Ausarbeitung

Multiagentensysteme in der Informationssuche

im Rahmen des Seminars „Ausgewählte Themen aus Agentensysteme“

Dariusz Kordonski

Themensteller und Betreuer: Dr. Dietmar Lammers
Institut für Informatik

Inhaltsverzeichnis

1 Traditionelle Information Retrieval im Web

1.1 Information Retrieval und das Internet

Information Retrieval (Informationswiedergewinnung, Informationsbeschaffung, Informationsrückgewinnung) ist eine Domäne der Informatik, die sich im Prinzip damit beschäftigt, die Nutzende mit Informationen zu versehen, wobei dieser Prozess computergestützt wird. Der Schwerpunkt von *Information Retrieval* (IR) ist daher die Suche nach den notwendigen Informationen.

Seit mehreren Jahren steht das Internet im Mittelpunkt der Interessen der IR. Das *World Wide Web* kann als eine riesige, ungeordnete, aber allgegenwärtige Datenbank gesehen werden [Bae99]. Es gibt eine Reihe von Problemen, die effektive Suche im Web erschweren. Diese liegen entweder an den Daten selbst oder an der Interaktion des Benutzers mit einem Suchsystem. Die Probleme, die sich auf Daten beziehen, sind vor allem [Bae99]:

- verteilte Daten
- größer Anteil von wandelbaren Daten
- riesige Ausmaße
- unstrukturierte und überflüssige Daten
- niedrige Qualität von Daten
- heterogene Daten (hinsichtlich des Formats)

1.2 Die grundlegenden Methoden und Technologien der IR

1.2.1 Suchmethoden

Es werden hier vor allem diejenige Technologien dargestellt, die sich mit Beschaffung der Textinformation beschäftigen. Die Suche anderer Ressourcen (wie z.B. Multimedia) bleibt immer noch unterentwickelt und greift oft auf die Erzielungen der traditionellen IR zurück [Fer03]. Die wichtigsten IR-Techniken, die im Web angewandt werden, sind Suchmaschinen (*search engines*), Webkataloge (*web directories*) und Hyperlink-Suchen [Bae99].

Suchmaschinen sind zurzeit das meist verbreitete Tool, die im Web zur Informationssuche benutzt wird. Die Handelnsweise dieser Systeme ist im Prinzip einfach: sie benutzen die so genannte Web-Roboter (auch u.a. als *spiders*, *web-wanderers* oder *web-crawlers* bekannt), um den Gehalt der Dokumente im Internet ständig durchzukämmen und in eigener Datenbank, in Form eines Schlüsselwortvektors, zu speichern. Die Anfragen der Benutzer werden daher sofort bearbeitet, indem das System auf seine Datenbank zurückgreift.

Die Webkataloge klassifizieren ausgewählte Web-Dokumente aufgrund ihres Themas unter bestimmten Kategorien. Häufig werden diese zwei Lösungen als ein komplementäres Service angeboten (z.B. *Google*, *Yahoo!*, *AltaVista*).

Die Suche im Hyperlink-Ansatz berücksichtigt die Graphstruktur des Web, wo virtuell sämtliche Dokumente mit anderen durch Links verbunden sind. Zu diesen, noch nicht verbreiteten Techniken zählen sog. Web-Query-Sprachen und die dynamische Suche. Auch verschiedene Agentensuchanwendungen benutzen diesen Ansatz.

1.2.2 Darstellungsmethoden

Zur Vorstellung des Dokumentes auf solche Weise, dass es in Hinsicht auf konkrete Anfrage bearbeiten werden kann, braucht man ein bestimmtes Modell. Eines der populärsten und meist benutzten solchen Modelle ist das Vektorraummodell. Dieses Modell stellt einen Dokument als

ein n-dimensionaler Vektor der Schlüsselwörter dar, wobei jedem Wort im Dokument eine Gewichtung zugeschrieben ist. Auch die Anfrage lässt sich als solche Struktur darstellen. Deswegen kann die Übereinstimmung eines bestimmten Dokumentes mit einer Anfrage mathematisch berechnet werden, als eine Zahl zwischen 0 und 1. Einer der bedeutendsten Probleme dabei ist die entsprechende Gewichtung der Schlüsselwörter. Am häufigsten benutzt man eine der Variationen der so genannten *term-frequency-inverted document-frequency*-Gewichtung (*tf-idf*-Gewichtung). Diese Methode zieht nicht nur in Betracht, wie häufig das Stichwort in einem bestimmten Dokument erscheint, sondern auch, wie einmalig ist es in der ganzen Sammlung, da je öfter das Wort in der Sammlung vorkommt, desto geringere seine Bedeutung in jedem einzigen Dokument dieser Kollektion [Bae99].

1.2.3 Bewertungsmethoden

Ein wichtiger Teil von IR ist die Messung der Wirksamkeit der Suchsysteme. Es gibt zwei grundlegende, allgemein verwendete Werte, die quantitative Bewertung der Suchleistung ermöglichen: Präzision (*precision*) und Vollständigkeit (*recall*) [Fer03, Bae99]. Als Präzision bezeichnen wir den Anteil der für eine Abfrage relevanten Dokumente unter den allen für diese Abfrage gefundenen Dokumenten. Unter Vollständigkeit hingegen versteht man den Anteil der für eine Abfrage relevanten Dokumente in der ganzen Kollektion, die auf diese Abfrage zurückgegeben wurden. Diese beide Größen stehen im engen Zusammenhang. Zur grafischen Veranschaulichung der Leistung verschiedener Algorithmen und Suchsysteme werden spezielle *precision-recall*-Diagramme konstruiert, die die Präzision auf 11 Standard-Vollständigkeitsniveaus abbilden.

1.3 Informationsüberfluss

Das riesige Wachstum des Internets in den letzten Jahren hat eine wahre Informationsrevolution verursacht. Der damit verbundene Problem des Informationsüberflusses wurde jedoch inzwischen zu einer der größten Anforderungen der modernen Informatik, besonders im Bereich der IR. Im letzten Jahr wurde die Größe des Web grob für z 11,5 Milliarden Web-Seiten geschätzt [Gul05]. Die *search engines*, wie z.B. berühmte *Google* oder *Yahoo!*, reichen schon nicht, um effektiv wertvolle Information im Web auszusuchen. Solche Suchmaschinen indizieren nur bis za. 8 Milliarden URLs[1]. Darüber hinaus, nur knapp 29% des ganzen Web wird durch alle vier größten *search engines* indiziert [Gul05]. Es fehlt Skalierbarkeit, damit sie mit vernünftigen Rechnungs- und Speicherressourcen das Wachstum des Web erfolgreich folgen könnten. Viele relevante Dokumente werden nicht indiziert, was zu niedriger *recall* führt. Außerdem, je mehr Seiten ein Suchsystem indiziert, desto seltener sind seine Web-Roboter in der Lage, die schon abgelegten Seiten wiederzubesuchen um die Datenbank zu aktualisieren. Die Aktualität der Ergebnisse lässt also viel zu wünschen übrig. Schließlich fehlen den Suchmaschinen effektive Bewertungsmechanismen, was zu niedrigem Niveau der relevanten Seiten in der Sammlungen der Ergebnisse führt, die jeweils riesige Größe dieser Ergebnisse zu verschweigen. Zusammenfassend lässt sich sagen, dass die Suchenden immer dazu gezwungen werden, eine Menge von irrelevanten und veralteten Links „durchzustöbern“, um an die benötigten Informationen zu gelangen. Diese Situation wird sehr bildlich durch ein Zitat dargestellt: *Zusammengefasst führt das Web zu der paradoxen Situation, dass der Einzelne jetzt unbeschränkten Zugang zu unendlich vielen Informationen hat, aber die meiste Zeit verbringt, das Gesuchte zu finden* [Cag98, S. 56].

[1] http://www.google.de/intl/de/why_use.html (Stand 26.12.2005)

2 Agenten in der Informationssuche

2.1 Was ist ein Agent?

Eine allgemeine Definition des Termins „Agent“ lautet: Eine Person oder Sache, die in der Lage oder ermächtigt ist, im Auftrag Dritter zu handeln [Gur70]. Schon bei dieser Definition ergeben sich die grundlegende Ideen, auf deren sich die Agententechnologie entwickelt hat. Es muss sich also selbstverständlich um ein Computerprogramm handeln, der sich aber ziemlich vom üblichen Software unterscheidet. Ein Computeragent soll nämlich etwas für eine andere Person, aber auch einen anderen Programm oder sogar Informationssystem (den Dritten) erledigen können, und auch, um das „etwas“ zu erledigen, soll er imstande sein, autonom zu handeln. Im Kontext der Information Retrieval geht es um Entlastung des Anwenders bei der Auffindung der nötigen Informationen. Das bedeutet Zeit- und Kostenersparnis. In nicht zu weiter Zukunft wird das wahrscheinlich überhaupt die einzige Möglichkeit für Menschen sein, etwas im Internet ausfindig zu machen. In Hinsicht auf den Bereich Suche scheinen hingegen folgende Eigenschaften der Agenten besonders wichtig:

- Autonomie – der Agent muss imstande sein, selbst die Entscheidungen zu treffen, welche Informationen für den Benutzer relevant sind
- Intelligenz – der Agent soll das Verhalten des Anwenders im Suchprozess gewissermaßen nachahmen. Dazu müssen unbedingt die Techniken aus dem KI-Bereich, wie z.B. Wissensdarstellung, Schlussfolgerung oder Lernen, angewandt werden.
- Mobilität – wenn entsprechende Standards weltweit verbreitet sind, die die unbeschränkte Verschiebung der Agenten zwischen Informationsquellen (verschiedene Server im Internet) ermöglichen, wird die Leistung der darauf zurückgreifenden Suchsystemen durch lokalen Zugriff auf Informationen erheblich gesteigert [Klsch99].

2.2 Software-Agenten im Bereich Suchen

Heutzutage sind schon viele Agentenanwendungen, die den Anwender bei der Suche unterstützen, breit verwendet. Die wichtigste Anwendungen in diesem Bereich kann man in folgenden Punkten zusammenfassen [Cag98, Klsch99]:

- Web-Roboter
- Informationssuchagenten
- Informationsfilteragenten
- kollaborative Agenten – sie klassifizieren den Anwender in einer bestimmten Gruppe, oder Gruppen, und, aufgrund dieser Klassifikation, passen die Suche entsprechend an, gegebenenfalls filtern zusätzlich die geschaffte Informationen. Der einfachste Beispiel stellen die „Kunde, der schon gekauft hat“-Empfehlungen, die bei vielen E-Markts zugänglich sind [Klsch90]

Diese Agenten tragen einen gewissen Grad von Autonomie und Intelligenz im Vergleich zu der traditionellen IR-Technologien, was sicherlich die Qualität der Suchoperationen in meisten Fällen steigert und den Anwender entlastet. Informationsagenten beruhen sich aber sehr oft an den traditionellen Methoden (z.B. benutzen die Suchmaschinen im Hintergrund), was automatisch ihre Möglichkeiten, neue Qualität in den Suchprozess hereinzubringen, erheblich begrenzt. Im Prinzip wird das im Punkt 1.3. dargestellte Problem des Informationsüberflusses durch solche Anwendungen nicht gelöst.

2.3 Multiagentensysteme

2.3.1 Die Technologie der MAS

In sehr vielen Anwendungsfällen, in denen Software-Agenten auftreten, handelt es sich um die Realisierungen, wo mehr als ein Agent eingesetzt wird. Solche Systeme bezeichnet man als Mehragentensysteme [Eym03]. Ein Multiagentensystem (MAS) stellt hingegen einen engeren Begriff dar. Hier handelt es sich immer um die Systeme, in denen nicht nur mehrere intelligente Agenten vorkommen, sondern sich miteinander kommunizieren und daher eine Art von „Gesellschaft" bilden. Jeder MAS ist also ein Mehragentensystem. Die grundlegende Merkmale von Multiagentensystemen lassen sich in folgenden Punkten vorstellen [Big01]:

- jeder einzelne Agent hat eine begrenzte Sicht auf den Zustand der Umgebung (des Systems)
- es gibt keine globale Steuerung
- die Daten sind dezentralisiert
- die Berechnungen (Handeln der Agenten) verlaufen asynchron

Besonders wichtig ist, das in einem MAS kein Zentralplaner das System verwalt, der die Aufgabe selbst aufteilt und an die untergeordnete Einheiten weiterleitet. Dadurch würden die Befugnisse und die Autonomie der einzelnen Agenten, und daher auch die Leistung des Systems, deutlich begrenzt [Eym03].

2.3.2 Vorteile der MAS in Hinsicht auf Informationssuche im Web

Die MAS erweisen sich als sehr zweckdienlich besonders in Fällen von Problemen, die aufgrund Ressourcebeschränkungen oder des zentralen Sicherheitsrisikos zu komplex und umfassend sind, damit eine einzige Entität sie lösen könnte, oder einfach in Fällen von verteilten Problemen, wo die Informationen verteilt vorliegen. Das Web weist viele von den Bedingungen vor, in denen sich MAS bewähren (siehe Punkt 1.1.) [Klsch99]:

- Dezentralisierung (verteilte Datenhaltung) – das ist auch die inhärente Eigenschaft der Multiagentensysteme [Big01], [Eym03].
- Rauschen (unstrukturierte und überflüssige Daten, niedrige Qualität der Daten) – bei den traditionellen Systemen wird das Handeln der Aufwendung zu einzigen Details geplant, was jedoch Zeit bedingt und auch nicht besonders flexibel ist. Die MAS dagegen entwickeln sich ständig weiter. Jedes Optimum ist im Moment der Berechnung schon wieder überholt. Das System wird sozusagen in „Echtzeit" geplant, wobei diese „Planung" nicht zentral läuft [Eym03]. Solcher Ansatz erweist sich als sehr wirksam in einer „chaotischen" Umgebung, die hinsichtlich der Inhalt auch das Web aufweist.
- Heterogenie (heterogene Daten) – die Multiagentensysteme können auch aus unterschiedlichen und vielfältigen Agenten bestehen. Dadurch taugen sie sich hervorragend zum Handeln in heterogenen Umgebungen, wie Internet.
- dynamische Struktur (größer Anteil von wandelbaren Daten) - die einzelne Agenten (Teile von MAS) sind immer im Kontakt zur Umgebung und jede Änderung kann schnell im System widerspiegelt sein. Daher passen sich die MAS äußerst schnell an die externe Bedingungen an [Eym03].
- die riesige Ausmaße und Wachstum – die dezentralisierte Architektur von MAS fördert Skalierbarkeit, d.h. die Fähigkeit zur nicht aufwändigen Anpassung des Systems an die Größe der Datensammlung.

2.3.3 Verwendung der neuen KI-Ansätze

Wie es sich aus obigen Erörterungen ergibt, kann eine Umsetzung der Ideen von MAS in Informationssuche sich als sehr erfolgreich herausstellen und die Probleme der traditionellen Tools überwinden. Eine Implementierung solches Systems ruft aber eine Reihe von Problemen hervor. Besonders problematisch scheint es, die Koordination und Kooperation zwischen Agenten sicherzustellen. Die Suche soll nicht zentral gesteuert werden, sondern selbständig durch einzige Agenten geführt. Sollen sie miteinander kooperieren und einander die Ergebnisse mitteilen? Oder vielleicht sollen sie wetteifern? Wie erlangt man die gleichzeitige Anpassung an die lokalen Bedingungen (Kontext) und an den Benutzer? Eine mögliche Lösung bieten die Technologien, die schon seit 80. und 90. Jahren des letzten Jahrhunderts erfolgreich im Bereich KI verwendet wurden.

Zur Implementierung der verteilten Architektur lässt sich gut der genetische Ansatz anwenden, der ursprünglich im Bereich Problemlösung entwickelt wurde. Hier werden die Hauptprinzipien der Theorie vom Überleben des Stärkeren von Charles Darwin angewandt und die Evolution in Natur als Modell angenommen. Dieser Ansatz zeichnet sich damit aus, dass es eine parallele Bewertung der Individuen (Mitglieder einer Population) unter Einsatz einer universellen, für alle geltenden „Fitness"- („Güte"-) Funktion gibt. Die „Stärksten" werden unter Einsatz der drei grundlegenden Operationen „fortgepflanzt" [Koza92]:

- einfache Reproduktion – Kopieren ins neue Population
- Mutation – die zufällig ausgewählte Teilen der Populationsmitglieder werden geändert.
- Cross-over – Kreuzen von zwei „Eltern" (an der zufällig gewählten Stelle), das zwei neue „Kinder" ergibt.

Der genetische Ansatz hat sich im Laufe der letzten Jahren als sehr erfolgreich erwiesen und gewinnt ständig an Popularität. Die genetische Methoden ermöglichen die Realisierung eines Multiagentensystems, dessen Mitglieder dezentral die Suche im Internet durchführen, und das als Ganze imstande ist, sich an die Bedingungen der Suche und der Suchumgebung schnell anzupassen.

Die Suchagenten bilden zwar ein System, das als eine Gesamtheit gute Leistung aufweisen soll, sie müssen aber selbst über einen gewissen Grad der Intelligenz verfügen, um sich zum Erfolg dieses Systems beitragen zu können. Solche Möglichkeiten bietet ein anderer moderner KI-Ansatz: die neuronalen Netze (NN). Sie haben die Struktur des menschlichen Gehirns als Vorbild: sie setzen sich aus mehreren einfachen Einheiten („Neuronen", „Zellen") zusammen, die durch gerichtete Verbindungen („Synapsen") verbunden sind [Zell94]. Solche Architektur hat eine entscheidende Eigenschaft – Lernfähigkeit auf dem niedrigsten denkbaren Niveau (Rohdaten). NN werden zumeist nicht programmiert, sondern durch ein Lernverfahren „trainiert". Das Lernen erfolgt in den meisten Fällen durch Anpassung der Verbindungsgewichten („Synapsen"), aufgrund eines bestimmten Algorithmus. Dadurch sind die neuronale Netze imstande, ihr Verhalten (die Ausgaben) wandelbaren Eingaben entsprechend anzupassen [Zell94].

3 MAS-Anwendungen im Praxis

3.1 *InfoSpiders* und *MySpider*s

3.1.1 Einführung – das *InfoSpiders*-System

InfoSpiders ist ein an der Iowa und Dayton Universitäten entwickeltes, evolutionäres Suchsystem. Es implementiert eine Population der intelligenten Agenten (*spiders*), die das Web unter Einsatz des Hyperlink-Ansatzes durchsuchen. Die Autoren gehen davon aus, dass die Komplexität der Informationsumgebungen des Internets ist dieser, die in naturellen Umgebungen auftritt, ähnlich. Deswegen soll man auch die Lösungen der Natur in der Informationswiedergewinnung nachahmen. Naturelle „Agenten" verfügen über eine Reihe von Merkmalen, die sich beim Suchen als sehr hilfreich erweisen können. Diese Merkmale sind u.a. lokale Adaptation, Internalisierung der Umgebungssignalen, verteilte Kontrolle und Verbindung des von Außen und Innern stimulierten Benehmens [Klsch99]. Das System ist daher aus einer Menge von Agenten zusammengesetzt, die autonom und evolutionär handeln, um die Anfrage des Benutzers möglichst befriedigend zu bearbeiten. Die einzelne Agenten handeln gewissermaßen ähnlich den üblichen Web-Roboter, d.h. verfolgen HTML-Links und suchen die Ressourcen durch. Die Suche fängt aber erst an, wenn die Anfrage schon bekannt ist, im Gegenteil zu normalen Suchmaschinen. Deshalb wählen die Suchagenten nur diese „Anker" aus, die hinsichtlich konkreter Anfrage relevant scheinen. Sie verfügen über lernfähige Intelligenzmechanismen zur Wahl des nächsten Links und befinden sich dazu ständig „unter Druck" der hohen Leistung, da ihre Ergebnisse auf dem laufenden bewertet sind und die schwächsten Mitglieder der Population einfach entfernt werden. Die stärksten Agenten werden dem genetischen und evolutionären Ansatz gemäß mutiert und zur Fortpflanzung benutzt**.** Die Leistung der einzelnen *spiders* kumuliert sich und versichert eine zufrieden gebende Ausführung der Suchaufgabe. Ursprünglich wurde das System als traditionelle Netzwerkanwendung, die aufgrund der *Client-Server* Architektur und HTTP-Protokoll-Nachrichten, operiert, entwickelt. Dies war die einzige realisierbare Lösung, obwohl die Agenten sequentiell handeln mussten, um sich mit verschiedenen HTTP-Server von einer Maschine aus zu kommunizieren. Künftige mobile Realisierung wurde von den Entwicklern vorausgesetzt.

3.1.2 Der Algorithmus

InfoSpiders benutzen evolutionären Ansatz, der als lokale Selektion genannt wird [Men00]. Jeder Agent wird zu Beginn auf einer Startseite positioniert (normalerweise wird die Menge der Startseiten durch Abfrage einer traditionellen Suchmaschine erzeugt) und mit bestimmter Menge von „Energie" versehen, die für alle Agenten in Population gleich ist. Außerdem wird jedem Agenten ein Startvektor der Schlüsselwörter zugeschrieben, der zufällig aus der Anfrage ausgewählt wird. Die Agenten fangen dann an, das Web durchzukämmen. Jeder Agent schätzt die Relevanz jedes Links, der sich auf „seiner" Seite befindet, in Bezug auf die Übereinstimmung mit seinem Stichwortvektor. Dazu wird das neuronale Netz verwendet, das über bestimmte Gewichtungen für jedes Wort verfügt, das im Schlüsselwortvektor des Agenten steckt. Diese Gewichte werden mit zufälligen Werten aktiviert und im Laufe der Suche und des Lernprozesses geändert, der Relevanzrückmeldung von Benutzer und den äußeren Bedingungen gemäß. Bei jedem Link wird seine Nachbarschaft durchgeforscht und die Schlüsselwörter herausgefischt. Jeder Auftritt des bestimmten Wortes wird mit einem Gewicht versorgt, das von der Distanz des Auftrittes zur Quellseite abhängig ist. Alle diese Daten werden benutzt, um das neuronale Netzwerk zu „füttern" und eine Bedeutungsabschätzung des Links herauszukriegen. Aufgrund dieser Abschätzung wählt der Agent seinen nächsten

„Aufenthaltsort“ aus, wieder mit großem Anteil von Zufallsfaktor, und „bewegt“ sich zur neuen Seite, d.h. herunterladet sie auf die lokale Maschine. Dann wird der Dokument, auf dem sich der Agent gerade befindet, aufgrund der Berechnung von Gewinn und Kosten bewertet, und dementsprechend das Energieniveau des Agenten modifiziert. Der Gewinn wird durch die Ähnlichkeit des Inhaltes des HTMLs mit dem Schlüsselwortvektor berechnet, wobei auch die äußere Gewichtungen für die einzelnen Wörter benutzt werden, falls der Benutzer die Relevanz der Seite bewertet hat („Relevanzrückmeldung“). Die Kosten können unterschiedlich berechnet werden. Die erste Implementierungen des Systems haben den festen Kostenwert benutzt, der nur so angepasst wurde, dass erst bei bestimmter Grenzzahl der besuchten Dokumente (*MAX_PAGES*) wäre die Population wegen des Energieverlusts ausgestorben. Weitere Forschungen haben bewiesen, dass man in Hinsicht auf Geschwindigkeit viel bessere Resultate erreicht, wenn die Kosten von der realen Zeit des Zugangs auf die Seite im Vergleich zu *timeout* des benutzten *socket* abhängig sind [Pan02, Deg01]. Der Wert des Gewinns wird dann mit früher errechneter Relevanzabschätzung des Links, die zu dieser Seite geführt hat, vergleicht. Diese Größen werden benutzt, um im Prozess des Q-Lernens das neuronale Netzwerk zu „trainieren“, indem die Gewichte für Schlüsselwörter geeignet verändert werden. Neue Werte sollen besser das Profil des Anwenders und die Gegebenheiten der Umgebung darstellen. Schließlich wird der Agent hinsichtlich seines Energieniveaus geprüft.

Falls dem Agenten die Energie ausgeht, wird er aus der Population eliminiert. Falls sie hingegen die Schwelle von doppelten Anfangsenergie erreicht hat, werden „Kinder“ durch Cross-over und Mutation erzeugt. Zum Cross-over benutzt man einen anderen, zufällig gewählten Agenten, wobei diesem Prozess nur der Schlüsselwortvektor untergeworfen wird. Mutation betrifft sowohl die Schlüsselwörter, als auch das neuronale Netz. Bei der Mutation des Schlüsselwortvektors berücksichtigt der Algorithmus auch den lokalen Kontext, dadurch dass es eine Möglichkeit besteht, dass ein Wort zum Vektor hinzugefügt wird, das bezüglich der Geburtsseite des Agenten relevant ist. Die Mutation des neuronalen Netzes besteht darin, dass die Gewichtungen etwa zufällig modifiziert werden.

Dem Benutzer werden nur die relevanteste Seiten vorgestellt, d.h. diejenige, deren Ähnlichkeit mit dem Genotyp eines Agenten hoch bewertet wurde, wobei der Agent selbst hohe Fitness vorweisen muss. Der ganze Prozess läuft solange, bis entweder die Population wegen Mangel an den relevanten Seiten ausstirbt, der Grenzwert *MAX_PAGES* (die im voraus definierte maximale Anzahl der besuchten Seiten) erreicht wird oder der Benutzer das System explizit anhält.

3.1.3 *MySpiders* – die Erweiterung des IS-Konzeptes

MySpiders ist ein System, dar auf Basis von *InfoSpiders*, als Java Applett entwickelt wurde. Der Hauptunterschied besteht darin, dass *MySpiders Multithreading* verwendet, was die Leistung deutlich steigert, da die Agenten parallel ihre Aufgaben ausführen können und viele vom System benutzten Ressource und Hilfsprogramme als gemeinsame Objekte gesammelt werden können. Wenn es um den Algorithmus der Suche geht, besteht die einzige Änderung darin, dass die Versicherung der Synchronisierung beim Zugang zu gemeinsamen Ressourcen, wie Cache sichergestellt wurde. *MySpiders* ist den öffentlichen Benutzer zugänglich[2] und bietet eine Java-basierte GUI zur ausführlichen Verwaltung und Beobachtung der Population der Agenten, die die Suche durchführen. Die Implementierung des Systems unter Einsatz mobiler *spiders* ist zurzeit, wegen Mangel entsprechender Standarte, nicht möglich.

[2] siehe http://myspiders.informatics.indiana.edu/

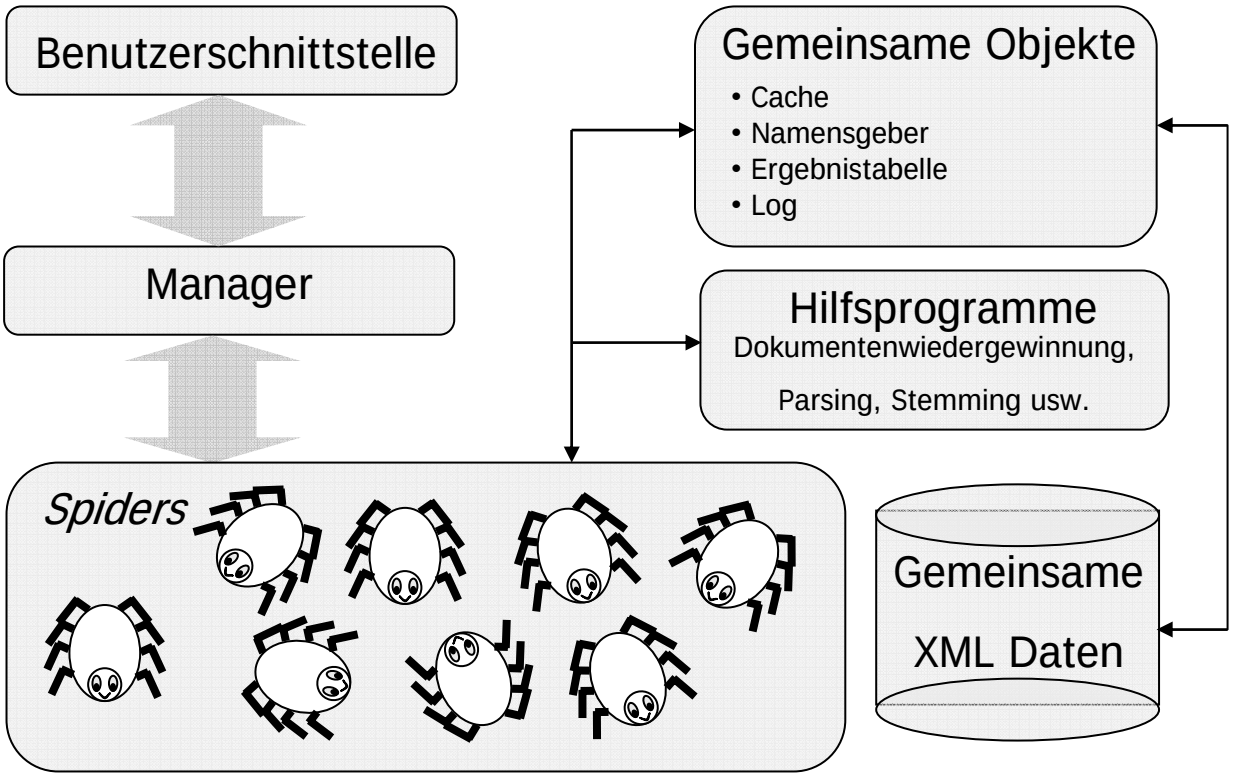

Abbildung 1: Architektur des *MySpiders*-Systems

3.1.4 Leistung

Eine Fallstudie, dargestellt in [Klsch99], zeigt, dass *InfoSpiders* die übliche Suchmaschinen auf die Plätze verweist, wenn es sich um die Aktualität der Ergebnisse handelt. *InfoSpiders* war imstande, drei höchstrelevanten Dokumente herauszusuchen, die erst seit einigen Tagen sich im Web befanden und noch nicht durch die größte *search engines* indiziert worden waren.

Das System wurde auch empirisch bewertet, im Laufe eines Vergleiches mit zwei nicht-adaptiven Algorithmen: einfachem *breadth-first-search* und einer der Implementierungen von *best-first-search* [Klsch99]. Gemessen wurden die so genannte Länge der Suche, d.h. die Gesamtzahl der besuchten Dokumente, bevor ein bestimmter Teil von relevanten Seiten zurück gewonnen wurde. Die Tests wurden auf verschiedenen „Tiefen“ der Anfrage durchgeführt, wobei als Tiefe man hier die minimale Anzahl der Zwischenseiten zwischen Startpunkten und relevanten Dokumenten versteht [Klsch99]. In erstem Test hat natürlich *InfoSpiders* das *breadt-first-search*, die durch Web-Roboter von Suchmaschinen benutzt wird, um mehrere Größenordnungen überholt. Im Laufe des Vergleiches mit *best-first-search* hat es sich herausgestellt, dass bei den mehr allgemeinen Anfrage (niedrige Tiefe) ist *InfoSpiders* erheblich effizienter (niedrigere Länge der Suche und höher Anteil der gelieferten relevanten Dokumenten), wobei bei den sehr spezifischen Anfragen war die Situation umgekehrt. Für beide Algorithmen war die Leistung mit der steigenden Tiefe immer schlimmer [Klsch99].

3.1.5 Zusammenfassung – Eigenschaften

Das *InfoSpiders* System hat viele Vorteile im Vergleich zu üblichen Suchmaschinen. Die Hauptmerkmale der Lösung, die *InfoSpiders* so gute Leistung verleihen, lassen sich in folgenden Punkten darstellen:

- Evolution – der Algorithmus benutzt die erfolgreichsten genetischen Ansätze.

- Anpassung – die Verwendung der evolutionären Techniken und des Q-Lernens führt dazu, dass der Algorithmus sich im Laufe des Suchens dynamisch an die Umgebung und gegebenenfalls an den Benutzer anpasst
- lokale Selektion – die Agenten, die eliminiert oder zur Reproduktion benutzt werden sollen, werden unter Benutzung von konstanten Schwellen, die gleich für die ganze Population sind, ausgewählt, anders gesagt – lokal. Das bedeutet, dass alle die Prozesse der Anpassung durch Evolution finden ohne zentraler Steuerung statt [Men99].
- lokale Internalisierung – die neu geborene *spiders* können auch die Stichwörter, die für ihren Geburtsort relevant scheinen, zu ihrem Schlüsselwortvektor hinzufügen. Daher ziehen *spiders* bei der Anpassung nicht nur die Präferenzen des Benutzers, sondern auch ihre lokale Umgebung in Betracht [Klsch99].
- selektive Expansion der Anfrage – die ständige Evolution des „Chromosoms“, die auch durch den lokalen Kontext beeinflusst werden kann, verursacht, das jeder Agent seine eigene Anfrage „entwickelt“. Die ursprüngliche Anfrage wird daher heterogen bearbeitet [Klsch99].
- Personalisierbarkeit – falls der Benutzer die Seiten hinsichtlich ihrer Relevanz bewertet, können die Agenten evolutionär ihre eigene Vorstellung der Präferenzen des Anwenders (als Gewichte des neuronalen Netzes) entwickeln [Klsch99].
- Skalierbarkeit – *InfoSpiders*-Crawler indizieren WWW-Seiten im voraus nicht, sondern führen eine dynamische, Echtzeit-Suche aus. Daher entfällt das Problem der Skalierbarkeit, das die traditionellen *search engines* kennzeichnet [Klsch99], [Men99]. Evolution und Adaptation der *spiders* versichern relativ hohe Geschwindigkeit des Suchprozesses im Vergleich zu „dummen“ Web-Roboter der Suchmaschinen, die einfache *breadth-first*-Suche durchführen [Hey99].
- Aktualität – die *spiders* benutzen keine Indizierung, womit man nur einen „Snapshot“ des Web schafft [Men99]. Daher ergibt die Suche mit *InfoSpiders* immer aktuellste Resultate, die keine ungültigen Links enthalten [Klsch99].
- Distribution – das Design der Agenten ermöglicht in der Zukunft eine völlig verteilte Version des Systems zu entwickeln, d.h. unter Benutzung der mobilen, direkt an der Informationsquelle handelnden *spiders*. Das wiederum würde die Leistung immer noch steigern, da die Agenten deutlich wenig lokale Ressourcen benutzen müssten und zur parallelen Suche fähig wären [Klsch99].

Der Konzept von *InfoSpiders* hat aber auch ganz bedeutende Nachteile, die den breiteren Einsatz des Systems verhindern. Vor allem ist der Algorithmus erheblich zeitintensiv, da die Agenten erst die Suche starten, wenn die Anfrage bekannt gemacht wird. Eine relevante Suche die die aktuellste Links liefert kann sogar Stunden dauern. Zweitens ist der System immer von den Suchmaschinen abhängig, dadurch dass immer die relevante Anfang-Links zur schnellen und erfolgreichen Suche nötig sind.

3.2 Amalthaea

3.2.1 Einführung

Amalthaea ist ein evolutionäres Multiagentensystem zur personalisierten Informationsfilterung, -suche und -beobachtung. Es assistiert dem Benutzer beim Internet-Surfing, indem es Empfehlungen bezüglich der für ihn interessanten Seiten bietet. Die Informationen werden dem Anwender in Form einer Übersicht (*digest*) vorgestellt, der neue, potentiell interessante URLs, sowie die Änderungen auf den schon vorgeschlagenen Seiten enthält. Durch evolutionäre Gestaltung der Populationen von Agenten lernt das System die Interessen des Benutzers und passt sich an sie dynamisch an. *Amalthaea* implementiert ein künstliches Ekosystem, das aus zwei Populationen von Agenten besteht: den

Informationsrückgewinnungsagenten (*Information Discovery Agents* – IDAs) und den Informationsfilteragenten (*Information Filtering Agents* – IFAs). IDAs beschäftigen sich mit Auffindung und Beobachtung relevanter Seiten, wobei sie dazu verschiedene Suchmaschinen benutzen. IFA wiederum „durchwühlen" die Dokumente, die IDA gefunden haben, um die interessantesten herauszusuchen und an den Anwender zu liefern. Diese Agenten agieren in einer Umgebung, die unter Ansatz der sog. „marktorientierten Kontrolle" gesteuert wird. Er nimmt sowohl Kooperation, als auch Wettbewerb zwischen einzelnen Agenten an. Das System stützt sich auf die Rückmeldung von Benutzern hinsichtlich der Relevanz. Die Bewertungen werden in den gegebenen Kredit umgesetzt, der die Fitness der Agenten bestimmt. Die davon ausgehende Verwendung der genetischen Ansätze ermöglicht die entsprechende Anpassung an die wandelbare Interessen des Benutzers, wobei die gesamte Fitness des Systems zum globalen Gleichgewicht konvergiert.

3.2.2 Die Agenten

3.2.2.1 Informationsrückgewinnungsagenten (IDA)

Die Informationsauffindungsagenten sind für Anfragen der Suchmaschinen und Weiterleiten der Ergebnisse an die Informationsfilteragenten zuständig. Jeder Agent setzt sich aus dem Genotyp (der evolutionäre Teil) und Phenotyp (der sich nicht entwickelte Teil) zusammen. Den Genotyp von IDA bildet Adresse der anzufragenden Suchmaschine, sowie technische Daten, die die Anfragen gestalten (wie z.B. Anzahl der Wörter in einer Anfrage).

Der Phenotyp eines Informationsrückgewinnungsagenten enthält die ihm beschreibenden Daten, wie die ID, das Fitnessniveau, das Erzeugungsdatum und die Geschichte der Transaktionen („Log"), sowie den Ausführungskode, der auf den Genotyp zurückgreift.

IDAs spezialisieren sich in drei Bereichen: Informationsauffindung (Beschaffung neuer, noch nicht bekannter Dokumente), Beobachtung (Ermittlung der Änderungen auf den vom Benutzer ausgewählten Seiten) und Handeln der Informationsströmung (Filtern von den Dokumenten, die an das System ankommen, z.B. durch Nachrichten-Services).

3.2.2.2 Informationsfilteragenten (IFA)

Die Informationsfilteragenten sind für Filterung der von IDA herausgesuchten Informationen und Lieferung der ausgewählten Dokumente an den Benutzer verantwortlich. Der Genotyp dieser Agenten besteht aus einem Vektor der Schlüsselwörter mit den Gewichtungen, die die Präferenzen des Anwenders darstellen. Der Phenotyp ist dem Phenotyp von IDA ähnlich (siehe Punkt 3.2.2.1.), wobei sie keinen Log bewahren, sondern eine boolesche Flag, die sagst, ob der bestimmte Agent explizit vom Benutzer geschafft wurde. Die Informationsfilteragenten richten die Aufträge an die Informationsauffindungsagenten aufgrund ihres Stichwortvektors. Sie vergleichen dann die Ergebnisse mit dem Vektor, unter Nutzung von *tf-idf*-Techniken und reichen die ihrer Meinung nach relevanteste Dokumente an den Anwender in Form der schon erwähnten Übersicht weiter.

3.2.3 Das System

3.2.3.1 Die Architektur

Das Herz des Systems bilden die beide Populationen von Agenten. Das niedrigste Niveau machen die Informationsrückgewinnungsagenten aus, die sich nur mit den Informationsfilteragenten kommunizieren. Dabei benutzen sie auch bestimmte Hilfsprogramme (u.a. eine Datenbank zum Speichern der Dokumente). Zwischen den IFAs und dem Benutzer besteht eine Java-Schnittstelle, die die Verwaltung des Systems ermöglicht. Man

bekommt dadurch die Übersichten der empfohlenen Dokumente, kann sie im Browser aufrufen und seine Bewertung („Rückmeldung“) angeben. Auch eine Reihe von Systemparameter, wie z.B. Anzahl der bestimmten Agenten oder die beobachteten Seiten, lässt sich durch Schnittstelle einstellen.

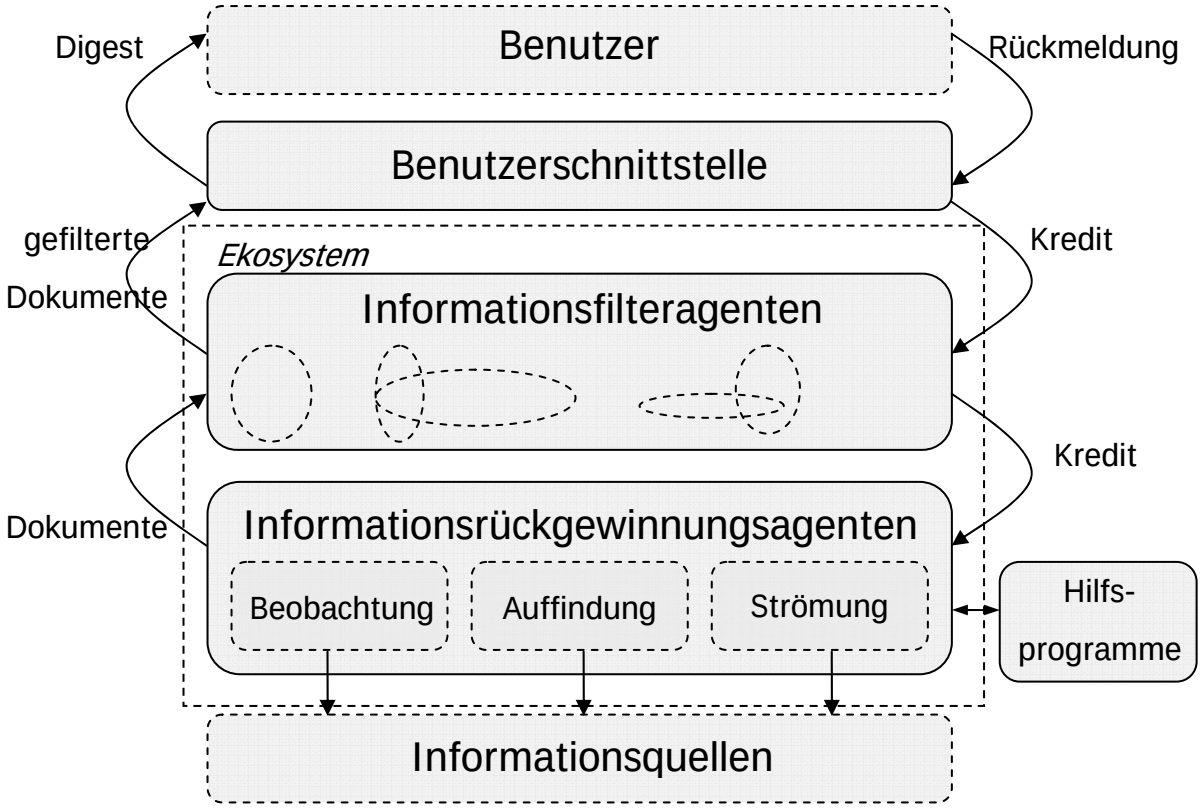

Abbildung 2: Architektur des *Amalthaea*-Systems

3.2.3.2 Der Algorithmus

Das System fängt mit der Erzeugung der ersten Generationen von IDAs und IFAs an. Die Genotypen der Informationsbeschaffungsagenten werden nach Zufallsprinzip generiert. Die ersten Informationsfilteragenten müssen aber für den Benutzer gewissermaßen relevant sein. Es gibt mehrere Optionen, um das sicherzustellen, u.a. durch Angabe der Liste der beliebten Seiten, Geschichte des Browsers oder explizite Generierung eines IFA unter Verknüpfung mit einem relevanten Dokument oder Textabschnitt. Die Erzeugung von IFAs lässt sich prinzipiell auf Schaffung eines Stichwortvektors mit entsprechenden Gewichtungen nieder.

Die Zusammenarbeit der Agenten besteht darin, dass die IFAs „Aufträge“ an die IDAs angeben, Dokumente mit bestimmten Termen aufzufinden. Diese Aufträge werden auf einem „Schwarzen Brett“ untergebracht und die IDAs wählen die für sie günstigsten Partner aus, indem sie die Geschichte der „Transaktionen“ mit dem bestimmten IFAs überprüfen (20% der Auftragswahlen wird aber probabilistisch determiniert, in Hinsicht auf das entsprechende Vielfältigkeit). Die gefundene Dokumente werden in der Datenbank gespeichert, in Form eines Vektors. Die IFAs vergleichen dann die für sie verschaffte Dokumente mit ihrem eigenen Stichwortvektor, der die Präferenzen des Benutzers darstellt. Sie wählen die beste aus und kalkulieren das Vertrauensniveau, aufgrund des Ähnlichkeitsmaßes und ihrer eigenen Fitness. Die besten Dokumente werden an den Anwender geliefert (die Anzahl der Dokumente in der Übersicht ist bestimmbar). Dabei handelt es sich hier um die vertrauenswürdigsten Seiten aus jeder bestimmten Interessengruppe, über deren die Informationsfilteragenten verteilt sind. Auch die Evoluton der IFAs erfolgt in der bestimmten Gruppen (die besten und die schwächsten Agenten in jeder Gruppe werden zur Reproduktion/Eliminierung gewählt). Die IFAs in einer bestimmten Gruppe stehen demnach im Wettbewerb zueinander. Üblicherweise

bekommt der Benutzer die Empfehlungen nur von ungefähr 40% der IFAs, die über das höchste Fitness verfügen. Der Rest der IFA-Population existiert nur wegen der Forderung an Mannigfaltigkeit. Der Benutzer kann die Empfehlungen prüfen und eine Note (von 1 bis 7 Punkte) ausstellen. Diese Bewertung wird ins Kredit umgesetzt, das der Informationsfilteragenten zugegeben wird, abhängig vom früher angegebenen Vertrauensniveau. Die IFAs leiten dann Teil des Kredits an ihre „Informationslieferanten" (IDAs) weiter. Die Entwicklung des System wird durch evolutionäre Ansätze gesichert. Die fitnessstärksten Agenten werden unter Einsatz von Mutation und Cross-over reproduziert. Die leistungsschwächsten Individuen werden dagegen eliminiert. Diese Operationen werden teilweise zentral gesteuert, indem sie von dem Fitness des gesamten Systems abhängig sind. Wenn das gesamte Fitness steigert, werden die Anteile der zu der evolutionären Entwicklung und Eliminierung gewählten Agenten geringer, damit das System sich langsamer entwickelt und Informationsgebiete besser entdeckt. Im umgekehrten Fall werden die entsprechende Sätze erhöht, damit sich *Amalthaea* besser an die wechselnden Interessen des Anwenders anpasst. Die Fitness von Informationsfilteragenten wird zusätzlich linear abgebaut, um die Situation zu verhindern, dass ein IFA keine Empfehlungen vorstellt. Auch die Empfehlung von Seiten, die dem Benutzer schon früher bekannt gegeben wurden, wird durch Entnehmung von Fitness bestraft.

3.2.4 Zusammenfassung – Leistung

Das System wurde ausführlich in zwei verschiedenen Richtungen getestet. Zuerst wurden die künstliche Benutzer mit den Profilen, die wahren menschlichen Interessen entsprachen, benutzt. Zuerst wurde festgestellt, dass das System in den stabilen Bedingungen (relativ unveränderte Interessen) seine Fitness immer steigert und zu einem Gleichgewicht in der Nähe von 1 (Optimum) konvergiert. Beim absichtlichen, heftigen Änderung der Interesse hat sich herausgestellt, dass *Amalthaea* nach den plötzlichen Fitnessverluste imstande war, sich an neuen Bedingungen wiederanzupassen und seine Fitness wiederzusteigern. Schließlich, wenn es keine Bewertung der Vorschläge des Systems gab, hat sich die gesamte Fitness zuerst schnell verringert, diese Verluste könnten aber später, wenn die Rückmeldung vorhanden war, ausgeglichen und überholt werden.

Eine andere Reihe der Experimente wurde mithilfe der menschlichen Benutzer durchgeführt, um zu erforschen, ob das System in der Tat relevante Dokumente empfehlen kann. Die Relevanznoten 5, 6 und 7 in einem Test, dessen Zahlbestand über 5000 Proben betrug, haben ungefähr 63% der Gesamtheit ausgemacht. Auch andere Schätzungen, sowie die Meinungen der Teilnehmer haben bestätigt, dass der Konzept, der hinter *Amalthaea* steckt, sich bewährt hat.

3.2.5 Eigenschaften

Amalthaea weist viele Eigenschaften vor, die schon bei *InfoSpiders* besprochen wurden. Evolution und Anpassung stehen im Mittelpunkt der Architektur, wobei das System besonders an Anpassung an den Benutzer ausgerichtet ist, was wiederum die Personalisierbarkeit von *Amalthaea* hervorhebt. Ein wichtiger Vorteil ist die Fähigkeit, die Fitness immer zu steigern und zum Optimum zu konvergieren, sogar wenn es „Turbulenz" der Interessen des Benutzers gibt. Auch Aktualität der Empfehlungen ist erhalten, dadurch dass ein Teil von IDAs sich mit der ständigen Beobachung der Änderungen beschäftigt.

Anderseits greift das System unvermeidlich auf die üblichen Suchmaschinen zurück, um die Informationen zu gewinnen, was zwangsläufig alle ihre Unvollkommenheiten nach sich zieht, besonders wenn es sich um Vollständigkeit (im Sinne *recall*) der Empfehlungen handelt.

4 Zusammenfassung: Erweiterung des MAS-Konzeptes

Die beiden Systeme, wie das besprochen wurde, besitzen Mängel, die ihre Leistung überschatten. Anderseits schaffen sie aber neue Ansätze, die sich gegebenenfalls zur Lösung des Informationsüberflussproblems beitragen könnten. Wenn man entsprechend die beste Eigenschaften der beiden Systeme verschmelzen würde, könnte eine evolutionäre, gewissermaßen intelligente Suchmaschine entwickelt werden, die die Mängel ihrer Vorgänger vermeidet. Als eine „Brücke“ zur Spannung der beiden, doch nicht so entfernten Konzepte, kann der kollaborative Ansatz dienen (siehe Punkt 2.2.).

Solches System könnte, wie *Amalthaea*, sich aus 2 Populationen von Agenten zusammensetzen – IDAs und IFAs, wobei es jedoch hier um eine Suchmaschine geht, die nur auf explizite Anfragen des Benutzers antwortet.

Die IDAs wären eher dem erfolgreichen Konzept von *InfoSpiders* ähnlich. Sie würden im Laufe der Suche ihre Anfrage dynamisch gestalten, um sich besser an den Kontext und an die Umgebung anzupassen. Sie würden aber die Suchaufträge von IFAs ähnlich dem IDAs von *Amalthaea,* annehmen. Die Annahme neuer Anfrage würde entsprechende Mutation des Genotyps (Stichwortvektors) auslösen. Genauer gesagt handelt es sich hier um eine Art Cross-over mit dem Vektor des Auftraggebers. Ein IDA müsste sowieso mit seinem IFA-„Auftraggeber“ irgendwo inhaltlich verbunden sein, deswegen würde schon zu Beginn der Suche sein Genotyp hinsichtlich der Anfrage ziemlich relevant. Möglicherweise könnte ein *spider* im Laufe einer Suche mehrere IFAs „bedienen“. Die Fitness der IDAs würde sowohl durch ihre eigene Suche, wie in *IS*-System (lokale Selektion), als auch durch eventuelles Kredit von den Auftraggebern (IFAs) gestaltet.

Die IFAs wären der wichtigste Teil des Systems. Sie würden in mehreren Gruppen verteilt, die dynamisch entstehen, verschwinden oder sich ändern könnten. Jede Gruppe könnte durch einen „gemeinsamen Nenner“ der Genotype der dazu gehörenden IFAs beschrieben werden. Dieses „Kennzeichen“ würde dazu dienen, Anfragen der Benutzer zu bestimmter Gruppe zu zählen, wobei fortgeschrittene kollaborative Techniken, die auf statistische Teste zurückgreifen, Verwendung finden könnten. Das System könnte auch möglicherweise mit lokalen Client-Anwendung (Browser-Plugin) kooperieren, damit die Geschichte der Anfragen benutzt werden könnte, um den Anfrageklassifizierungsprozess zu optimieren. Die bestimmte Gruppen wurden erhalten, solange ihre gesamte Fitness (Summe der Fitness der dazu gehörenden Agenten) entsprechend hoch wäre. Ein wichtiger Teil der Existenz der IFAs wäre Wettbewerb – zuallererst um die Anfragen, die an ihre Gruppe ankämen (eine Form der „Versteigerung“), wobei die Ähnlichkeit zum Genotyp, sowie Fitness der Agenten ausschlaggebend wären. Ausserdem würden IFAs, wie in *Amalthaea*, untereinander um Überleben und gegebenenfalls Reproduktion „kämpfen“. Es besteht auch die Möglichkeit, dass die ganze Gruppen konkurrieren würden, wenn sich darin genügend ähnliche Populationen der IFAs entwickelt hätten. Jede Gruppe würde Zugriff auf eigene Datenbank haben, wo die „frischste“ Ergebnisse der von der Gruppe bedienten Anfragen vorlägen. Das wäre die allererste Quelle, um die Anfrage des Benutzers zu beantworten. So könnte der Hauptmangel von *InfoSpiders* gelöst werden. Aktualität würde gesichert, indem die entsprechend alte Ergebnisse aus der Datenbank gelöscht würden. Die Fitness der IFAs würde, wie im Falle von *Amalthaea*, auf dem Kredit des Benutzer basieren.

Der Zahlbestand der beiden Populationen würde teilweise zentral gesteuert, in Abhängigkeit von der gesamten Fitness, ähnlich dem *Amalthaea*-Ansatz. Die beiden Konzepte haben sich schon bewährt, deswegen kann man vermuten, dass wenn nur solches System entsprechend implementiert und mit „guten“ Parameter versehen würde, könnte es einen weiteren Schritt in Richtung voller Skalierbarkeit im Gesicht der Größe des Internets schaffen. Die einzige Voraussetzung, die sich als vorteilig ergeben könnte, ist die vermutlich hohe Anzahl der Anwender, um das Potential des Systems voll ausnutzen zu können.

5 Literaturverzeichnis

[Bae99] Baeza-Yates, R., Ribeiro-Neto, B., Modern Information Retrieval, Addison-Wesley, Bonn 1999

[Big01] Bigus, J., Bigus, J., Intelligente Agenten mit Java programmieren, Addison-Wesley, München 2001

[Böhm99] Böhme, T., Agentensysteme – aktueller Entwicklungsstand und Konzeption eines universellen News-Watcher-Agent, Institut für Informatik, Universität Leipzig, 1999

[Cag98] Caglayan, A. K., Harrison, C. G., Intelligente Software-Agenten, Hanser, München 1998

[Deg01] Degeratu, M., Pant, G., Menczer, F., Latency-dependent fitness in evolutionary multi-threaded Web agents, 2001, in: Internet URL: http://citeseer.ist.psu.edu/453977.html (Stand 21.01.2006)

[Eym03] Eymann, T., Digitale Geschäftsagenten, Springer, Berlin Heidelberg 2003

[Fer03] Ferber, R., Information Retrieval, dpunkt.verlag, Heidelberg 2003, in: Internet URL: http://information-retrieval.de/irb/ir.html (Stand 19.12.2005)

[Gul05] Gulli, A., Signorini, A., Indexable Web Size, 2005, in: Internet URL: http://www.cs.uiowa.edu/~asignori/web-size/ (Stand 10.01.2006)

[Gur70] Guralnik, D.B, Websters World Dictionary, World Publishing Company, Cleveland 1970

[Hey99] Heydon, A., Najork, M., Mercator: a Scalable, Extensible Web-crawler, 1999, in: Internet URL: http://www.research.compaq.com/SRC/mercator/papers/www/paper.html (Stand 18.12.2005)

[Klsch99] Klusch, M., Intelligent Information Agents, Springer, Berlin 1999

[Koza92] Koza, J., Genetic Programming, MIT Press, Cambridge, Mass, 1992

[Men99] Menczer, F., Belew, R. K., Adaptive Retrieval Agents: Internalizing Local Context and Scaling up to the Web, 1999, in: Internet URL: http://citeseer.ist.psu.edu/200755.html (Stand 08.12.2005)

[Mouk97] Moukas, A., Zacharia, G., Evolving a Multi-Agent Information Filtering Solution in Amalthaea, 1997, in: Internet URL: http://citeseer.ist.psu.edu/149984.html (Stand 03.01.2006)

[Mouk98] Moukas, A., Maes, P., Amalthaea: An Evolving Multi-Agent Information Filtering and Discovery System, 1998, in: Internet URL: http://citeseer.ist.psu.edu/390925.html (Stand 03.01.2006)

[Pan02] Pant, G., Menczer, F., MySpiders: Evolve your own intelligent Web crawlers, 2002, in: Internet URL: http://citeseer.ist.psu.edu/pant02myspiders.html (Stand 06.12.2005)

[Zell94] Zell, A., Simulation Neuronaler Netze, Addison-Wesley, Bonn 1994