

Reihe: Telekommunikation @ Mediendienste · Band 11

Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof. Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Rainer Kuhlen, Konstanz, Dr. Rudolf Pospischil, Brüssel, und Prof. Dr. Claudia Löbbcke, Köln

PD Dr.-Ing. habil. Martin Engelien  
Dipl.-Inf. Jens Homann (Hrsg.)

# Virtuelle Organisation und Neue Medien 2001

Workshop GeNeMe2001  
Gemeinschaften in Neuen Medien

TU Dresden, 27. und 28. September 2001



**JOSEF EUL VERLAG**  
Lohmar · Köln

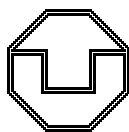
Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Virtuelle Organisation und Neue Medien 2001 / Workshop GeNeMe 2001 – Gemeinschaften in Neuen Medien – TU Dresden, 27. und 28. September 2001. Hrsg.: Martin Engeli; Jens Homann. – Lohmar; Köln: Eul, 2001  
(Reihe: Telekommunikation und Mediendienste; Bd. 11)  
ISBN 3-89012-891-2

© 2001

Josef Eul Verlag GmbH  
Brandsberg 6  
53797 Lohmar  
Tel.: 0 22 05 / 90 10 6-6  
Fax: 0 22 05 / 90 10 6-88  
<http://www.eul-verlag.de>  
[info@eul-verlag.de](mailto:info@eul-verlag.de)  
Alle Rechte vorbehalten  
Printed in Germany  
Druck: RSP Köln

**Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.**



Technische Universität Dresden  
Fakultät Informatik • Institut für Angewandte Informatik  
Privat-Dozentur „Angewandte Informatik“

PD Dr.-Ing. habil. Martin Engelen,  
Dipl.-Inf. Jens Homann  
(Hrsg.)

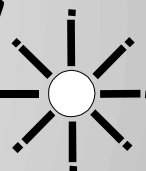
Dresden, 27./28.09.2001

# ***GENEME 2001***

***Gemeinschaften in Neuen Medien***

*Workshop zu Organisation, Kooperation und  
Kommunikation auf der Basis innovativer Technologien*

*Forum für den Dialog zwischen Wissenschaft und Praxis*



an der  
Fakultät Informatik der Technischen Universität Dresden

gefördert von der Klaus Tschira Stiftung  
gemeinnützige Gesellschaft mit beschränkter Haftung



am 27. und 28. September 2001  
in Dresden

<http://pdai.inf.tu-dresden.de/geneme>  
Kontakt: Thomas Müller (tm@pdai.inf.tu-dresden.de)

---

## A.3. Adaptive Datenverwaltung im Internet

Markus Wulff

Dr. Herwig Unger

Fachbereich Informatik, Universität Rostock

### 1. Einführung

Mit der ständig wachsenden Bedeutung des Internets als Medium für Kommunikation und Datenverarbeitung werden auch Qualitätsmerkmale wie Verfügbarkeit, Zuverlässigkeit und Sicherheit immer wichtiger. Das gilt insbesondere für die Anwendung im Rahmen des e-Business und anderer kommerzieller Anwendungen [13].

Große Computernetzwerke, wie z.B. das Internet, sind meistens als Client-Server- oder Brokersysteme organisiert. Die zentrale Instanz, also der Broker oder Server, ist damit die Schwachstelle des Systems. Die oben genannten Qualitätsmerkmale können nicht gesichert werden. Die Probleme, die zentrale System haben, sind vielfältig.

- *Verfügbarkeit*

Die Netzwerklast kann im Verlauf z.B. eines Tages stark schwanken. Der Server ist zumeist nicht in der Lage, sich den aktuellen Gegebenheiten automatisch anzupassen, was dann während der Lastspitzen zu Einschränkungen führen kann. Die Antwortzeiten können in Abhängigkeit von der Last ebenfalls stark schwanken. Wenn der Server oder das Netzwerk überlastet sind, kann das sogar dazu führen, dass der entsprechende Dienst zeitweilig überhaupt nicht verfügbar ist [19,27].

- *Sicherheit*

Ein Ausfall der zentralen Instanz hat in den meisten Fällen auch einen Ausfall des entsprechenden Dienstes zur Folge. Damit ist die Sicherheit des Systems, z.B. durch *denial-of-service* Angriffe gefährdet. Auch die Fehlertoleranz ist in einem solchen Client-Server-System zumeist nicht gewährleistet, da wie schon erwähnt, in einem Fehlerfall alle zentral gehaltenen Ressourcen nicht mehr verfügbar sein können.

- *Aktualität*

Zentrale Kataloge, Datenbanken oder andere Datenbestände sind meistens sehr groß und daher nicht einfach zu pflegen. Ein Beispiel dafür sind die Verzeichnisse der Web-Suchmaschinen, die aufgrund der Fülle der zu erfassenden Seiten zum einen unvollständig sind und zum anderen nicht immer aktuell, da ein ausreichend häufiges Update nicht möglich ist.

Das Internet, speziell das World Wide Web (WWW oder Web), gewinnt für die Beschaffung und den Austausch von Informationen immer mehr an Bedeutung. Es besteht derzeit aus etwa vier Milliarden Seiten mit stark wachsender Tendenz [31].

Dabei gibt es nahezu keinerlei Strukturierung der Dokumente und durch das häufige Hinzufügen und Entfernen von Seiten unterliegt es auch einer ständigen Änderung. Die zur Recherche zur Verfügung stehenden Suchmaschinen haben, wie oben schon beschreiben, den Nachteil, dass sie wegen der Größe der Datenbestände nicht immer zufriedenstellend arbeiten.

Eine Lösung für die beschriebenen Probleme ist die Verwendung dezentraler Konzepte in der Netzwerkumgebung. Dezentrale Systeme zeichnen sich dadurch aus, dass sie keine zentralen Server, Daten o.ä. haben. Alle Informationen und Dienste, die in dem System angeboten werden, sind auf alle Mitglieder verteilt. Das gilt auch für die Information über die Größe und Struktur des gesamten Systems. Sie sind lokal in speziellen Warehouses gespeichert [18,19,28].

Mit diesem Ansatz kann eine sehr hohe Flexibilität und Fehlertoleranz erreicht werden. Beim Ausfall eines Knotens geht nur ein kleiner Teil der Ressourcen verloren, welche zusätzlich auch noch auf anderen Knoten vorhanden sein können. Auch das Hinzufügen eines neuen Knotens bereitet keine Schwierigkeiten, da keine zentrale Systeminformation aktualisiert werden muss [27].

Im folgenden werden Vorteile und Funktionsweise dezentraler Systeme anhand von Communities näher erläutert und untersucht.

## **2. Communities**

Im Internet gibt es Anbieter von Informationen und Diensten mit gleichen oder ähnlichen Interessen, Arbeitsgebieten etc. Diese bilden dadurch eine so genannte *Community* [8,10,12,16,20]. Da die jeweiligen Nutzer auch auf Dienste anderer Mitglieder der Community zugreifen, bilden sie implizit eine logische Struktur, die die physikalische Netzwerkstruktur überlagert. Im Gegensatz zu der festen, im allgemeinen nicht veränderbaren Topologie des Netzwerkes ist die durch die Communities gebildete Struktur veränderbar und kann so an bestimmte Anforderungen angepasst werden. Somit kann folgende Definition für Communities gegeben werden [28]:

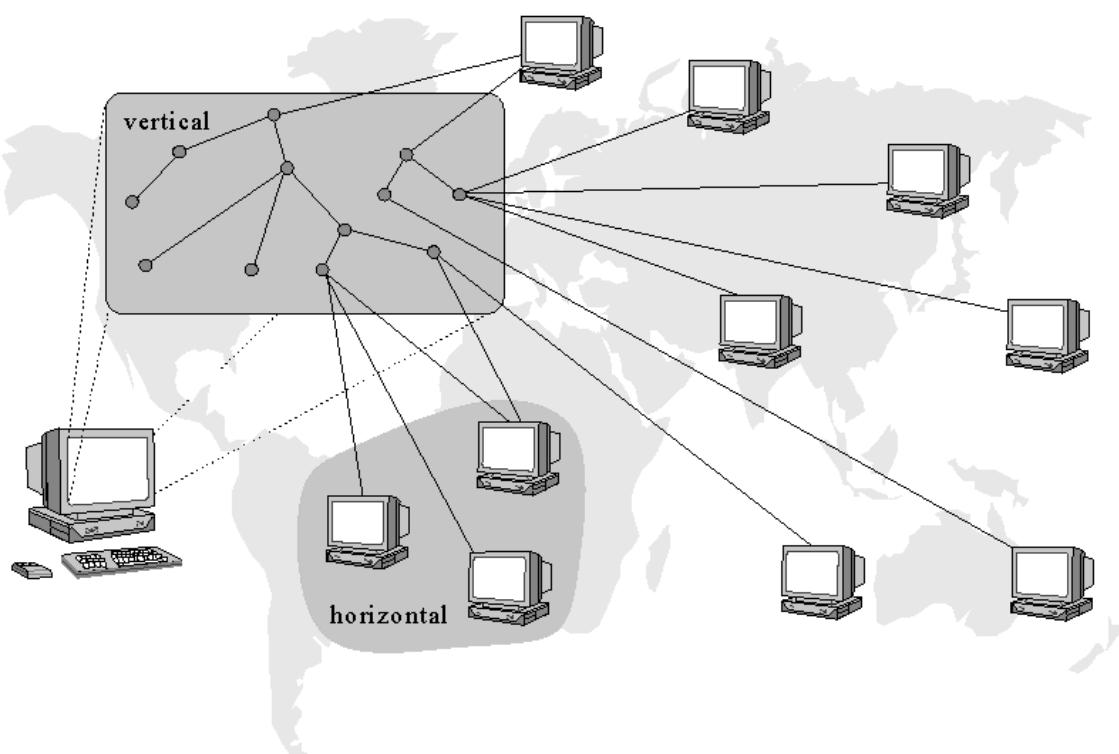
Eine Community wird durch die Nachbarschaftsbeziehungen zwischen Anbietern gleicher oder ähnlicher Inhalte im Netz gebildet.

Wenn man diese Struktur gezielt aufbaut und z.B. in lokalen Warehouses die für den jeweiligen Nutzer interessanten Nachbarknoten speichert, kann dadurch eine leistungsfähige, skalierbare und flexible logische Vernetzung geschaffen werden. Diese ist durch die auftretende Redundanz tolerant gegenüber Störungen auf den Knoten oder im Netzwerk.

Durch das Vorhandensein eines Servers und verschiedener Warehouses auf jeden Knoten können in einer solchen Umgebung auch Funktionen realisiert werden, wie sie

von verteilten Betriebssystemen bekannt sind [15]. Zum Beispiel sind effektive Suchverfahren für verteilte Systeme in einer Community realisierbar [29].

Ein Nutzer kann Mitglied in mehr als einer Community sein. Je nach seinen Interessengebieten können die Knoten in der Nachbarschaft verschiedenen Themengebieten zugeordnet und getrennt verwaltet werden. Die einzelnen Gebiete können aber auch miteinander in einer Beziehung stehen und somit entsteht durch diese Verbindungen eine *vertikale Community*. Diese Strukturierung sowie die Wahl der Schlüsselworte bleibt dabei dem Nutzer überlassen und stellt daher seine Sicht auf die Unterteilung der Themengebiete dar. Die unter jedem Punkt zusammengefassten Verweise zu anderen Rechnern sind durch den gemeinsamen Eintrag zu einem bestimmten Thema ebenfalls logisch verknüpft und bilden dadurch eine *horizontale Community*. Durch diese Form der Organisation wird der Suchprozess unterstützt, indem Abstraktionen und Verfeinerungen dem Nutzer durch statistische Auswertung und Vergleich der vertikalen Communitygraphen anderer Knoten, die bei der Suche betrachtet wurden, vom System vorgeschlagen werden können.



**Abbildung 1: Vertikale und horizontale Communities**

In Abbildung 1 ist der oben beschriebene Ansatz noch einmal graphisch dargestellt. Die vom Nutzer definierten Relationen zwischen den einzelnen Themen sind als „Bildschirm“ rechts oben abgebildet und von dort ausgehend die Verweise zu Rechnern mit den entsprechenden Ressourcen.

Die Suche in den Communities ist ein zentraler Punkt bei dem Aufbau der lokalen Warehouses und dem Finden von Ressourcen im Netz. Im nächsten Abschnitt ist die Suche deshalb genauer beschrieben und es werden auch Vorschläge zur Optimierung angegeben.

### 3. Suche in Communities

Trotz der Verteilung aller Informationen über die gesamte Community und dem Fehlen jeglicher zentraler Informationen über verfügbare Ressourcen, die Struktur und die derzeitigen Mitglieder der Community muss es möglich sein, auf alle Ressourcen zuzugreifen zu können. Eine weitere Schwierigkeit ist die Dynamik in einem solchen Verbund. Zu jeder Zeit kann ein Knoten aus der Community verschwinden oder ein neuer hinzukommen. Es ist klar, dass hierfür ein leistungsfähiger Mechanismus benötigt wird, um das Auffinden der vom jeweiligen Nutzer benötigten Ressourcen zu ermöglichen.

*Message Chains* [30] sind eine spezielle Form der Kommunikation in verteilten Systemen wie z.B. Communities. Sie funktionieren wie folgt: Wenn ein Knoten nach einer bestimmten Ressource suchen will, sendet er einfach die Anfrage an einen zufällig ausgewählten Nachbarn und dieser wird die Anfrage dann bearbeiten, das Ergebnis zurücksenden und die ursprüngliche Anfrage nun an einen seiner Nachbarn senden. Damit die Message Chain nach einer bestimmten Anzahl von Hops terminiert, wird auf jeden besuchten Knoten der Hop-Counter um eins verringert und die Nachricht nur solange weitergeleitet, wie der Zähler größer null ist. Damit kann vor dem Senden festgelegt werden, wie viele Knoten besucht werden sollen.

Durch einen entsprechenden Wert für den Hop-Counter kann gewährleistet werden, dass alle Ressourcen gefunden werden. Das gesamte Internet hat zur Zeit etwa einen Durchmesser von 19 [31]. Damit ist der Aufwand für die Suche in einem effizienten Rahmen.

Dadurch, dass nun alle Knoten derartige Nachrichten aussenden, kann die Last im Netzwerk stark ansteigen. Um diese Last zu verringern, wurde das *Merging* [28] eingeführt. Dabei werden zwei sich auf einem Knoten treffende Message Chains zu einer Nachricht verbunden. Es wird zwar weiterhin jede der Einzelnachrichten auf den jeweiligen Knoten separat verarbeitet und auch der Hop-Counter bleibt separat, das Weitersenden geschieht jedoch immer gemeinsam. Dadurch werden weniger aber etwas größere Message Chains im Netz unterwegs sein, was aber die Gesamtlast im Netz verringert.

Message Chains sind somit ein mächtiges und universelles Werkzeug für alle anfallenden Kommunikationsaufgaben in Communities. Die Suche kann aber durch

eine geeignete Strukturierung der Warehouses noch effizienter gestaltet werden. Im Gegensatz zu einer beliebig gewachsenen Communitystruktur können die lokalen Einträge so organisiert werden, dass zum einen die Suche innerhalb einer Community und zum anderen auch die Suche auf allen lokal gespeicherten Knoten optimiert wird. Folgende Anforderungen müssen an eine optimale Struktur einer Community gestellt werden.

- Die neue Topologie sollte nur aus lokaler Information aus unstrukturierter Community zu konstruieren sein.
- Der Durchmesser der Community sollte bekannt und möglichst klein sein (was bei einer gewachsenen, unstrukturierten Community nicht bekannt ist).
- Bei möglichst geringer Valenz sollte eine hohe Fehlertoleranz vorhanden sein.
- Es sollten bekannte Algorithmen z.B. für das Routing angewendet werden können.

Eine Möglichkeit hierzu ist die Topologie des n-dimensionalen *Hypercubes*. Diese hat einige sehr gute Eigenschaften, die bei der Suche erhebliche Vorteile bringt. Ein n-dimensionaler Hypercube hat einen Durchmesser von n bei einer maximalen Knotenzahl von  $2^n$ . Diese Topologie läßt sich mit Hilfe nur der lokalen Informationen im Community-Warehouse aufbauen.

Der Knoten  $v$  ist Bestandteil der Menge  $V$  aller Knoten, die der neu zu organisierenden Community angehören. Jeder Knoten kann einen Zustand  $s$  aus  $S(v) = \{\text{passiv, aktiv, kind, null}\}$  haben. Um eine Hypercubestruktur aufzubauen muss der folgende Algorithmus ausgeführt werden:

### 1. Suche nach einem neuen Knoten

Wenn  $S(v) = \text{passiv}$

- Finde einen Knoten  $x$  mit  $S(x) = \text{null}$
- Setze  $N(x) = \{v\}$  und  $M(x) = N(v)$  wobei  $N(a)$  die Nachbarschaft des Knotens  $a$  ist.
- Setze  $N(v) = N(v) \cup \{x\}$
- $S(v) = \text{aktiv}$  und  $S(x) = \text{kind}$

### 2. Einbinden des neuen Knotens

- Wenn  $S(x) = \text{kind}$  tue für alle  $z$  aus  $M(x)$

Wenn  $S(z) = \text{aktiv}$  und  $y = y(z)$  (ist Sohn von  $z$ ), setze

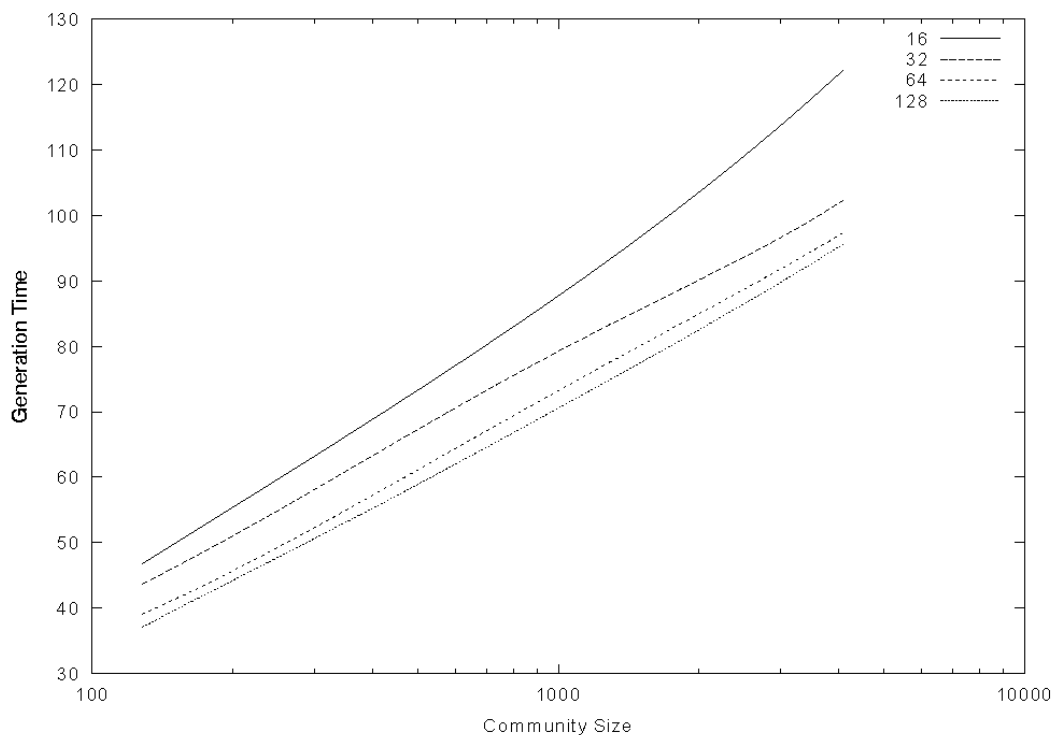
- $M(x) = M(x) - \{z\}$
  - $N(x) = N(x) \cup \{y\}$
- Wenn  $S(x) = \text{kind}$  und  $M(x) = \emptyset$  setze
  - $S(x) = \text{passiv}$
  - Setze Vater  $v$  von  $x$   $S(v) = \text{passiv}$



Jeder Knoten im Hypercube bekommt eine eindeutige ID, welche auch einen Zeitstempel beinhaltet. Treffen sich nun zwei verschiedene Hypercubes, so darf der mit der höheren Dimension bzw. bei gleicher Dimension der ältere den anderen zerstören, d.h. ihm den benötigten Knoten wegnehmen.

Neben der vom Nutzer Stück für Stück angelegten unstrukturierten Community kann also noch eine optimierte Hypercubestruktur aufgebaut werden. Da ein Nutzer Mitglied mehrerer Communities sein kann, kann damit für jede Community ein solcher Hypercube aufgebaut werden, was die Operationen, insbesondere die Suche in der jeweiligen Community optimiert. Dies ist auch deshalb möglich, da ein Hypercube relativ wenig lokale Einträge benötigt. Es ist ebenfalls ein globaler Hypercube denkbar, der alle auf dem Knoten gespeicherten Links zu einer gemeinsamen Struktur verbindet und damit z.B. eine effiziente Suche über Community-Grenzen hinweg ermöglicht.

In der Abbildung 2 ist die Zeit für die simulierte Erzeugung eines globalen Hypercubes in Abhängigkeit von der Communitygröße dargestellt. Die verschiedenen Graphen stehen dabei für die unterschiedliche Größe der lokalen Nachbarschafts-Warehouses.



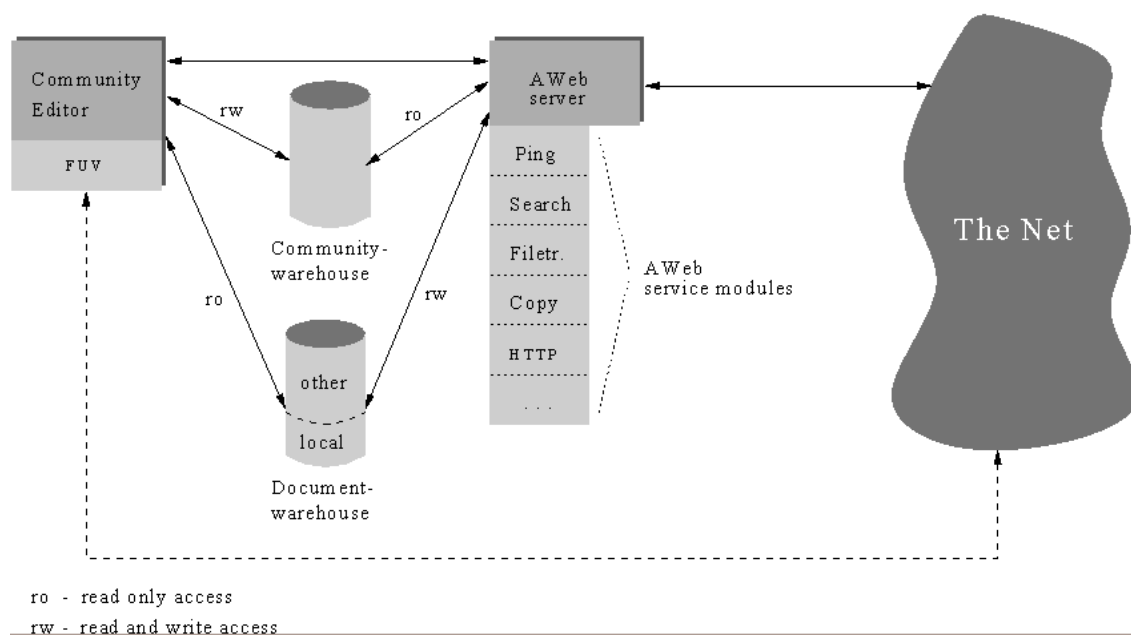
**Abbildung 2: Zeit für die Erzeugung eines globalen Hypercube**

In dem oben gezeigten Diagramm verlaufen die Graphen für die einzelnen Dimensionen fast linear. Die leichte Abweichung zum Ende hin ist durch die längeren Suchzeiten zu

erklären, die entstehen, wenn die Anzahl der zur Verfügung stehenden Knoten kleiner wird.

#### 4. AWeb – ein dezentrales Informationssystem

Es gibt bereits mehrere Projekte, die die Communities im Internet nutzen. Jedes benutzt dieses Konzept aber nur, um eine bestimmte Idee zu verwirklichen. Gnutella [11] z.B. nutzt die Communities, um ein dezentrales *files-sharing* System aufzubauen und Freenet [6] stellt auf der Basis der Community-Idee ein Informationssystem ähnlich dem World Wide Web bereit. Das Hauptziel des AWeb-Projektes dagegen ist es, eine reale Versuchsumgebung für die Untersuchung der Eigenschaften von Communities zu schaffen. Es soll zum einen ein offenes Testfeld bereitstellen, zum anderen aber auch Möglichkeiten, wie sie von verteilten Betriebssystemen wie z.B. dem WOS [18] bekannt sind, bereitzustellen.



**Abbildung 3: Aufbau einer AWeb-Knotens.**

Als konkrete Anwendung wurde eine verteilte, auf der Community-Struktur basierende Bibliothek für wissenschaftliche Veröffentlichungen gewählt. Diese soll dazu genutzt werden, um Anwendern die Möglichkeit zu geben, auf fremde Dokumente zuzugreifen und eigene Dokumente in dieser Bibliothek zu veröffentlichen, d.h. anderen zugänglich zu machen. Ein weiterer wichtiger Punkt ist die Verwaltung der Dokumente mit Hilfe einer graphischen Oberfläche und dem oben erwähnten Konzept der vertikalen und horizontalen Communities. Das AWeb-Programmpaket ist vollständig mit Hilfe der Programmiersprache Java [14] implementiert, um eine gute Portabilität zu erreichen.

Ein AWeb-Knoten besteht aus zwei Hauptkomponenten – dem *Community-Server* und dem *Community Editor*. Der Server beantwortet alle Anfragen anderer Mitglieder der Community und ist gleichzeitig der Kommunikationsclient des lokalen Knotens. Der Server benutzt die oben beschriebenen Message Chains für alle Kommunikationsaufgaben. In der folgenden Abbildung ist der Aufbau eines AWeb-Knotens dargestellt.

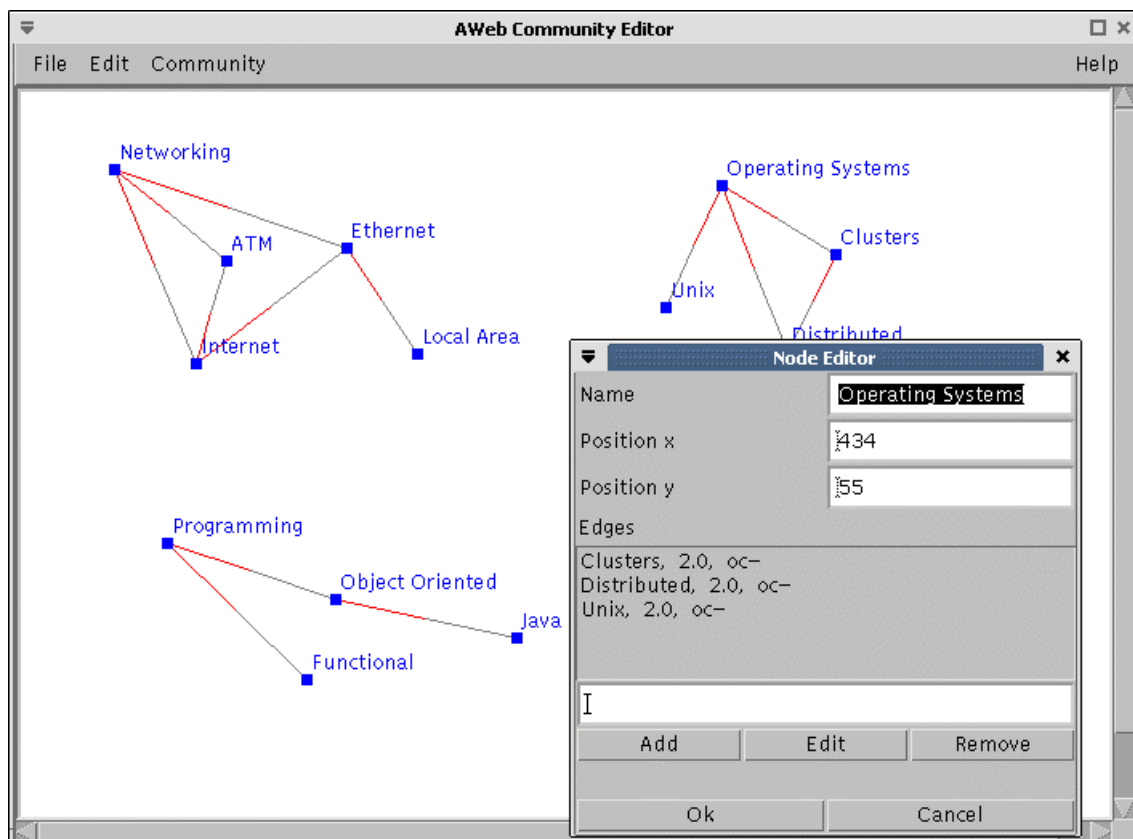
Die einzelnen Dienste, die der Server anbietet, sind als Module realisiert und können auch während des Betriebs hinzugefügt werden. Das macht den Server flexibel und erweiterbar. Die folgenden Standardmodule sind bereits im Server enthalten:

- *Ping*

Das Ping-Modul ist ein grundlegender Service im Aweb, welcher es den einzelnen Knoten ermöglicht, andere Maschinen im Netzwerk zu kontaktieren und herauszufinden, ob diese ebenfalls einen AWeb-Server aktiviert haben oder nicht.

- *Search*

Das Modul für die Suche in der Community ist ebenfalls sehr wichtig. Es wird verwendet, um eingehende Suchanfragen zu beantworten. Um eine Anfrage beantworten zu können, greift es auf die lokalen Community-Warehouses zu.



**Abbildung 4: Die graphische Benutzeroberfläche des AWeb Editors**

---

Die horizontalen und vertikalen Communities werden mit dem Community-Editor verwaltet. Er unterstützt den Anwender bei der Strukturierung der Community-Graphen sowie bei der Wahl der Schlüsselworte. Dabei können auch Zugriffsbeschränkungen angegeben werden, um diese vom Nutzer geschaffene Struktur nicht oder nur an bestimmte Gruppen von Anwendern weiterzugeben. Der Editor stellt auch das Interface für die Suchfunktion sowie für das Einfügen eigener Dokumente in die Bibliothek bereit. Alle Daten, die für die Organisation der Community benötigt werden, sind im *Community-Warehouse* gespeichert. Die lokal gespeicherten Dokumente befinden sich im *Document-Warehouse*. Dieses ist in zwei Bereiche unterteilt. Einen für die Dokumente, die der lokale Benutzer zur Verfügung stellt und einen für die vom Nutzer heruntergeladenen fremden Dokumente. Links, die auf fremde Dokumente weisen, werden in bestimmten Abständen auf Aktualität geprüft und bei Bedarf aktualisiert.

Um Mitglied in der AWeb-Community zu werden, ist neben der erwähnten Software auch das Wissen um einen weiteren Knoten notwendig, der bereits zur Community gehört. Durch den Suchmechanismus werden weitere Mitglieder gefunden und in dem lokalen *Community-Warehouse* gespeichert. Durch diese Einträge werden die Communities des Nutzers aufgebaut.

## **5. Zusammenfassung**

Zusammenfassend kann gesagt werden, dass Communities interessante Eigenschaften besitzen, die durchaus das Potential haben, die zukünftigen Entwicklungen im Bereich der Computernetzwerke und der Verteilten Systeme zu beeinflussen. Diese Entwicklung befindet sich nach Meinung der Autoren noch im Anfangsstadium. Erste Untersuchungen zeigten, dass, basierend auf dem Community-Konzept, selbstanpassende und selbstorganisierende Systeme geschaffen werden können und diese somit an die jeweiligen Erfordernisse optimal anzupassen sind.

Durch spezielle Mechanismen zur Verwaltung der Communities und der Suche und Kommunikation innerhalb der Communities sind diese eine sehr flexible und gut handhabbare Grundlage für eine Reihe effizienter Werkzeuge und Arbeitsumgebungen. Am Beispiel des AWeb konnte dies gezeigt werden. Es verbindet die einfache Handhabung und Mächtigkeit vorhandener zentraler Client-Server-Systeme und die Flexibilität und Fehlertoleranz verteilter, dezentraler Architekturen.

## 6. Literatur

- [1] G. Babin, P. Kropf, and H. Unger. A Two-Level Communication Protocol for a Web Operating System (WOS). In *IEEE 24th Euromicro Workshop on Network Computing*, pages 934–944, Sweden, 1998.
- [2] T. Böhme and H. Unger. Search in the WOSNet. In *Distributed Computing on the Web (DCW)*, Rostock, Germany, 1998.
- [3] E. Bonabeau, G. Theraulaz, J. Deneubourg, S. Aron, and S. Camazine. Selforganization in social insects. *Trends in Ecol. Evol.* 188–193, 1997.
- [4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
- [5] G. Di Caro and M. Dorigo. AntNet: A Mobile Agents Approach to Adaptive Routing, 1997.
- [6] Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *ICSI Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, 2000.
- [7] Marco Dorigo, Vittorio Maniezzo, and Alberto Coloni. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. on Systems, Man, and Cybernetics–Part B*, 26(1):29–41, 1996.
- [8] Gary Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, August 20–23 2000.
- [9] L. N. Foner. YENTA: A multi-Agent Referral System for Matchmaking. In *First International Conference on Autonomous Agents*, Marina del Rey, California, 1997.
- [10] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *UK Conference on Hypertext*, pages 225–234, 1998.
- [11] Gnutella. [www.gnutellanews.com](http://www.gnutellanews.com), 2001.
- [12] Francis Heylighen. Collective Intelligence and its Implementation on the Web: Algorithms to Develop a Collective Mental Map. *Computational & Mathematical Organization Theory*, 5(3):253–280, 1999.
- [13] G. Hommel. Quality of Communication-Based Systems. In *International Workshop on Quality of Communication-Based Systems*, TU Berlin, Germany, 1994. Kluwer Academic Publishers.
- [14] Sun Microsystems Inc. The Java Language Specification, 2000. [java.sun.com/docs/](http://java.sun.com/docs/).

- 
- [15] J. A. Kaplan and M. L. Nelson. A Comparison of Queueing, Cluster and Distributed Computing Systems. NASA Technical Memorandum 109025, Langley Research Center, 1993.
- [16] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective, 1999.
- [17] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: Measurements, models, and methods. In T. Asano, H. Imai, D. T. Lee, S. Nakano, and T. Tokuyama, editors, *Proc. 5th Annual Int. Conf. Computing and Combinatorics, COCOON*. Springer-Verlag, 1999.
- [18] P. Kropf. Overview of the WOS Project. In SCS A. Tentner, editor, *ASTC High Performance Computing*, pages 350–356, San Diego, CA, 1999.
- [19] P. Kropf, J. Plaice, and H. Unger. Towards a Web Operating System. In *WebNet '97*, Toronto, 1997.
- [20] S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. *WWW8 / Computer Networks*, 31(11-16):1481–1493, 1999.
- [21] V. Menko, D. J. Neu, and Q. Shi. AntWorld: A Collaborative Web Search Tool. In Kropf et al., editor, *Distributed Communities on the Web (DCW)*. Springer Verlag Berlin, 2000.
- [22] Stanley Milgram. The small-world problem. *Psychology Today*, 1967.
- [23] D. Milojicic. Operating Systems - now and in the future. *IEEE Concurrency*, 1(7):12–21, 1999.
- [24] J. Plaice and P. Kropf. Intensional Communities. In World Scientific Press, editor, *Intensional Programming II*, Singapore, 2000.
- [25] F. Reynolds. Evolving an operating system for the Web. *IEEE Computer*, 9(29):90–92, 1996.
- [26] D. Tavangarian, P. Eschholz, M. Koch, C. Pitz, and S. Preuss. Hypercomputing: A Concept for a Network-based Computer Architecture. In *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '98)*, Las Vegas/Nevada, 1998.
- [27] H. Unger. Distributed Resource Location Management in the WOS. In SCS A. Tentner, editor, *ASTC High Performance Computing*, Washington, 2000.
- [28] H. Unger and T. Böhme. Distribution of information in decentralized computer communities. In A. Tentner, editor, *ASTC High Performance Computing*, Seattle, Washington, 2001.

- [29] H. Unger, P. Kropf, G. Babin, and T. Böhme. Simulation of search and distribution methods for jobs in a Operating System (WOS). In SCS A. Tentner, editor, *ASTC High Performance Computing*, pages 253–259, Boston, 1998.
- [30] M. Wulff and H. Unger. Message Chains as a new Form of Active Communication in the WOSNet. In A. Tentner, editor, *ASTC High Performance Computing*, Washington, 2000.
- [31] N. Deo and P. Gupta. *World Wide Web: A Graph Theoretic Approach*. CS TR-01-001, University of Central Florida, 2001.