

**Universität
Koblenz-Landau**
Abteilung Koblenz
Fachbereich 4 - Informatik

**Seminar:
Semantic Grid**

WS 2004/05

**Semantisches Matchmaking und
Semantische
Ressourcenbeschreibung**

Betreuung:

Prof. Dr. Steffen Staab
Dipl.-Inform. Bernhard Tausch

Daniela Schmitz
dschmitz@uni-koblenz.de

Inhaltsverzeichnis

1	Einleitung	3
2	Condor Matchmaker	3
3	Ontologie-basierter Matchmaker	4
3.1	Architektur des Ontologie-basierten Matchmakers	5
3.1.1	Ontologien	6
3.1.2	Domain Background Knowledge	7
3.1.3	Matchmaking Rules	8
3.2	Der Matchmaking-Prozess	8
3.3	Schwächen des Ontologie-basierten Matchmakers	9
4	Ontologie-basierter P2P-Ansatz	10
4.1	Ressourcenbeschreibung	10
4.1.1	T-Box	10
4.1.2	A-Box	11
4.2	Wissen	11
4.2.1	Ein Knoten tritt dem Netzwerk bei	12
4.2.2	Ein Knoten verlässt das Netzwerk	13
4.3	Matchmaking	13
4.4	Schwächen des Ontologie-basierten P2P-Ansatzes	14

Abbildungsverzeichnis

1	Architektur des Ontologie-basierten Matchmakers; nach[TDK03]	6
2	Beispiel: Ressourcen-Angebot [TDK03]	7
3	Beispiel: Ressourcen-Gesuch [TDK03]	7
4	Beispiel: Domain Background Knowledge [TDK03]	8
5	Der Matchmaking-Prozess [TDK03]	9
6	Lokaler Classification DAG [HHK04]	11
7	Das Wissen von Knoten 1 über das Konzept Athlon 32 [HHK04]	13

1 Einleitung

Das Grid ist eine Plattform, die es dynamischen, multi-institutionellen Organisationen ermöglicht Ressourcen zu teilen und Probleme koordiniert zu lösen. An einem Grid nehmen verschiedene geographisch verteilte Parteien teil.

Diese Parteien sind zum einen die Resource Provider, die Anbieter der Ressourcen.

Die andere Partei sind die Requester, die Kunden, die eine bestimmte Anwendung ausführen möchten und dazu eine Ressource benötigen, auf der diese Anwendung läuft.

Bevor diese bestimmte Anwendung des Requesters auf einer Ressource ausgeführt werden kann, muss als erstes eine Ressource gefunden werden, die den Ansprüchen der auszuführenden Anwendung gerecht wird.

Der Prozess der Ressourcensuche basierend auf Anforderungen der Anwendung wird Resource Matching oder Matchmaking genannt.

Die Voraussetzung für ein Matchmaking ist allerdings, dass sowohl die angebotenen Ressourcen, als auch die gesuchten Ressourcen (Requests) mit allen Eigenschaften und Beschränkungen beschrieben werden.

Nur so kann ein passendes Ressource-Request-Paar gefunden werden.

Kapitel 2 wird in diesem Zusammenhang, anhand des Condor Matchmakers, eine Einführung in den Bereich des traditionellen Matchmakings und der Ressourcenbeschreibung ohne Semantik gegeben.

In den Kapiteln 3 und 4 werden anschließend zwei semantische Matchmaking-Mechanismen, der Ontologie-basierte Matchmaker (OMM) und der Ontologie-basierte P2P-Ansatz vorgestellt.

2 Condor Matchmaker

Die Funktionsweise der traditionellen Matchmaker wird in diesem Kapitel anhand des Condor Matchmakers aufgezeigt. Die Beschreibung des Ressourcen-Gesuchs und Angebots geschieht in diesem Ansatz mit Hilfe von „Attribut-Wert“-Paaren. Hier ist die Beschreibung eines Ressourcenangebotes dargestellt.

Resource ClassAd:

```
[ Type = „Machine“; Name = „m1“; Disk = 30000; Arch = „INTEL“;
OpSys = „SOLARIS251“; ResearchGrp = „user1“, „user2“;
Constraint = member(other.Owner,ResearchGrp) &&& DayTime _ 18*60*60;
Rank = member(other.Owner,ResearchGrp) ]
```

Der Matchmaking-Mechanismus von Condor basiert auf bilateralem, symmetrischem und Attribut-basiertem Matching.

Bilateral bedeutet, dass von beiden beteiligten Parteien (Kunde und Provider) Einschränkungen gemacht werden können. Der Kunde kann in seinem Gesuch die in Frage kommenden Ressourcen anhand von Beschränkungen genau eingrenzen. Gleichzeitig hat aber auch jeder Anbieter die Möglichkeit seine Zugriffsbedingungen ausdrücken, also zu bestimmen, wer Zugriff auf seine Ressourcen

ce erhalten darf (Siehe Abbildung 2).

Der Matchmaker beachtet die Forderungen beider Parteien bei einem Match.

Syntax-Matching bedeutet, dass Condor nur die Syntax der Beschreibungen vergleicht. Die Vokabeln, welche im Angebot des Providers und im Gesuch des Kunden verwendet werden, mitsamt dahinter stehender Werte, werden syntaktisch miteinander verglichen (syntactic string/numeric equality).

Bei **symmetrischem Matchmaking** verwenden Anbieter und Kunde das gleiche Vokabular. Eine genaue Syntax der Attribute und Werte ist festgelegt, an die sich die beteiligten Parteien halten müssen.

Nachteile des Syntax Matchings:

Generell bringt das Matching basierend auf Syntax einige Probleme mit sich. Dieses Problem soll anhand eines Beispiels gezeigt werden.

In dem in Abbildung 2 dargestellten Angebot sind die Eigenschaften der Ressource genau beschrieben.

...

OpSys = „SOLARIS251“

...

Diese Ressource zeichnet sich unter anderem durch das Betriebssystem Solaris 251 aus.

Das Ressourcengesuch eines Kunden enthält nun die Anforderung:

OpSys = „UNIX“.

Da Solaris ein Unix-Betriebssystem ist, würde das oben beschriebene Angebot auf das Gesuch des Kunden passen. Condor jedoch verfügt nicht über dieses „Hintergrundwissen“ und aus diesem Grund kann bei einem reinen Syntax-Matching im oben genannten Beispiel kein Treffer gefunden werden.

Nachteil des symmetrischen Matchings:

Ein Nachteil des symmetrischen Matchings ist, dass sich Ressourcen-Anbieter und Kunde im Vorfeld auf bestimmte Vokabeln, nämlich die Attributnamen und Werte, einigen müssen, damit es überhaupt zu Treffern kommen kann. Daraus ergibt sich der Nachteil der schlechten Erweiterbarkeit.

Es ist schwierig neue Attributnamen und Werte in dieses Konzept einzuführen, da alle Teilnehmer über diese Veränderungen informiert werden müssen, und das gestaltet sich gerade in offenen Systemen, wie dem Grid, relativ schwierig. [TDK03]

3 Ontologie-basierter Matchmaker

Anhand des Condor-Matchmakers wurden in Kapitel 1 die Schwächen der traditionellen, symmetrischen und Atribut-basierten Matchmaking-Mechanismen aufgezeigt.

Um den Vorgang des Matchmakings besser und komfortabler zu gestalten, muss ein anderer Ansatz gefunden werden, der die Vorteile der traditionellen Matchmaker (wie Condor) übernimmt und sie um Hintergrundwissen erweitert. Die Autoren Tangmunarunkit, Decker und Kesselman stellen in [TDK03] einen Ontologie-basierten, flexiblen und erweiterbaren Ansatz zur „Grid resource selection“ vor. Dieser Ontologie-basierte Matchmaker (OMM) betreibt bilaterales, asymmetri-

ches und semantisches Matching. Im Folgenden werden diese Begriffe genauer erklärt.

Bilaterales Matching

Wie schon in Kapitel 2 beschrieben, haben hier beide Parteien die Möglichkeit bestimmte Einschränkungen zu machen.

Asymmetrisches Matching

Zur Beschreibung der Ressourcen haben die Entwickler unterschiedliche und voneinander unabhängige Ontologien mit Hilfe von RDF Schema entwickelt. Da die Ontologien voneinander unabhängig sind, ist es leichter diese Ontologien zu erweitern, da keine Absprache zwischen diesen Parteien mehr notwendig ist. Die Ontologien sind außerdem einfacher zu warten und zu verstehen als die sonst verwendeten Attributlisten.

Semantisches Matching

Ein Treffer zwischen Angebot und Gesuch wird hier nicht aufgrund von Sytax-Vergleichen vorgenommen, sondern basierend auf Semantik. Dazu wird mit Hilfe von Regeln ein Hintergrundwissen aufgebaut. Wann es zu einem Treffer zwischen Angebot und Gesuch kommt ist ebenfalls mit Hilfe von Regeln genau definiert.

Neben diesen Eigenschaften bietet der OMM weitere Möglichkeiten den Matchmaking-Prozess zu verbessern:

Matching preferences

Falls der Matching-Mechanismus mehrere Treffer hervorbringen sollte, gibt es die Möglichkeit eine Rangliste erstellen zu lassen, in der die Treffer anhand einer definierten Bedingung geordnet sind.

Integrity check

Der Matchmaker ist in der Lage sein „Wissen“ zu nutzen, um jeweils die angebotenen Ressourcen und die Gesuche daraufhin zu prüfen ob sie Fehler enthalten. Sollte ein Kunde in seinem Gesuch einen bestimmten Prozessor mit einer bestimmten Leistung wünschen, und sollte es diese Kombination nicht geben, so wird dieses Gesuch vom Matchmaker schon zu Beginn abgelehnt.

3.1 Architektur des Ontologie-basierten Matchmakers

Der Ontologie-basierte Matchmaker besteht aus drei Komponenten, welche in der oben genannten Beschreibung der Features bereits kurz erwähnt wurden.

Diese Komponenten sind:

1. die Ontologien,
sie erfassen das domain model und beinhalten die Vokabeln um Ressourcen zu beschreiben
2. das Domain Background Knowledge,
es bietet zusätzliches Wissen über die Domäne
3. die Matchmaking Rules,
sie beschreiben, wann ein Match zwischen Angebot und Gesuch vorliegt

Abbildung 1 zeigt die Beziehung zwischen diesen Komponenten.

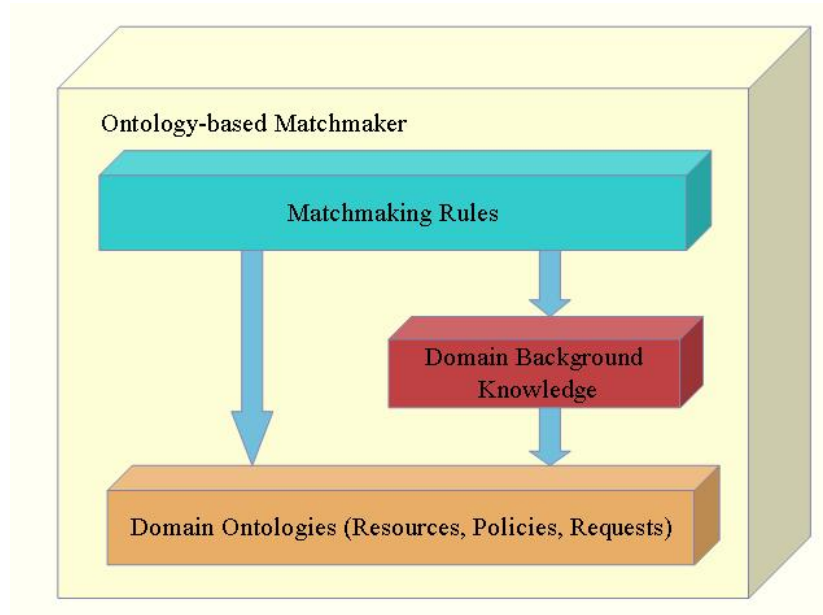


Abbildung 1: Architektur des Ontologie-basierten Matchmakers; nach [TDK03]

Das Background Knowledge nutzt die Vokabeln der Ontologien um die Background-Information zu bilden. Die Matchmaking Rules nutzen die Ontologien und das Background Knowledge um den Matchmaking-Prozess durchzuführen. In den nächsten Abschnitten werden diese Komponenten genauer beschrieben.

3.1.1 Ontologien

Wie bereits erwähnt betreibt der OMM asymmetrisches Matching. Dazu haben die Entwickler unterschiedliche, voneinander unabhängige Ontologien modelliert.

Diese Ontologien sind:

- Resource Ontology
- Resource Request Ontology
- Policy Ontology

Jede dieser Ontologien definiert Objekte, die Eigenschaften dieser Objekte und Beziehungen zwischen diesen Objekten.

Resource Ontology:

Mit Hilfe der Ressourcen Ontologie kann ein Anbieter seine Ressourcen beschreiben, ihre Fähigkeiten und die Beziehungen zwischen ihnen. (Siehe Abbildung 2)

Resource Request Ontology:

Die Resource Request Ontologie dient dazu ein Ressourcengesuch zu beschreiben, mit Angaben zum Verfasser des Gesuchs, Charakteristiken des Gesuchs

und Anforderungen. (Siehe Abbildung 3)

Property Names	Property Values
UnitaryComputer.Name	"Almaak.usc.edu"
UnitaryComputer.AuthorizedGroup	"rcf@usc.edu"
UnitaryComputer.NumberOfAvailableCPUs	64
UnitaryComputer.ComputerSystemProcessor.MaxClockSpeed	900
UnitaryComputer.HostedFileSystem.AvailableSpace	500
UnitaryComputer.RunningOS.OSType	"SunOS"
UnitaryComputer.RunningOS.Version	"5.8"
UnitaryComputer.RunningOS.FreeVirtualMemory	4000
UnitaryComputer.RunningOS.FreePhysicalMemory	4000
UnitaryComputer.RunningOS.MaxProcessCPUs	64
UnitaryComputer.RunningOS.MaxProcessMemorySize	2000

Abbildung 2: Beispiel: Ressourcen-Angebot [TDK03]

Property Names	Property Values
JobRequest.Name	"Request1"
JobRequest.Owner	"User1"
JobRequest.JobType	"MPI"
JobRequest.NumberOfResources	1
JobRequest.RequestResource.ResourceType	"ComputerSystem"
JobRequest.RequestResource.RankBy	"CPUClockSpeed"
JobRequest.RequestResource.RequiredOS.OSType	"Unix"
JobRequest.RequestResource.RequiredCPU.MinNumberCPUs	32
JobRequest.RequestResource.RequiredMemory.MinPhysicalMemory	1000
JobRequest.RequestResource.RequiredMemory.MinVirtualMemory	1000
JobRequest.RequestResource.RequiredFS.MinDiskSpace	200

Abbildung 3: Beispiel: Ressourcen-Gesuch [TDK03]

Policy Ontology:

Die Policy Ontologie bietet ein Modell um die Autorisierung und die Zugriffsvoraussetzungen zu beschreiben. Beispielsweise können die Accounts angegeben werden, die autorisierten Zugriff zu einem bestimmten Computersystem haben. Diese Funktion ist jedoch noch stark eingeschränkt.

3.1.2 Domain Background Knowledge

Das Domain Background Knowledge beinhaltet zusätzliches Wissen über die Domäne, welches nicht in der oder den Ontologien beschrieben ist oder dort nicht beschrieben werden kann. Wie in Abbildung 1 gezeigt, wird dieses Wissen während des Matchmaking-Prozesses benötigt. Das Wissen ist dargestellt in Form von Regeln. Diese Regeln verwenden die Vokabeln die in der Ontologie definiert sind. Um diese Regeln aufzustellen, wird eine Regel-Sprache TRIPLE verwendet, welche auf Horn-Logik und F-Logik basiert. Die in Abbildung 4 dargestellten Regeln bestimmen, welches Betriebssystem sich mit welchem anderen Betriebssystem „verträgt“ (compatible). Die Regeln definieren außerdem den


```

@gridBackground { // specifies grid background knowledge
Linux[rdfs:subClassOf->GR:OperatingSystem].
Unix[rdfs:subClassOf->GR:OperatingSystem].
Debian[rdf:type->Linux]. Redhat[rdf:type->Linux].
SunOS[rdf:type->Unix]. Linux[rdf:type->Unix].

// transitivity axiom
FORALL X,Y,Z X[compatibleWith->Z] <- X[compatibleWith->Y] AND Y[compatibleWith->Z].

// identity axiom
FORALL X X[compatibleWith->X].

//symmetry axiom
FORALL X,Y X[compatibleWith->Y] <- Y[compatibleWith->X].

FORALL X,Y,Z X[substitutes->Z] <- (Y[rdf:type->Z] and
X[substitutes->Y]) or X[compatibleWith->Z].
}

```

Abbildung 4: Beispiel: Domain Background Knowledge [TDK03]

Begriff „ersetzen“ (substitute) mit Hilfe des Begriffs „compatible“. Welches Betriebssystem durch welches ersetzt werden kann, ist eigentlich in der Ontologie beschrieben. Verträglichkeiten können nicht mit einer Ontologie beschreiben, deshalb werden sie mit Hilfe der Regeln definiert.

3.1.3 Matchmaking Rules

Die Matchmaking Rules bestimmen wann es zu einem Treffer zwischen einem Angebot und einem Ressourcengesuch kommt. Diese Regeln sind ebenfalls mit TRIPLE definiert. Matchmaking Rules nutzen die Ontologien und das Background Knowledge um den Matchmaking-Prozess durchzuführen.

3.2 Der Matchmaking-Prozess

Die am Matchmaking-Prozess teilhabenden Parteien sind:

- der Ontologie-basierte Matchmaker
- die Resource Provider
- die Resource Requester

Der genaue Ablauf ist in Abbildung 5 dargestellt.

1. Die Anbieter (Provider) senden periodisch ihre Ressourcen-Angebote (mit den speziellen Eigenschaften) an einen oder mehrere Matchmaker. Erhält ein Matchmaker eine solche Nachricht, nimmt er die angebotene Ressource in seine Liste mit verfügbaren Ressourcen auf. Die Ressourcen Anbieter haben später noch die Möglichkeit ihre Ressourcenangebote zu aktualisieren. Sie haben die Möglichkeit bestehenden Angebot upzudaten, falls sich etwas an der Ressource geändert haben sollte und sie haben die Möglichkeit den Matchmaker darüber zu informieren, dass eine Ressource nicht mehr zur Verfügung steht. In diesem Fall löscht der Matchmaker die zurückgezogene Ressource aus seiner Liste.

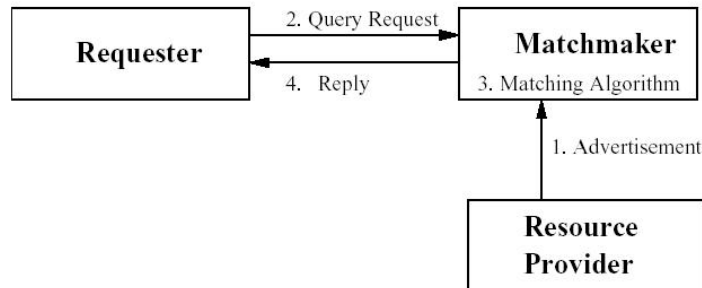


Abbildung 5: Der Matchmaking-Prozess [TDK03]

Für jedes Ressourcenangebot wird ein Timer gesetzt. Sollte in einer festgelegten Zeitspanne kein Update für dieses Angebot kommen, gilt die Ressource als veraltet und wird aus der Liste entfernt.

2. Ein Kunde sendet ein Gesuch an den Matchmaker, in welchem alle gewünschten Eigenschaften der Ressource genau beschrieben sind.
3. Wenn der Matchmaker das Ressourcengesuch erhält, aktiviert er den Matching Algorithmus, um eine Liste von potentiellen Ressourcen zu finden. Eventuell werden diese Treffer nach der Vorgabe (preference criteria) des Kunden sortiert.
4. Anschließend gibt der Matchmaker die Liste mit gefundenen Ressourcen an den Kunden zurück.

Neben dem Matchmaking-Service hat der Kunde auch die Möglichkeit einen so genannten Brokering-Service anzurufen. Dieser Brokering-Service baut auf dem Matchmaking-Service auf und enthält zwei zusätzliche Schritte, in denen sich der Matchmaker mit den Ressourcenanbietern zuvor abstimmt.

3.3 Schwächen des Ontologie-basierten Matchmakers

Ein Nachteil der traditionellen Matchmaking-Mechanismen ist neben der symmetrischen Ressourcenbeschreibung die Schwierigkeit neue Konzepte einzuführen, da über diese Veränderungen alle Teilnehmer informiert werden müssen.

Wie sieht es in dieser Hinsicht mit der Erweiterbarkeit des OMM aus?

Tangmunarunkit, Decker und Kesselman bezeichnen in [TDK03] die Ontologie-Modellierung als einen ständigen Prozess. Wenn die Vokabeln sich ändern, ändern sich auch Hintergrundwissen und Regeln.

Sie merken an, dass es einfach sei den Matchmaker zu erweitern, indem die Ontologien durch neue Vokabeln (request description models) und schlussfolgernde Regeln (mapping rules) ergänzt werden. Die Autoren geben weiter an, da die Ontologien auf Semantic Web Standards basieren, sei es leicht die Ontologien mit bestimmten Tools untereinander auszutauschen.

4 Ontologie-basierter P2P-Ansatz

Heine, Hovestadt und Kao sehen in der schlechten Erweiterbarkeit der bestehenden Matchmaking-Mechanismen ein großes Problem:

Der Begriff Ressource wird für viele Dinge verwendet: Hardware, Services usw. und das Wissen über diese Ressourcen wächst ständig. Neue Ressourceneigenschaften kommen hinzu, die wieder neu in Beziehung zu anderen Eigenschaften stehen.

Die Ontologien müssen ständig um dieses neue Wissen erweitert werden.

Wie kann aber sichergestellt werden, dass zu jeder Zeit alle Beteiligten, inklusive Matchmaker, Zugriff auf die komplette Ontologie der Domäne haben?

Heine, Hovestadt und Kao merken an, dass derzeit die Ontologien nur lokal verwendet werden. Aus diesem Grund stellen sie in ihrer Arbeit „Towards Ontology-Driven P2P Grid Resource Discovery“ [HHK04] eine Architektur vor, in der Domain-spezifisches Wissen bei Bedarf erworben werden kann.

Diesem Ansatz zugrunde liegt ein P2P-Netzwerk. Alle Teilnehmer dieses Netzwerks sind dafür verantwortlich den Ressourcen- Katalog zu verwalten.

Jeder Teilnehmer kann im Netz seine Ressourcen anbieten, Hintergrundwissen liefern und jeder Teilnehmer hat die Möglichkeit im Netz nach Ressourcen zu fragen.

Damit der Vorgang des Matchmaking funktionieren kann, ist hier, im Gegensatz zum Ontologie-basierten Matchmaker, keine allgemeingültige Ontologie notwendig. Jeder Teilnehmer hat seine eigene, eventuell auch unvollständige Ontologie, welche von den anderen Teilnehmern vervollständigt werden kann.

4.1 Ressourcenbeschreibung

Zur semantischen Beschreibung der Ressourcen wird in diesem Ansatz die Description Logik (DL) verwendet. Description Logiken eignen sich besonders gut dazu Wissen zu repräsentieren.

In DL-Systemen ist das Wissen in zwei Teile aufgeteilt:

1. T-Box (taxonomical)
2. A-Box (assertional)

4.1.1 T-Box

Die T-Box enthält das Hintergrundwissen. Zum Aufbau dieses Hintergrundwissens werden so genannten Konzepte definiert. Konzepte sind Teile eines Individuums der Domäne. Das bedeutet, ein Individuum der Domäne ist beispielsweise ein bestimmter Rechner. Mit Hilfe der Konzepte können nun alle Aspekte dieses bestimmten Rechners, wie Ressourcen-Typ, Betriebssystem und Speicherkapazität, beschrieben werden.

Weiter werden in der T-Box noch Beziehungen zwischen diesen Konzepten definiert. Die dortige Information ist als Graph dargestellt. Die Knoten in diesem Graphen sind dabei die Konzepte und die Kanten stellen die Beziehungen zwischen diesen Konzepten dar.

Wie oben beschrieben besitzt jeder Knoten seine eigene lokale Ontologie und damit seinen eigenen Graphen. In Abbildung 6 ist beispielsweise der Graph des

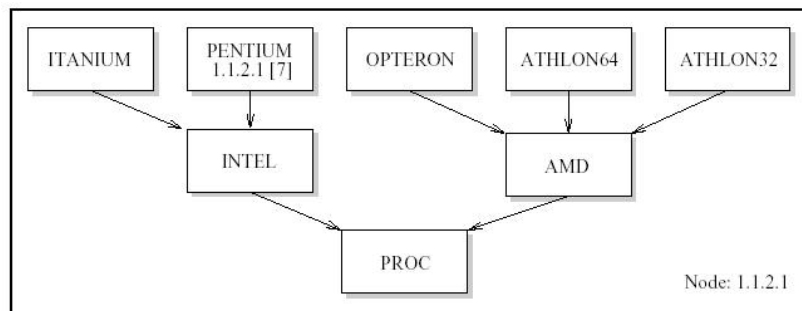


Abbildung 6: Lokaler Classification DAG [HHK04]

Knotens 1.1.2.1 dargestellt.

Proc ist dabei ein Superkonzept von Intel.

Pentium ist das Subkonzept von Intel.

Jeder Knoten kann bei Bedarf seinen eigenen Graphen um neue Konzepte erweitern. Dieses Wissen um neue Konzepte, also um neues Hintergrundwissen, wird später den anderen Teilnehmern des Netzwerkes mitgeteilt (Siehe Kap 4.2).

4.1.2 A-Box

Die A-Box repräsentiert konkretes Wissen über Individuen in der Domäne, also Wissen über eine existierende Ressource. Die A-Box wird dazu verwendet diese Ressourcen zu beschreiben.

Ein Teilnehmer des Netzwerkes beschreibt seine Ressourcen indem er A-Box-assertments macht, d.h. er bildet Instanzen von Konzepten.

Dieser Vorgang wird an einem Beispiel (siehe Abbildung 6) erläutert:

Jeder Anbieter gibt seinen Ressourcen, welche er anderen Teilnehmern anbieten möchte, eine fortlaufende Nummer, eine ID.

Der Knoten 1.1.2.1 besitzt eine Ressource mit der ID 7 vom Typ Pentium, die er anbieten möchte.

Knoten 1.1.2.1 signalisiert dies durch die „7“ in dem Pentium-Konzept.

Die Ressource 7 ist also eine Instanz dieses Konzepts Pentium.

Alle Eigenschaften einer Ressource können beschrieben werden indem Instanzen von den betreffenden Konzepten gebildet werden.

4.2 Wissen

Jeder Knoten kann seine eigene unvollständige Ontologie besitzen.

Indem das Wissen von allen Teilnehmern, also A-Box und T-Box-Wissen, über das Netzwerk verteilt wird, können die fehlenden Teile geliefert werden. Dieses Wissen wird jedoch nicht im Netz geflutet, also an alle Teilnehmer weitergeleitet, sondern es wird immer nur der Knoten informiert, den dieses Wissen betrifft.

Jeder Knoten speichert Informationen über ein Konzept (abhängig vom DHT-Algorithmus).

Dieser Vorgang wird wieder anhand eines Beispiels (siehe Abbildung 7) demonstriert.

Knoten 1 (Peer 1) ist verantwortlich für die Speicherung der Informationen, wel-

che das Konzept „ATHLON 32“ betreffen. Zu „seinem“ Konzept speichert jeder Knoten

- im T-Box-Part: das zugehörige Superkonzept
- im A-Box-Part: alle Ressourcen der Teilnehmer, die Instanzen dieses Konzepts sind

Der A-Box-Part ist eine Liste von Ressourcen, welche Instanzen des zugehörigen Konzepts sind. Für jede Ressource wird die IP Adresse des Knotens gespeichert, welcher diese Ressource anbietet und die ID der Ressource.

„ IP_1 [resources of IP_1], IP_2 [resources of IP_2], ..., IP_n [resources of IP_n]“.

Die Konzepte werden über das Netzwerk mit Hilfe des Distributed Hash Table (DHT) Algorithmus verteilt, auf dem dieses Netzwerk basiert. Dieser Algorithmus erlaubt es mit Hilfe eines Schlüssels einen Netzteilnehmer zu lokalisieren. Wenn als Schlüssel ein bestimmtes Konzept eingesetzt wird, so ist es immer möglich den Teilnehmer zu erreichen, der Informationen über dieses bestimmtes Konzept vorhält.

Im Anschluss werden zwei Szenarien beschrieben, die verdeutlichen wie die Informationsweiterleitung im Netzwerk abläuft.

4.2.1 Ein Knoten tritt dem Netzwerk bei

Tritt ein Knoten dem Netzwerk bei, so muss er zuerst seinen lokalen Graphen veröffentlichen. Wie oben beschrieben sendet er dabei nicht sein ganzes Wissen an alle Netzteilnehmer, sondern er informiert nur bestimmte Knoten.

Der neu beigetretene Knoten sendet Nachrichten an die Knoten,

- die über neue Super-Konzepte informiert werden müssen
- die für die Konzepte zuständig sind, von denen der „neue“ Knoten Instanzen hat. Mit diesem zweiten Schritt ist es dem „neuen“ Knoten möglich, seine eigenen Ressourcen an betreffender Stelle im Netz anzubieten.

Erhält ein Knoten eine Nachricht eines anderen Knotens, muss er die darin angegebenen Änderungen vornehmen und das neue Konzept, das Superkonzept und die Instanzen aufnehmen. Daraufhin muss dieser Knoten wiederum andere betroffene Knoten über die getätigten Veränderungen informieren.

In zwei Fällen werden Nachrichten an andere Knoten geschickt:

1. trägt der Knoten 1 (aus Abbildung 7) in seinem A-Box-Part eine neue Instanz ein, so muss er eine Nachricht mit dieser Information an alle Knoten verschicken, die zuständig für die Superkonzepte des eigenen Konzeptes sind. In diesem Fall an den Knoten der zuständig ist für das Konzept ATHLON.
2. erfährt der Knoten von einem neuen Superkonzept des eigenen Konzeptes, so muss er alle Instanzen, die er für dieses Konzept gespeichert hat an den Knoten senden, der das neue Superkonzept speichert.

Concept: ATHLON32		Peer 1
A-Box	1.1.3.7	[1,1,1,0,1,0,1,1]
	1.1.3.9	[1,0,0,0,1,0,0,1]
T-Box	ATHLON	

Abbildung 7: Das Wissen von Knoten 1 über das Konzept Athlon 32 [HHK04]

Ein Knoten antwortet immer auf eine Nachricht, indem er eine Nachricht zurückschickt, die die Liste von Superkonzepten enthält die er über seines eigenes Konzept kennt.

Dadurch kann der ursprüngliche Knoten Super-Superkonzepte löschen, die er fälschlicherweise als direktes Superkonzept gespeichert hat.

Die Informationen über Veränderungen werden so lange unter den Netzteilnehmern ausgetauscht, bis ein stabiler Zustand erreicht ist.

4.2.2 Ein Knoten verlässt das Netzwerk

Möchte ein Teilnehmer das Netzwerk verlassen, so darf er nicht plötzlich aussteigen, sondern muss die anderen Teilnehmer informieren.

Auch bei diesem Schritt werden wieder nur die „betroffenen“ Knoten benachrichtigt.

An dem T-Box-Wissen der anderen Knoten ändert sich nichts. Lediglich am A-Box-Wissen mancher Knoten im Netz müssen Veränderungen vorgenommen werden, wenn ein Knoten sein Ressourcenangebot zurückzieht. Die entsprechenden Instanzen in den Konzepten werden gelöscht.

4.3 Matchmaking

In den Kapiteln 4.1 und 4.2 wurde darauf eingegangen, wie die semantische Ressourcenbeschreibung in diesem Ansatz gelöst ist. In diesem Kapitel wird kurz der Vorgang des semantischen Matchmakings beschrieben.

Wenn ein Knoten eine Ressource mit dem Betriebssystem UNIX sucht, dann bestimmt dieser Knoten (mit Hilfe des Algorithmus) zuerst den Teilnehmer, der verantwortlich für dieses bestimmte Konzept UNIX ist.

Der Knoten bittet diesen Teilnehmer um eine Liste mit A-Box-Einträgen zu diesem UNIX-Konzept.

In dieser A-Box stehen die Adressen alle Teilnehmern, die Instanzen für dieses Konzept haben. Anschließend setzt sich der suchende Knoten mit diesen Knoten direkt in Verbindung, um nach der Verfügbarkeit dieser Ressourcen zu fragen.

Im Allgemeinen möchte ein Knoten seine gesuchte Ressource nicht nur durch eine Eigenschaft klassifizieren, sondern ein genaueres Ressourcengesuch abgeben. Ein komplexes Gesuch wird formuliert, indem Konzepte (gewünschte oder nicht gewünschte Eigenschaften) mit UND, ODER und NICHT verknüpft werden. Im Anschluss werden, wie oben beschrieben, Instanzenlisten dieser Konzepte

erfragt und die Formel nach der gesuchten Ressource hin ausgewertet.

4.4 Schwächen des Ontologie-basierten P2P-Ansatzes

Die Entwickler geben an, dass dies nur die ersten Schritte zu einem System mit verteiltem „resource matching“ basierend auf einem P2P-Netzwerk mit Semantik sind.

Zukünftig wollen sie an weiteren Verbesserungen arbeiten:

- Die Abfragesprache soll besser gestaltet werden.
- Es wird immer davon ausgegangen, dass sich ein Knoten abmeldet, bevor er das Netzwerk verlässt. Das kann nicht immer sichergestellt werden. Auch hierfür muss eine Lösung gefunden werden.
- Es soll möglich sein die Ergebnisse nach einem Kriterium zu ordnen (ranking).
- Bis jetzt bleibt das gesamte Wissen aus den T-Boxen immer erhalten, derzeit wird an einer Garbage Collection gearbeitet, um dieses Problem zu lösen.
- Ein weiteres Problem, welches in allen Netzwerken ohne zentrale Administration auftritt, ist die Sicherheit. Wie kann verhindert werden, dass ein Teilnehmer absichtlich oder unabsichtlich falsche Informationen im Netzwerk zu verbreitet?

Literatur

- [HHK04] *Heine, F.; Hovestadt, M.; Kao, O. (2004): Towards Ontology-Driven P2P Grid Resource Discovery, <http://wwwcs.upb.de/pc2/papers/files/436.pdf>*
- [TDK03] *Tangmunarunkit, H.; Decker, S.; Kesselman, C. (2003): Ontology-based Resource Matching in the Grid - The Grid meets the Semantic Web, In Proceedings of SemGRID 2003, <http://epicenter.usc.edu/docs/iswc03.pdf>*