit ist wie folgt strukturiert: Abschnitt 2 stellt ein auf dem Objektbegriff bauendes Modell einer offenen verteilten Umgebung vor, unter dem besonderen Gesichtspunkt der Dienstvermittlung. Abschnitt 3 definiert ein abstraktes Modell für die wissensbasierte Dienstvermittlung und der Abschnitt 4 geht auf die Zusammenhänge mit dem Begriff des maschinellen Lernens ein. Abschnitt 5 stellt die einzelnen Phasen des Vermittlungsprotokolls informell dar. Eine Zusammenfassung beschließt diese Arbeit.

2 Dienstvermittlung in offenen verteilten Umgebungen

Das Objektmodell (siehe [2]) ist wegen seiner zentralen Eigenschaft der Kapselung von Daten und Kode als Abstraktion eines verteilten Systems geeignet. Ein offener verteilter Dienstmarkt kann als eine Menge von kooperierenden Objekten angesehen werden. Es ergeben sich zwei Rollen, die Objekten in einer solchen Umgebung zugeordnet werden können, je nachdem ob sie einen Dienst anbieten oder nutzen. Es stellt sich die Frage, wie ein Dienstnutzer Kenntnis über einen passenden Dienstanbieter erhält, um sich an diesen anzubinden. Das Problem entsteht wegen der hohen Dynamik einer offenen verteilten Umgebung (d.h. es kommen neue Dienste hinzu während andere nicht mehr zur Verfügung stehen), wodurch es einem Dienstnutzer unmöglich ist, über die notwendigen globalen Informationen zu verfügen, um diese Dienste in Anspruch zu nehmen.

Das ist die Motivation für sogenannte *Dienstvermittler* (im Englischen *Trader* genannt), die konzeptionell globales Wissen über alle Dienstanbieter verfügen. Ein Objekt nimmt somit die

Rolle des Traders ein, der die dynamische Auswahl von Diensten zur Laufzeit erlaubt (siehe [1, 7, 4, 5, 3]). Folgende Interaktionen zwischen einem Trader und den anderen Objekten in Abhängigkeit ihrer Rollen können durchgeführt werden; im ODP Kontext als *Export* und *Import* bezeichnet (siehe auch Abbildung 1):

Export: Ein Dienstanbieter kann sich bei dem Trader durch eine Dienstbeschreibung bekannt machen, wodurch diese zusammen mit seiner Adresse (d.h. seiner Objektidentität) in einer Dienstdatenbank abgelegt wird. Danach steht dieser Dienst potentiellen Dienstnutzern zur Verfügung.

Import: Ein Dienstnutzer kann seinerseits an den Trader eine Suchanfrage mit der gewünschten Dienstbeschreibung stellen. Der Trader sucht an Hand der Dienstbeschreibung des Dienstnutzers in seiner Datenbank nach einem passenden Dienst und präsentiert dem Dienstnutzer eine Menge von Diensten, die seinen Anforderungen entsprechen.

2.1 Typbeschreibungen in offenen verteilten Umgebungen

Die oben erwähnte Dienstbeschreibung entspricht einem *Typ*, der sich aus der im Objektmodell geforderten Kapselung von Daten und Kode ergibt. Dabei beschreibt ein Objekttyp das nach außenhin sichtbare Verhalten des Objekts. ODP definiert einen Dienst als eine Instanz eines Diensttyps. Jeder Dienst wird durch Schnittstellen und jeder Diensttyp durch einen Schnittstellentyp beschrieben. Die Schnittstellen werden durch die Signaturen der Funktionen beschrieben, die das Verhalten des Objektes definieren. Dies entspricht einer syntaktischen Typdefinition durch eine IDL.

Das im ODP–Kontext beschriebene Trading beschäftigt sich mit der Vermittlung von *Objekten* zur Laufzeit und nicht deren *Typen* zur Übersetzungszeit. Es wird somit eine exakte Kenntnis der Spezifikation eines Dienstes vorausgesetzt. Der Prozeß, durch den sich die Teilnehmer eines verteilten Systems auf die Typspezifikationen verständigen, soll im folgenden *Meta–Trading* genannt werden. Dieser Vorgang wird nicht durch einen herkömmlichen Trader unterstützt und setzt a priori Absprachen zwischen den Teilnehmern des offenen Dienstmarktes voraus. Jeder Teilnehmer hat eine abstrakte Vorstellung über Eigenschaften, die er mit einem Dienst assoziiert. Da üblicherweise die Typspezifikationen a priori festgelegt sind, werden alle abstrakten Intentionen auf eine syntaktische Repräsentation des Typs abgebildet (siehe Abbildung 2).

Für eine Objektinteraktion muß eine bekannte Schnittstellendefinition in Form einer IDL vorausgesetzt werden. Eine syntaktische Typdefinition ist somit im ODP–Kontext für den Trading–Vorgang auf einer Objektebene ausreichend. Ein Mechanismus für das Meta–Trading bedingt jedoch eine semantische Dienstbeschreibung, um die unterschiedlichen Vorstellung der kognitiven Ebene beschreiben zu können. Die verschiedenen abstrakten Intentionen werden dabei auf unterschiedliche syntaktische Artefakte abgebildet, die jedoch semantisch übereinstimmen.

Ziel der semantischen Typspezifikation ist es, den Prozeß des Meta–Tradings zu automatisieren. Die Aufgabe des Meta–Tradings besteht somit darin, die semantische Überlappung (auch *semantische Schnittmenge* genannt) festzustellen. Der Vorteil liegt darin, daß die Typspezifikationen nicht mehr a priori festgelegt werden müssen. Um den Prozeß des Meta–Tradings zu automatisieren, müssen jedoch geeignete Beschreibungssprachen und Mechanismen, die die Abbildung der einzelnen Typspezifikationen auf einen Typ, entwickelt werden (siehe [10] und [9]). Für die Aufgabe des Meta–Tradings lassen sich folgende Annahmen an eine offene verteilte Umgebung und eine Typbeschreibungsmethodik identifizieren:

Open World Assumption: Die Teilnehmer in einem offenen verteilten System besitzen keine globale Sicht auf die im Typraum definierten Dienste. Aufgrund der hohen Dynamik des Dienstmarktes ist ein konsistenter Schnappschuß auch nicht möglich.

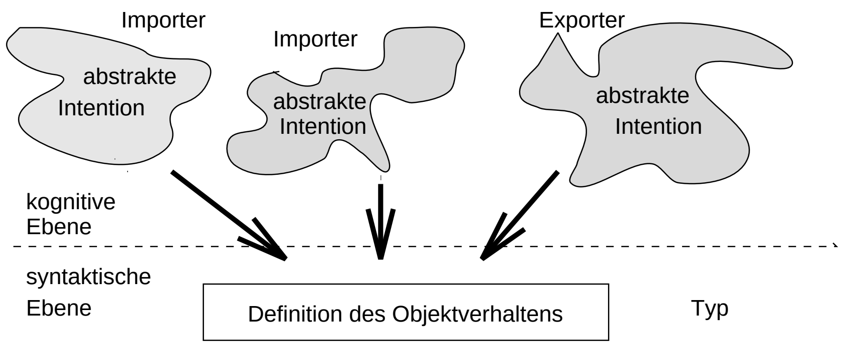

Abbildung 2: Zusammenhang zwischen Dienstbeschreibungen auf der kognitiven und syntaktischen Ebene.

Semantische Typdefinition: Um a priori Absprachen bei den Typdefinitionen zu vermeiden und den Prozeß des Meta–Tradings zu automatisieren, muß eine semantische Typbeschreibungsmethodik vorausgesetzt werden.

Polymorphie: Durch die Polymorphie werden Spezialisierungsrelationen zwischen Typen definiert. Die Menge der potentiellen Dienste, die durch einen Typ erbracht werden, erhöht somit das quantitative Dienstangebot.

Unscharfe Interpretation: Aus Gründen der Entscheidbarkeit wird eine unscharfe Interpretation der Semantik eines Typs verlangt (siehe [8]). Dies erlaubt eine größere Ausdruckskraft; evtl. auftretende „Fehler“ müssen jedoch von dem Trading Protokoll aufgefangen werden.

2.2 Überblick über den wissensbasierten Trader

Beim wissensbasierten Trader werden semantische Diensttypspezifikationen auf Basis der *Konzeptgraphen* verwendet. Konzeptgraphen entstammen dem Gebiet der Wissensrepräsentation und sind von ihrem Abstraktionsniveau der natürlichen Sprache verwandt (siehe [12]) und werden im Rahmen des ANSI Technical Committee X3T2 standardisiert. Im mathematischen Sinne ist ein Konzeptgraph ein gerichteter, bipartiter Graph, der aus *Konzept-* und *Relationsknoten* besteht. Je zwei Konzepte werden durch eine Relation verbunden.

Ein Konzept stellt ein Ding unserer Vorstellung sowohl abstrakter als auch physikalischer Natur dar. In bezug auf Dienstbeschreibungen können beispielsweise Begriffe wie *Drucker*, *Auflösung* oder *Geschwindigkeit* durch jeweilige Konzepte beschrieben werden. Relationsknoten dienen dazu, zwei oder mehrere Konzepte in einen semantischen Bezug zu stellen. Folgender Konzeptgraph beschreibt die Programmiersprache *C++*[1]. Konzepte werden dabei in eckige und Relationen in runde Klammern geschrieben:

```
[OO-SPRACHE:{"C++"}] -
          -> (OBERMENGE-VON) -> [PROGRAMMIERSPRACHE:{"C"}],
          -> (KAPSELT) -
                   -> [ZUSTAND],
                   -> [VERHALTEN].,
          -> (UNTERSTUETZT) -> [KLASSEN] -
```

[1]Auf eine detaillierte Beschreibung des „Dienstes“ *C++* (d.h. ob es sich beispielsweise um eine Produktbeschreibung oder den Sprachstandard handelt) wurde verzichtet.

```
-> (ENTHALTEN) -> [METHODEN],
-> (ENTHALTEN) -> [ZUSTAND].
.
```

Die informelle Semantik, die sich hinter diesem Konzeptgraphen verbirgt, lautet: *C++ ist eine objektorientierte Programmiersprache und eine Obermenge der Programmiersprache C. Sie kapselt Zustand und Verhalten und unterstützt Klassen, die ihrerseits Methoden und einen Zustand enthalten.*

Die Namen *C++* und *C* in diesem Beispiel sind Instanzen der Konzepttypen `OO-SPRACHE` und `PROGRAMMIERSPRACHE`. Ein derartiger Konzeptgraph korrespondiert mit einer Sicht auf eine Dienstbeschreibung. Sowohl der Export– als auch der Import–Vorgang verwenden Konzeptgraphen, um eine Sicht zu spezifizieren. Der wissensbasierte Trader speichert die Dienstbeschreibungen in seinem *Type Repository* und implementiert durch die folgenden beiden Operationen eine Lern– bzw. Abfragekomponente (siehe auch Abbildung 1):

Join: Konstruiert aus zwei Diensttypbeschreibungen in Form von Konzeptgraphen eine detailliertere Diensttypbeschreibung wiederum in Form eines Konzeptgraphens.

Match: Bildet eine Diensttypbeschreibung auf alle in der Wissensbasis gespeicherten Diensttypen in Form von Konzeptgraphen ab. Dabei wird ein Ähnlichkeitsmaß berechnet, das die Qualität eines Treffers beschreibt und durch die Größe der semantischen Schnittmenge zweier Konzeptgraphen definiert ist.

Eine ausführliche Darstellung der Dienstbeschreibung durch Konzeptgraphen findet sich in [10] und [9].

3 Abstraktes Tradingmodell

Das abstrakte Tradingmodell beschreibt Begriffe der kognitiven Ebene und ist notwendig, um das Meta–Trading zu automatisieren, d.h. von der kognitiven Ebene auf die syntaktische Ebene zu verlagern. An Hand der Definitionen dieses Modells können dann die Folgen, die diese Verlagerung für das Trading–Protokoll haben, beschrieben werden.

Definition: Sicht Eine *Sicht* ist die semantische Typspezifikation eines Dienstes, die die Präferenzen dessen berücksichtigt, der den Dienst beschreibt. Diese Präferenzen basieren auf seinem Wissen über den Dienst und den Anforderungen, die er im Rahmen seiner Anwendung an diesen stellt.

Die Vorstellung eines Teilnehmers über einen abstrakten Diensttyp auf der kognitiven Ebene entspricht seiner individuellen Sicht und wird beim wissensbasiertem Trader durch einen Konzeptgraphen repräsentiert. Es können mehrere Sichten auf einen Dienst existieren. Ein Experte, der einen Dienst genau kennt, wird diesen subjektiv gesehen präziser und detaillierter beschreiben, als ein Anwender, der nicht über dieses Wissen verfügt und folglich den Dienst allgemeiner beschreibt. Er hat aber eine Vorstellung davon, was der gesuchte Dienst leisten soll und beschreibt den Dienst aus dieser Sicht. Da keine Einschränkungen bei den Möglichkeiten einen Dienst zu beschreiben gemacht werden sollen, wird jeder Teilnehmer eine individuelle Sicht eines Dienstes haben. Dies hat für ein Trading–Protokoll die Folge, daß der Dienstexport kein einphasiger Vorgang sein kann, sondern mehrere Phasen, in denen die verschiedenen Sichten berücksichtigt werden müssen, umfaßt.

Würden sich alle Sichten eines Dienstes völlig unterscheiden, dann wäre es unmöglich eine Abbildung zu definieren, die den Grad Übereinstimmung mißt und damit ein Entscheidungskriterium angibt, wann zwei Sichten denselben Dienst beschreiben. Dieses Ähnlichkeitsmaß muß aber vom Trader berechnet werden können, damit er eine Suchanfrage, die auf der Sicht eines Anwenders basiert, mit der Sicht des Anbieters zur Überdeckung bringen kann.

Definition: Kernbereich Ein *Kernbereich* definiert die grundlegenden funktionalen und technischen Eigenschaften eines Dienstes. Er ergibt sich aus der semantischen Schnittmenge „sinnvoller" Sichten eines Dienstes. Es ist Aufgabe eines Dienstanbieters zu entscheiden, wann eine Sicht sinnvoll ist.

Diese Eigenschaften müssen sich bei der Konstruktion einer allgemeinen Sicht abheben und bilden die Basis für ein Ähnlichkeitsmaß. Eine Sicht muß nicht alle Kernbereiche enthalten. Trotzdem muß sie einen Teil der Kernbereiche beschreiben, damit der Trader eine genügend große Ähnlichkeit feststellen kann. Die Forderung nach Kernbereichen ist Voraussetzung für die Polymorphie zwischen Sichten.

Definition: Polymorphie für Sichten *Polymorphie* ist die Ähnlichkeit zweier Sichten, die durch ein Ähnlichkeitsmaß, daß an Hand der semantischen Schnittmenge dieser Sichten berechnet wird, angegeben werden kann.

Es kann nur die Ähnlichkeit zweier Sichten bestimmt werden und ist damit im Unterschied zur klassischen Subtyprelation symmetrisch. Somit ist die Aufgabe des Dienstanbieters, der ein Experte seines Dienstes ist, vorgegeben. Er muß bei seiner initialen Beschreibung seines Dienstes Teile des Kernbereichs identifizieren. Diese initiale Beschreibung sollte eine „gute" Beschreibung des Dienstes sein, aber sie ist sicher unvollständig.

Eine Konsequenz aus der Polymorphie von semantischen Beschreibungen eines Dienstes betrifft den Anwender. Ist seine Sicht eines Dienstes zu allgemein, so wird der Trader bei einer Suche an Hand dieser Beschreibung sehr viele Dienste als potentielle Treffer identifizieren. Der Anwender ist dann gefordert, ebenfalls seine Sicht zu verfeinern. Das Trading–Protokoll muß dies unterstützten, weshalb auch der Dienstimport mehrere Phasen umfassen kann.

Definition: Widersprüche zwischen Sichten Ein *Widerspruch* zwischen zwei Sichten liegt dann vor, wenn die eine Sicht eine Eigenschaft der anderen Sicht explizit negiert.

Da semantische Beschreibungen eines Dienstes mehrdeutig sind, können Widersprüche zwischen den verschiedenen Beschreibungen auftreten. Der Anbieter ist der Experte bzgl. seines Dienstes und hat deshalb ein großes Wissen über den von ihm angebotenen Dienst. Daher kann nur er entscheiden, ob eine andere Sicht eines Dienstes korrekt ist und gegebenenfalls Widersprüche zwischen den verschiedenen Sichten auflösen. Diese Aufgabe kann nicht vom Trader übernommen werden, da diese nicht automatisierbar ist. Der Trader kann aber Widersprüche erkennen. Dabei muß er zwei Fälle unterscheiden. Beschreibt ein Anbieter während des Export Vorgangs einen Dienst aus mehreren Sichten, so kann der Trader diesen Anbieter auf Widersprüche hinweisen und ein Auflösung anfordern. Treten Widersprüche zwischen einer Anbietersicht und einer Anwendersicht während einer Import Operation auf, so kann von einer unterschiedlichen Dienstbeschreibung ausgegangen werden.

Definition: Browser Der *Browser* ermöglicht es, alle Diensttypspezifikationen die der Trader in seiner Datenbank verwaltet, durchzusehen. Er wird erst verwendet, wenn eine Suchanfrage kein Ergebnis geliefert hat oder auf Grund von Fehlern die Suchanfrage nicht den gewünschten Dienst geliefert hat.

Jede Anfrage (d.h. Import) eines Anwenders kann als ein Test angesehen werden, bei dem geprüft wird, ob diese Sicht bereits durch eine generelle Sicht abgedeckt wird. Ist dies nicht der Fall, müssen Mechanismen entwickelt werden, um festzustellen welchen Dienst diese neue Sicht beschreibt. Der Browser ermöglicht es dem Anwender alle Beschreibungen von Diensten durchzusehen und zu entscheiden, welcher Dienst durch seine Sicht beschrieben wurde. Dazu muß die Menge der Typspezifikationen strukturiert sein, um ein effizientes Navigieren aller Dienstangebote zu ermöglichen (beispielsweise durch eine Diensttyphierarchie, die durch eine Spezialisierungsrelation definiert ist).

Aus den obigen Diskussionen folgt, daß ein einfaches Protokoll wie es bei ODP–basierten Tradern verwendet wird, nicht ausreicht. Im abstrakten Modell lassen sich bereits mehrere Phasen des wissensbasierten Tradings identifizieren. Beim wissensbasierten Trading soll der Prozeß des Meta–Tradings durch den Trader unterstützt werden. Da es sich um semantische Dienstbeschreibungen handelt, bietet es sich an, Mechanismen der künstlichen Intelligenz und insbesondere des maschinellen Lernens auf ihre Anwendbarkeit beim Trading zu untersuchen.

4 Maschinelles Lernen

Bevor ein spezieller Lernvorgang präsentiert wird, muß eine Definition des Begriffs *Lernen* gegeben werden, die einem Lernprozeß zu Grunde gelegt wird. Dabei handelt es sich nicht um eine formale Definition im mathematischen Sinne, sondern um eine informelle Begriffsbestimmung, die beschreiben soll, wie dieser Begriff im folgenden verstanden werden soll (siehe [11]).

Definition: Allgemeiner Lernbegriff *Lernen* ist jede Veränderung eines Systems, die es diesem erlaubt, eine Aufgabe bei der Wiederholung derselben Aufgabe (oder einer Aufgabe derselben Art) besser zu lösen.

Eine Anwendung dieser Definition auf die vorher beschriebenen Anforderungen führt zu folgender Definition des Lernens beim wissensbasierten Trading:

Definition: Lernen von Diensten Lernen ist die Konstruktion einer generellen Sicht eines Dienstes mit Hilfe eines Inferenzmechanismus, der diese aus den verschiedenen individuellen Sichten erzeugt, so daß der Trader bei einer erneuten Anfrage den gesuchten Dienst identifizieren kann.

Diese Definition beschreibt die Anwendung des maschinellen Lernens auf den Prozeß des Meta–Tradings und macht keine Aussagen über den Inferenzmechanismus. In der künstlichen Intelligenz sind einige Verfahren bekannt, die maschinelles Lernen realisieren (eine Übersicht verschiedener Lernverfahren findet sich in [6]). Für den wissensbasierten Trader wurde ein Lernverfahren verwandt, das auf der Vermittlung von Beispielen und Gegenbeispielen durch Konzeptgraphen beruht.

Beim Lernen durch Beispiele werden dem System eine Menge von Beispielen und Gegenbeispielen eines Konzeptes präsentiert, um daraus eine generelle Beschreibung zu generieren. Dabei hat das System vor dem Lernschritt kein Wissen über das Konzept. Die Beispiele können alle auf einmal oder inkrementell in mehreren Schritten präsentiert werden. Beim inkrementellen Prozeß generiert das System in jedem Schritt eine Beschreibung, die mit jedem weiteren Beispiel verfeinert wird.
Bei diesem Lernszenario lassen sich zwei Rollen identifizieren:

- Lehrer
- Lerner

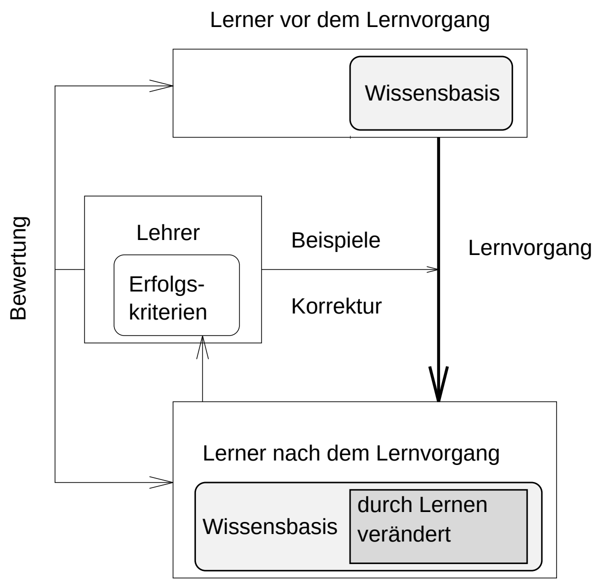

Abbildung 3: Lernmodell.

Der Lehrer verfügt über spezielles Wissen eines Gebiets, welches dem Lerner vermittelt werden soll. Der Lernvorgang wird vom Lehrer gesteuert und dieser nimmt gegebenenfalls Korrekturen vor, wenn Fehler auftreten oder der Lerner Widersprüche innerhalb der gegebenen Beispiele erkennt. Außerdem stellt er die Erfolgskriterien auf, mit denen er feststellt, wann der Lernvorgang basierend auf diesen Beispielen abgeschlossen ist. Die Erfolgskriterien beinhalten ein Lernziel und im allgemeinen auch Gütekriterien, die eine Bewertung des Resultates des Lernvorgangs erlauben. Das Lernziel hängt davon ab, was gelernt werden soll. Bei der Konstruktion einer allgemeinen Beschreibung eines Konzeptes an Hand von Beispielen, könnte das Lernziel z.B. fordern, daß sich diese gegenüber anderen allgemeinen Beschreibungen genügend abgrenzt. Eine mögliche mehrfache Interaktion zwischen Lehrer und Lerner ist die Folge (siehe Abbildung 3).

5 Wissensbasierte Import– und Export–Operationen

Der Prozeß des Meta–Tradings soll beim wissensbasierten Trading automatisiert werden. Jede Sicht wird durch einen Konzeptgraphen repräsentiert und der Trader hat die Aufgabe diese verschiedenen Diensttypspezifikationen zu einer generellen Diensttypspezifikation zu verbinden.

Das Lernmodell kommt bei den verschiedenen Interaktionen zwischen den Objekten des Tradermodells zur Anwendung. Dabei muß es den Aufgaben, die durch das abstrakte Modell beschrieben werden, entsprechend angepaßt werden. Eine Folge davon ist ein mehrphasiges Protokoll, daß die einzelnen Phasen des Lernmodells im abstrakten Modell realisiert.

Jedes der im Tradermodell beschriebene Objekt übernimmt beim wissensbasierten Trading weitere Rollen. Diese sind die im Lernszenario beschriebenen Rollen des Lehrers und des Lerners. Wann welches Objekt eine bestimmte Rolle annimmt, wird bei der Beschreibung des gesamten Tradingszenarios deutlich. Da Dienstanbieter, Trader und Dienstnutzer als Objekte in dem verteilten System angesehen werden, die über eine geeignete Infrastruktur kooperieren, beschreibt folgendes wissensbasiertes Tradingszenario die einzelnen Phasen des Protokolls.

Der Dienstanbieter tritt beim Dienstexport gegenüber dem Trader zunächst in der Rolle des Lehrers auf. Er präsentiert ihm sukzessive eine Menge von Dienstbeschreibungen, die verschiedene Sichten des Dienstes repräsentieren. Zur Abgrenzung gegenüber den Beschreibungen anderer Dienste können Negationen von Teilbeschreibungen aus diesen verwendet werden (d.h. Gegenbeispiele). Der Trader hat hier die Rolle des Lerners. Er konstruiert an Hand seiner Mechanismen eine generelle Sicht dieses Dienstes. Erkennt er innerhalb der Sichten Inkonsistenzen, so teilt er dies dem Lehrer mit, der daraufhin eine Korrektur vornimmt. Beendet der Dienstanbieter den Lernprozeß, so wendet der Trader die generelle Beschreibung eines Dienstes auf alle anderen gelernten Dienstbeschreibungen an. Er berechnet das Ähnlichkeitsmaß und präsentiert dem Dienstanbieter eine Liste von Diensten, die dem neu gelernten Dienst ähnlich sind. Hier nimmt der Trader die Rolle des Lehrers und der Dienstanbieter die Rolle des Lerners an. An Hand der vom Trader präsentierten Beispiele muß er nun entscheiden, ob er seine Beschreibung gegen andere Dienste abgrenzen möchte, oder ob seine Beschreibung nur eine andere Sicht eines dieser Dienste darstellt.

Das Ähnlichkeitsmaß wird verwendet, um den Lernerfolg zu messen. Dem Trader muß ein Schwellenwert angegeben werden, der festlegt, wann die Beschreibungen zweier Dienste zu ähnlich sind. Für den Dienstnutzer beschreibt dieses Ähnlichkeitsmaß den Grad der Polymorphie.

Der Dienstnutzer stellt eine Anfrage in Form einer Dienstbeschreibung an den Trader. Dieser versucht, einen diesen Anforderungen genügenden Dienst zu finden. Eine Liste der möglichen Dienste, die der Anforderung entsprechen, werden dem Dienstnutzer vom Trader präsentiert. Der Trader tritt hier in der Rolle des Lerners auf und der Dienstnutzer in der Rolle des Lehrers. Der Dienstnutzer kann nun einen Dienst auswählen oder seine Anfrage modifizieren. Dabei stehen ihm die Beschreibungen der Dienste, die der Trader gefunden hat, zur Verfügung. Hat der Dienstnutzer nach mehrfacher Modifikation seiner Anfrage den gesuchten Dienst gefunden, so bekommt der Dienstanbieter die Dienstbeschreibung vom Trader präsentiert, damit dieser diese neue Sicht seines Dienstes berücksichtigen kann. Hier hat der Trader die Rolle des Lehrers während der Dienstanbieter als Lerner auftritt. Kann der Trader keinen passenden Dienst zu der gegebenen Dienstbeschreibung finden, so kann der Dienstnutzer den Browser verwenden, um die gesamten Dienstbeschreibungen durchzusehen und einen passenden zu finden.

Jeder Dienstanbieter tritt in der Lernphase gegenüber dem Trader als Lehrer auf. Da ein Dienstanbieter zunächst nicht die Beschreibungen der Dienste der anderen Dienstanbieter kennt, bekommen alle Dienstanbieter, die eine ähnliche Sicht eines Dienstes beschrieben haben, die jeweils anderen Sichten dieses Dienstes vom Trader präsentiert.

Die in diesem Lernszenario beschriebenen mehrfachen Interaktionen zwischen den einzelnen Objekten müssen durch ein Protokoll realisiert werden, das im Gegensatz zum traditionellen Trading mehrere Phasen umfaßt. Die einzelnen Phasen können direkt aus dem Lernszenario abgeleitet werden.

6 Zusammenfassung

Traditionelle Vermittlungsdienste konzentrieren sich auf die Vermittlung von Instanzen zur Laufzeit, vernachlässigen aber den wichtigen Bereich des Meta–Tradings. Um diesen Vorgang zu automatisieren, und dadurch die a priori Kenntnis eines Diensttyps zu vermeiden, ist eine semantische Dienstbeschreibung notwendig, die die kognitive Ebene eines Menschen unterstützt.

Der in dieser Arbeit vorgestellte wissensbasierte Vermittlungsansatz verwendet mit Konzeptgraphen eine Wissensrepräsentationstechnik, die in ihrem Abstraktionsgrad der natürlichen Sprache verwandt ist. Der Vermittlungsdienst beinhaltet einen Lernalgorithmus, durch den er die unterschiedlichen Sichten der Teilnehmer auf einen Dienst erlernen kann. Der Freiheitsgrad in der Beschreibungsmächtigkeit führt jedoch zu Fehlern, da der Vermittlungsdienst mit unscharfem Wissen arbeitet. Das hat unmittelbare Konsequenzen für die Import– und Export–Operationen, die dadurch sich in mehrere Phasen gliedern.

Die vorgestellten Konzepte sind von den Autoren implementiert worden. Der wissensbasierte

Vermittlungsdienst verarbeitet Dienstbeschreibungen in Form von Konzeptgraphen, die sich von ihrer Syntax an dem Beispiel aus Abschnitt 2.2 halten. Ein graphisches Werkzeug erleichtert die Eingabe derartiger Dienstbeschreibungen und vermindert durch seine Benutzerführung die Komplexität des mehrphasigen Vermittlungsprotokolls. Die vollständigen Quelltexte auf Basis von C++ und Tcl/Tk sind frei verfügbar und können von den Autoren bezogen werden.

Literatur

[1] M. Bearman. ODP–Trader. In *Proceedings of the IFIP TC6/WG6.1 International Conference on Open Distributed Processing*, pages 37–51, Berlin, Germany, 13–16 September 1993. North–Holland.

[2] G. Booch. *Object Oriented Design with Applications.* Benjamin Cummings Publishing Company, Inc, Redwood City, California, 1991.

[3] ODP Trading Function, ITU/ISO Committee Draft Standard ISO/IEC DIS13235 Rec. X.9tr, May 1995.

[4] E. Kovacs and S. Wirag. Trading and Distributed Application Management: An Integrated Approach. In *Proceedings of the Fifth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, Toulouse, 1994.

[5] L. Kutvonen and P. Kuvonen. Broadening the User Environment with Implicit Trading. In *Proceedings of the IFIP TC6/WG6.1 International Conference on Open Distributed Processing*, pages 129–140, Berlin, Germany, 13–16 September 1993. North–Holland.

[6] R.S. Michalski. Learning strategies and automated knowledge acquisition. In *Computational Model of Learning.* Springer Verlag, 1987.

[7] Object Management Group (OMG), The Common Object Request Broker: Architecure and Specification, Revision 1.1, December 1991.

[8] A. Puder. A Declarative Extension of IDL–based Type Definitions within Open Distributed Environments. In *International Conference on Object Oriented Information Systems (OOIS)*, South Bank University, London, 19–21 December 1994. Springer Verlag.

[9] A. Puder, S. Markwitz, and F. Gudermann. Service Trading Using Conceptual Structures. In *3rd International Conference on Conceptual Structures (ICCS'95)*, Santa Cruz, University of California, 14–18 August 1995. Springer Verlag.

[10] A. Puder, S. Markwitz, F. Gudermann, and K. Geihs. AI–based Trading in Open Distributed Environments. In *3rd International IFIP TC6 Conference on Open Distributed Processing (ICODP'95)*, Brisbane, Australia, 20–24 February 1995. Chapman and Hall.

[11] H.A. Simon. Why should machines learn? In Michalski, Carbonell and Mitchell, editor, *Machine Lerning: An Artifical Intelligence Approach*, volume 1, pages 25–37, Palo Alto, 1983.

[12] J.F. Sowa. *Conceptual Structures, information processing mind and machine.* Addison–Wesley Publishing Company, 1984.