



# Das semantische Netz mit DAML

- Einführung in DARPA Agent Markup Language -

Veranstaltung: Software-Agenten  
Veranstalter: Institut für Informatik  
Dozent: Prof. Dr. Lippe  
Semester: WS 2001/2002

Autor: Dirk Redbrake (redbrak@uni-muenster.de)

---

|  |           |
|--|-----------|
| <b>1 EINLEITUNG</b>                            | <b>2</b>  |
| <b>2 ENTWICKLUNG VON DAML</b>                  | <b>2</b>  |
| 2.1 DARPA                                      | 2         |
| 2.2 Das DAML-Programm                          | 3         |
| 2.3 Ziele                                      | 3         |
| <b>3 EINSATZ VON AGENTEN</b>                   | <b>4</b>  |
| <b>4 DAML-BEISPIEL FÜR EINE ONTOLOGIE</b>      | <b>5</b>  |
| 4.1 Header                                     | 5         |
| 4.2 Klassen                                    | 5         |
| 4.3 Properties                                 | 6         |
| 4.4 Property restrictions                      | 6         |
| <b>5 ANWENDUNGEN</b>                           | <b>7</b>  |
| 5.1 Suchmaschine der DAML-Homepage             | 7         |
| 5.2 Wall Street Journal (zur Zeit noch fiktiv) | 8         |
| 5.3 DAML Crawler                               | 8         |
| 5.4 PalmDAML                                   | 9         |
| 5.5 DAML Viewer                                | 14        |
| 5.6 HyperDAML                                  | 15        |
| <b>6 ZUSAMMENFASSUNG</b>                       | <b>16</b> |
| <b>7 LITERATUR</b>                             | <b>17</b> |

# 1 Einleitung

DAML ist die DARPA Agent Markup Language und wurde als Erweiterung der Extensible Markup Language und des Resource Description Framework entwickelt. Technische Entwicklerteams befassen sich jedoch nicht nur mit der Sprache selbst, sondern entwickeln auch Anwendungen, welche die Nutzung von DAML erleichtern sollen. Zum heutigen Stand muss man jedoch vom DAML + OIL Standard sprechen, da DAML eine US-amerikanische Entwicklung ist, während OIL (Ontology Interference Layer) das europäische Gegenstück dazu ist. Auf OIL wird an dieser Stelle nicht weiter eingegangen. Festzuhalten ist, dass der heutige Standard eine Zusammenführung beider Sprachen ist. Das Resultat dieses noch jungen Standards erweitert XML und RDF um verbesserte Möglichkeiten, Ontologien zu beschreiben, d.h. Objekte und ihre Beziehungen untereinander zu definieren. Die sogenannte „DAML Vision“ besteht letztendlich darin, aus dem World Wide Web ein Semantic Web zu machen, in dem nicht länger mehrdeutige Daten dargestellt werden, sondern durch Beschreibung der Semantik eindeutige Informationen gewonnen werden, die nicht von jedem Nutzer individuell herausgearbeitet werden müssen.

## 2 Entwicklung von DAML

### 2.1 DARPA

Die Defense Advanced Research Projects Agency ist die zentrale Forschungs- und Entwicklungsorganisation für das US-amerikanische Department of Defense (DoD). Die DARPA wurde 1958 im Kalten Krieg gegründet und ist in Arlington, Virginia beheimatet. Finanziert wird diese Organisation vor allem durch einen Steuergelderfond. Teilnehmer an den Forschungsprojekten sind zumeist Wissenschaftler und Ingenieure, die nicht in Diensten der Regierung stehen. Die Projekte, die besonders für das Militär von großem Interesse sind, gelten nach eigenen Angaben als „äußerst risikobehaftet aber mit extrem großem Pay-Off“ und sollen dazu dienen, den Vereinigten Staaten einen technischen Vorsprung zu gewährleisten, der „weit vor der normalen technischen Evolution liegt“. So war auch die Entwicklung des Internets eine Initiative der DARPA. Ein durchschnittliches DARPA-Projekt ist auf drei bis fünf Jahre begrenzt und hat ein Jahresbudget von 2,5 – 10 Mio. US-Dollar. Ferner sind im Schnitt zehn Firmen oder andere Institutionen sowie zwei Universitäten an einem Projekt beteiligt. Die Verlängerung eines Projektes ist im Normalfall nicht vorgesehen.

## 2.2 Das DAML-Programm

Das DAML-Programm wurde von der DARPA offiziell im August 2000 in Boston gestartet. Somit ist mit einer Projektlaufzeit bis etwa 2004 zu rechnen. Mittlerweile arbeiten 16 Entwicklerteams an Programmen zur Nutzung von DAML. An den derzeit laufenden 21 DAML-Forschungsprojekten sind zehn Firmen und Institutionen sowie überdurchschnittliche sieben Universitäten beteiligt. Das DAML-Projekt gilt als „unclassified“, so dass Rechenschaftsberichte und Forschungsergebnisse frei zugänglich sind. Das Budget des Projektes deutet ebenfalls auf eine überdurchschnittliche Bedeutung hin. So wird das Budget von 2001, bei dem es sich auf 13,1 Mio. Dollar belief, in diesem Jahr auf 15,9 Mio. Dollar aufgestockt.

Die Zuständigkeit des DAML-Programms wechselte am 1.11.2001 vom Information Technology Office (ITO) zum Information Exploitation Office (IXO). Diese Abteilung entwickelt Technologien für Sensor- und Informationssysteme wie z.B. intelligente Zielsysteme.

## 2.3 Ziele

Die Hauptziele des Projektes liegen darin, eine interoperable, agenten-basierte und heterogene Integration verschiedener militärischer Quellen und Systeme in Echtzeit zu erreichen. Dabei werden insbesondere InteLink und C2Link („command and control“) erwähnt, auf die an dieser Stelle jedoch nicht weiter eingegangen wird. Um ein Ergebnis in Echtzeit zu erzielen, ist es zunächst erforderlich, eine große Menge von DAML-Statements zu entwickeln. Daher ist man bemüht, das derzeitige Archiv von 850000 DAML-Statements in diesem Jahr auf 1,5 Mio. Statements zu vergrößern, damit empirische Laufzeittests aussagekräftiger werden.

Ein weiteres Ziel ist die Entwicklung von Tools, die DAML transparent und sinnvoll in Webseiten und andere Informationsquellen einbetten. Mit diesen Tools sollen agenten-basierte DAML-nutzende Programme entwickelt, instanziiert, ausgeführt und ausgetestet werden. Durch empirische Tests wird die Steigerung der Produktivität gemessen, die durch diese Tools erreicht werden soll. Einige dieser Tools sollen dann zur Agenten-Herstellung, Behandlung militärischer Probleme usw. eingesetzt werden. Diese Agenten sollen komplexe semantische Strukturen erkennen können. Bisher war es für Agenten zu schwierig, im Internet Semantiken und Inhalte zu erkennen und zu strukturieren. Das gleiche gilt für Informationsstrukturen bei Sensordaten. Es ist daher anzunehmen, dass DAML langfristig nicht nur im Internet, sondern auch als agenten-basiertes Kommunikationsmedium verschiedener militärischer Technologien wie Satelliten und Abwehrsysteme nutzbar ist.

Inhaltlich geht es auch um die Erstellung eigener oder die Erweiterung bestehender maschinen-lesbarer Ontologien mit DAML. Ein Bottom-up-Design von semantischen Bedeutungen bietet die Möglichkeit, diese Ontologien auf höheren semantischen Ebenen mit anderen Information Communities zu teilen. Daher besteht ferner ein großes Interesse an Mechanismen zur ausdrücklichen Präsentation von Services, Prozessen, Geschäftsmodellen, aber auch zur Darstellung von Informationen, die von Programmen oder Sensoren nicht ausdrücklich freigegeben werden. DAML ermöglicht es, an Objekte Informationen aller Art wie Funktionsbeschreibungen etc anzuhängen, die ausgelesen und semantisch eindeutig identifiziert werden können.

### 3 Einsatz von Agenten

DAML wird vor allem für den Einsatz mit Software-Agenten, die dynamisch Informationsquellen identifizieren und verstehen können, entwickelt. Agenten sollen bezüglich ihrer Semantik interoperabel sein. Eine mögliche Lösung ist das Verpacken von Daten in Notationen, die lesbar und interpretierbar sind. Aus Datenaustausch wird Informationsaustausch. Agenten können verschiedene Quellen wie Web-Seiten, Datenbanken, Programme, Sensoren, etc. zusammenbringen („information fusion“). Agenten sollen demnach eigenständig in verschiedenen Informationsquellen gewünschte Objekte suchen und identifizieren können und dabei eventuell auch Informationen auslesen. DAML hilft diesen Agenten, Informationen zu identifizieren sowie an andere Agenten weiterzuleiten, um dadurch die Interoperabilität zu erhöhen. Der heutige technische Stand ermöglicht es Agenten noch nicht, DAML-Resources völlig eigenständig zu finden. Es ist zur Zeit noch notwendig, dem Agenten vorher mitzuteilen, auf welchen URIs er suchen soll. Damit Agenten auch eigenständig in gewünschter Echtzeit weit verteilte DAML-Statements identifizieren können, wird man während der Entwicklung zu Testzwecken das oben bereits erwähnte große Archiv benötigen. Wahrscheinlich wird man als Endresultat des DAML-Projektes ein hierarchisch strukturiertes Agentennetzwerk entwickeln, das in der Lage ist, räumlich verteilte und mit unterschiedlichen Sprachen und Betriebssystemen operierende Informationsquellen semantisch fehlerfrei und maschinen-unabhängig kommunizieren zu lassen.

## 4 DAML-Beispiel für eine Ontologie

Die aktuelle Version des DAML + OIL – Standards stammt aus dem März 2001. Syntaktisch ist es eine Markup Language wie RDF und XML, so dass hier nur einige Beispiele angeführt werden, um einen Eindruck von DAML zu vermitteln. Der Standard besteht aus 35 Elementen. Am 27.11.2001 waren bereits 169 Ontologien verfügbar, in denen 49 Properties verwendet wurden. Es ist davon auszugehen, dass diese Zahlen schnell ansteigen werden. Exemplarisch soll hier die Verwendung von DAML anhand einer Ontologie demonstriert werden. Sie enthält beliebig viele oder auch keinen Header und ebenso beliebig viele Klassen, Properties und ihre Restriktionen sowie Instanzen. Beispiele für Instanzen fehlen hier bewusst, weil sie durch RDF-Konstrukte realisiert werden.

### 4.1 Header

Ein Beispiel für eine Ontologie mit einem Header wäre z.B.:

```
<Ontology rdf:about="">
  <versionInfo>$Id: reference.html,v 1.10 2001/04/11 16:27:53
                                mdean Exp $</versionInfo>
  <rdfs:comment>An example ontology</rdfs:comment>
  <imports rdf:resource="http://www.daml.org/01/03/daml+oil"/>
</Ontology>
```

Bis auf die `comment` – Anweisung handelt es sich um DAML-Befehle. `Ontology` bildet den Rahmen für die ganze Ontologie. Mit `versionInfo` kann ein beliebiger Text angegeben werden, der zwar eine Aussage über die DAML-Version machen soll, aber er hat keine logische Bedeutung für die Ontologie. Mit `imports` – Anweisungen können Ontologien transitiv importiert werden. Wenn A B und B C importiert, so importiert A sowohl B als auch C. Wenn A B einerseits und B A andererseits importiert, dann sind A und B äquivalent.

### 4.2 Klassen

Im Vordergrund steht bei DAML die Entwicklung von Klassen, mit denen Objekte der sogenannten object domain beschrieben werden können. Unter der object domain versteht man die Menge aller Einzelobjekte, die mit DAML oder RDF beschreibbar sind. Hingegen findet die datatype domain, die aus XML-Datentypen besteht, weniger Verwendung. Sie können dennoch mit dem DAML-Befehl `datatype` unter Angabe der URI verwendet werden.

Ein vereinfachtes Beispiel für eine DAML-Klasse (durch `Class` - Befehl), die stets eine Unterklasse von RDF-Klassen bildet, könnte so aussehen:

```
<daml:Class URI_name>
  <daml:equivalentTo
    rdf:resource="http://www.daml.org/daml#Person"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#name"/>
      <daml:toClass
        rdf:resource="http://www.w3.org/XMLSchema#string"/>
      <daml:maxCardinality>1</daml:maxCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Auf die Property-Anweisungen wird im nächsten Abschnitt eingegangen. Auf Klassenebene ist die `equivalentTo` - Anweisung zu nennen, die auf die Klasse `Person` hinweist, die semantisch äquivalent zu dieser Beispiel-Klasse ist. Es gibt weitere DAML-Befehle, die die semantische Beziehung zwischen Klassen wie gemeinsame Instanzen definieren können. Der `subClassOf` - Befehl stammt aus RDF und wird in DAML-Dokumenten oft verwendet.

### 4.3 Properties

Die vielfältigen Möglichkeiten zur Beschreibung von Properties stammen überwiegend aus dem RDF-Befehlsschatz. Daher werden an dieser Stelle einige DAML-Befehle nur genannt und kurz erläutert. So kann mit `samePropertyAs` eine Property angegeben werden, die äquivalent zur aktuellen Property ist. Wie schon bei Klassen gibt es auch für Properties eine `equivalentTo` - Anweisung, die auf eine semantisch äquivalente Property verweist. Eine `TransitiveProperty` ist transitiv ( Instanzen:  $(x,y), (y,z) \rightarrow (x,z)$  ), während bei einer `UniqueProperty` zwei Instanzen niemals den gleichen Wert annehmen dürfen.

### 4.4 Property restrictions

Die mit `subClassOf` definierte anonyme Klasse aus dem Beispiel aus Punkt 4.2 beinhaltet einige Befehle, die Restriktionen der Klasse in Bezug auf Properties beschreiben. Sie werden

innerhalb der `Restriction` – Anweisung beschrieben, die in anonymen Unterklassen festgelegt werden. Mit `onProperty` werden die Properties bzw. URIs definiert, auf die sich die Restriktion bezieht. In diesem Beispiel ist es ein Name. Durch `toClass` wird die Klasse ausgewählt, die alle Werte enthält, die die Property Name annehmen kann. In diesem Fall muss ein Name immer ein String nach XML-Definition sein. Es gibt noch weitere DAML-Befehle, die Beziehungen zwischen Properties und ihren zulässigen Werten festlegen. Der letzte hier angesprochene Befehlstypus beschäftigt sich mit Kardinalitäten. In diesem Fall beschreibt `maxCardinality`, dass die Beispielklasse höchstens einen Namen haben darf. Eine Ontologie mit z.B. zwei definierbaren Namen könnte demnach keine Instanz dieser Klasse sein.

## 5 Anwendungen

Die aufgeführten Anwendungsbeispiele bieten nur eine kleine Auswahl aus Dutzenden von Tools, die entweder etwas ähnliches leisten oder noch derart neu und unausgereift sind, dass eine akzeptable Präsentation zu diesem Zeitpunkt noch nicht möglich ist. Es gibt viele JAVA-Anwendungen, die oft auch nicht ohne weiteres lauffähig sind und weitere Software benötigen. Die Programme werden nach und nach auf der DAML-Homepage zur Verfügung gestellt.

### 5.1 Suchmaschine der DAML-Homepage

Auf der DAML-Homepage gibt es eine Suchmaschine, mit der man gezielt alle definierten Klassen und Properties suchen kann. Das Ergebnis ist eine Liste mit allen Links, auf denen die gesuchten Objekte benutzt werden. Es ist auch möglich, sich die Ontologien und die dazugehörigen Links nach Kriterien wie URI, Registrierungsdatum, Schlüsselwort, Klassen oder Properties ausgeben zu lassen. Da es sich hier um die offizielle DAML-Homepage handelt, kann man davon ausgehen, dass sich wohl alle Institutionen und Firmen, die sich ernsthaft mit DAML-Entwicklungen beschäftigen, hier registriert sind. Die Qualität der Links ist jedoch äußerst verschieden, da viele DAML-Anwendungen überdurchschnittlich viele Fehler und Warnungen enthalten. Es sind sogar Server registriert, die gar keine DAML-Anwendungen anbieten. Allerdings ist diese Suchmaschine auch aufgrund seiner anzunehmenden weitgehenden Vollständigkeit eine gute Informationsquelle. Da auch kleinste DAML-Konstrukte analysiert und eindeutig identifiziert werden können, ist diese Suchmaschine spürbar zuver-



lässiger als herkömmliche Suchmaschinen, die in HTML-Dokumenten nach Begriffen suchen, da sie auch semantisch die gewünschten Ergebnisse liefern kann.

## 5.2 Wall Street Journal (zur Zeit noch fiktiv)

Angenommen, das Wall Street Journal würde nicht mehr nur als Text veröffentlicht werden, sondern würde in ihrer Online-Ausgabe auch DAML verwenden. Dann könnten z.B. Ontologien wie Firmen und Artikel erstellt werden. Eine Firma würde unter anderem ihren Namen und einige Wirtschaftsdaten wie Börsenkurse als Attribute bekommen. Ein Artikel könnte mit den Attributen Titel, Datum und Herausgeber versehen sein. Ein Agent könnte nun einem Benutzer automatisch jedes Mal eine E-Mail schicken, wenn in DAML-kompatiblen Publikationen (oder eben gezielt im Wall Street Journal) ein Artikel über die gewünschte Firma erscheint. Da der Agent die Ontologie „Firma“ kennt und diese beim Wall Street Journal gefunden hat, könnte er die Firmendaten selbständig aus einem DAML-Dokument auslesen und sie dem Benutzer mitschicken. Somit kennt der Benutzer die aktuellen verfügbaren Eckdaten der Firma beim Erscheinen des Artikels. Eine höhere semantische Ebene wäre die Einteilung aller Firmen in Branchen, so dass ein Agent z.B. alle Artikel mit ihren aktuellen Firmeneckdaten zur Branche „Banken“ liefern kann.

## 5.3 DAML Crawler

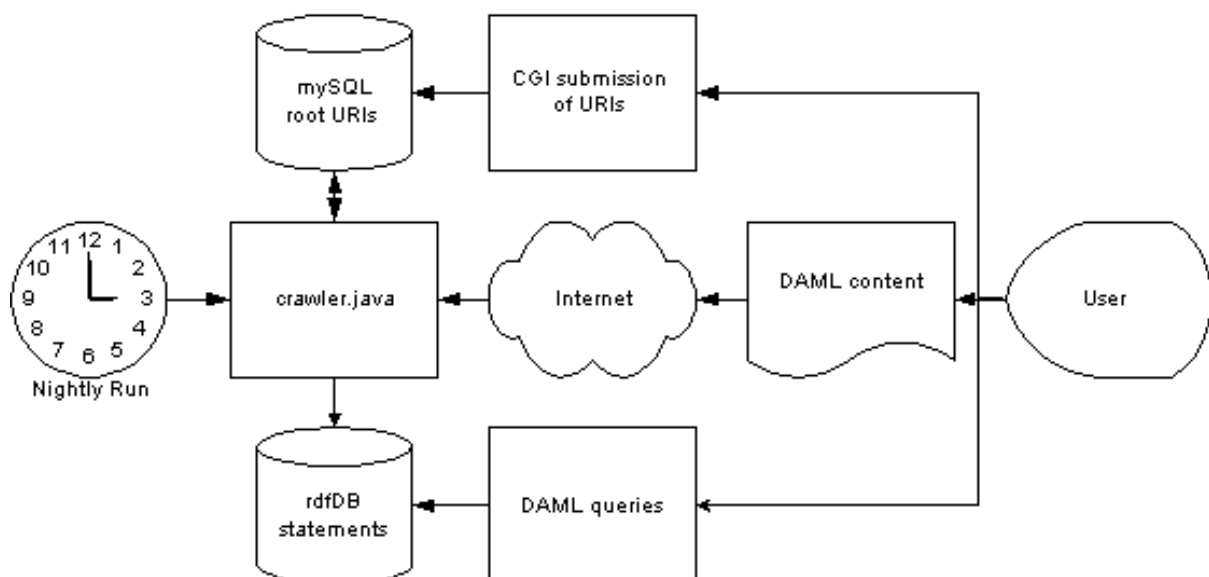


Abb. 1: Architektur des DAML Crawlers (DAML-Homepage)

Der DAML Crawler ist ein JAVA-Programm, das selbständig nach DAML-Dokumenten sucht und diese analysiert. Dazu müssen die zu analysierenden URIs in die Crawler-Datenbank eingetragen worden sein, um von Agenten periodisch auf DAML-Objekte untersucht werden zu können. Der Crawler selbst liest nur Resultate aus einer Datenbank aus (Beispiel vom 27.11.2001 mit eingetragenen 60 Servern – Tabelle gekürzt):

### Site Index

| Site  | DAML Pages            | DAML Statements | All Pages | Exceptions          | Warnings             |
|---|-----------------------|-----------------|-----------|---------------------|----------------------|
| <a href="http://daml.umbc.edu/">http://daml.umbc.edu/</a>                         | <a href="#">90</a>    | 50588           | 3161      | <a href="#">119</a> | <a href="#">2764</a> |
| <a href="http://grcinet.grci.com/">http://grcinet.grci.com/</a>                   | <a href="#">30</a>    | 42136           | 941       | <a href="#">9</a>   | <a href="#">66</a>   |
| <a href="http://ontolingua.stanford.edu/">http://ontolingua.stanford.edu/</a>     | <a href="#">2</a>     | 39396           | 8765      | <a href="#">7</a>   | <a href="#">2</a>    |
| <a href="http://opencyc.sourceforge.net/">http://opencyc.sourceforge.net/</a>     | <a href="#">4</a>     | 52004           | 10537     | <a href="#">70</a>  | <a href="#">185</a>  |
| <a href="http://projects.teknowledge.com/">http://projects.teknowledge.com/</a>   | <a href="#">15655</a> | 215175          | 20276     | <a href="#">54</a>  | <a href="#">135</a>  |
| <a href="http://www.aifb.uni-karlsruhe.de/">http://www.aifb.uni-karlsruhe.de/</a> | <a href="#">9</a>     | 566             | 9868      | <a href="#">135</a> | <a href="#">1066</a> |
| <a href="http://www.daml.org/">http://www.daml.org/</a>                           | <a href="#">93</a>    | 847573          | 4171      | <a href="#">14</a>  | <a href="#">621</a>  |
| <a href="http://www.daml.ri.cmu.edu/">http://www.daml.ri.cmu.edu/</a>             | <a href="#">64</a>    | 66803           | 107       | <a href="#">1</a>   | <a href="#">12</a>   |
| <a href="http://www.isi.edu/">http://www.isi.edu/</a>                             | <a href="#">52</a>    | 30889           | 11223     | <a href="#">262</a> | <a href="#">1678</a> |
| <a href="http://www.kestrel.edu/">http://www.kestrel.edu/</a>                     | <a href="#">31</a>    | 92047           | 2083      | <a href="#">52</a>  | <a href="#">178</a>  |
| <a href="http://www.semanticweb.org/">http://www.semanticweb.org/</a>             | <a href="#">8</a>     | 475287          | 14551     | <a href="#">30</a>  | <a href="#">31</a>   |
| <a href="http://www.w3.org/">http://www.w3.org/</a>                               | <a href="#">377</a>   | 14215           | 15995     | <a href="#">986</a> | <a href="#">1640</a> |
| TOTAL   | 17019                 | 2690283         | 275281    | 2779                | 23069                |

Tab. 1: Auszug aus einer Ergebnisausgabe des DAML Crawlers (DAML-Homepage)

Die letzte Zeile der Tabelle bezieht sich entsprechend auf alle 60 Server und nicht nur auf diese Auswahl.

## 5.4 PalmDAML

Bei diesem Programm handelt es sich um eine Anwendung für Palm Desktops. Dabei handelt es sich um kabellose Organizer, mit denen Terminkalender, E-Mails, Adressen usw. mit kabellosem Internet-Anschluß verwaltet werden können. Die Architektur dieser Anwendung hat folgende Gestalt:

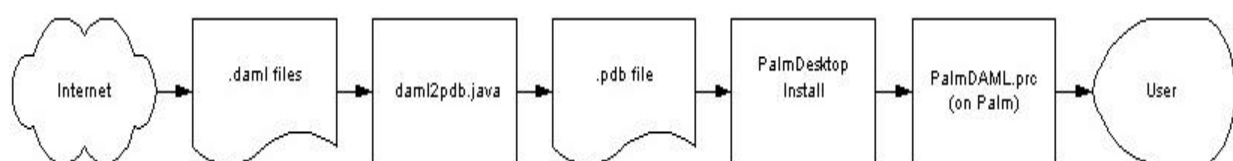


Abb. 2: Architektur des DAML Crawlers (DAML-Homepage)

Aus den im Internet befindlichen DAML-Dateien lesen Agenten alle relevanten Informationen für das Programm daml2pdb aus, das in JAVA entwickelt wurde und diese Informationen in eine Palm-Datenbank schreibt, die man mit dem Palm-Organizer aus dem Internet herunterladen kann. Sie wird dann mit dem PalmOS-Programm HotSync konvertiert, so dass sie von PalmDAML verwendet werden kann. Die Größe einer Datenbank ist zur Zeit jedoch noch auf 64 KB begrenzt. Die Verwendung einer solchen Datenbank sähe dann z.B. so aus:



Abb. 3: Anwendungen auf Palm Desktop (DAML-Homepage)

Wählt man die Anwendung PalmDAML aus, so erhält man die Anzahl der Einträge der aktuellen Datenbank. In dieser Beispiel-Datenbank finden sich Daten über den europäischen Adel, die scheinbar aus mehreren DAML-Dateien ausgelesen werden.



Abb. 4: Größe der Palm-Datenbank (DAML-Homepage)

Danach folgt eine Liste mit allen Knoten, die aus Platzgründen auf abgekürzte Namen (hier: gedcom als Verzeichnisname mit den DAML-Dateien) und die Resource-Namen beschränkt ist.



Abb. 5: Listenauszug der Knoten (DAML-Homepage)

Ein Klick auf einen Knoten liefert alle Statements, die in diesem Knoten vorhanden sind.



Abb. 6: Inhalte eines Knotens (DAML-Homepage)

Diese Ansicht besteht aus drei Flächen. In der obersten Fläche steht die komplette URI für den Knoten. Der mittlere Bereich zeigt alle gefundenen Statements an. Die Pfeile geben an, wer in der Beziehung zwischen URI und Statement-Wert das Subjekt und wer das Objekt ist. So lässt sich ableiten, dass die „F16“-URI die Objekte event17 (Scheidung), event18 (Heirat) und family als Typendefinition hat. Andererseits ist „F16“ ein Objekt der Subjekte 1115 und 1116. Zu jeder dieser vermeintlichen Personen kann es dann URIs geben. In der unteren Fläche werden alle Informationen zu einem Knoten, den man in der mittleren Fläche auswählen kann, angezeigt. Darüber hinaus gibt es auch eine Suchfunktion für die Datenbank:



Abb. 7: Suchfunktion (DAML-Homepage)

Als Ergebnis würden dann alle Knoten angezeigt, in denen der Suchbegriff gefunden wurde:



Abb. 8: Suchergebnis (DAML-Homepage)

Die gefundenen Knoten lassen sich auswählen, um sich den Knoten detailliert anschauen zu können. So lässt sich Diana Spencer als Subjekt I165 identifizieren.

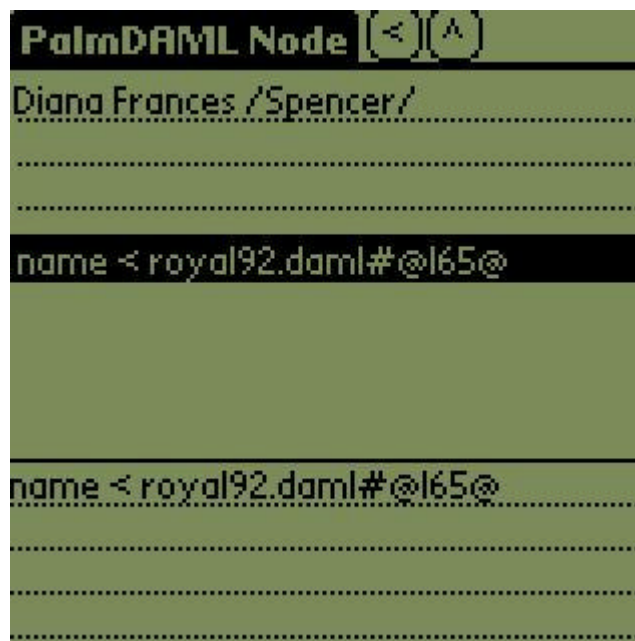


Abb. 9: Knotenansicht des Suchergebnisses (DAML-Homepage)

Bei PalmDAML handelt es sich um eine Anwendung, die durchaus das Potential hat, schon bald serienmäßig auf allen Palm Desktops kommerziell vertrieben werden zu können. Einschränkungen wie die geringe Datenbankgröße sind jedoch noch ein Problem. Außerdem wird



daran gearbeitet, DAML-Statements filtern zu können und auch selber eigene DAML-Statements zu erzeugen.

## 5.5 DAML Viewer

Dieses JAVA-Programm ist ein weiteres Tool zur Abfrage von DAML-Statements. Es ist als Applet und als Applikation erhältlich. Auch hier muss wieder die zu durchsuchenden URI angegeben werden. Die Angabe einzelner Dateien ist auch möglich. Eine Eingabe sähe so aus:

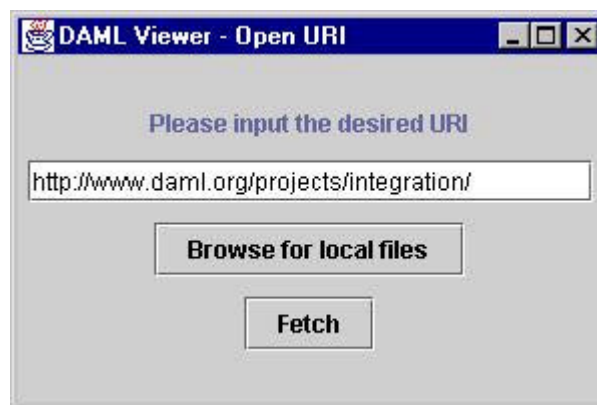


Abb. 10: Eingabefenster des DAML Viewers (DAML-Homepage)

Ein Klick auf den „Fetch“-Button führt zu einer Liste mit allen URIs, die zu den gefundenen Ressourcen gehören. Ein Auszug zur Beispiel-URI hätte folgende Form:

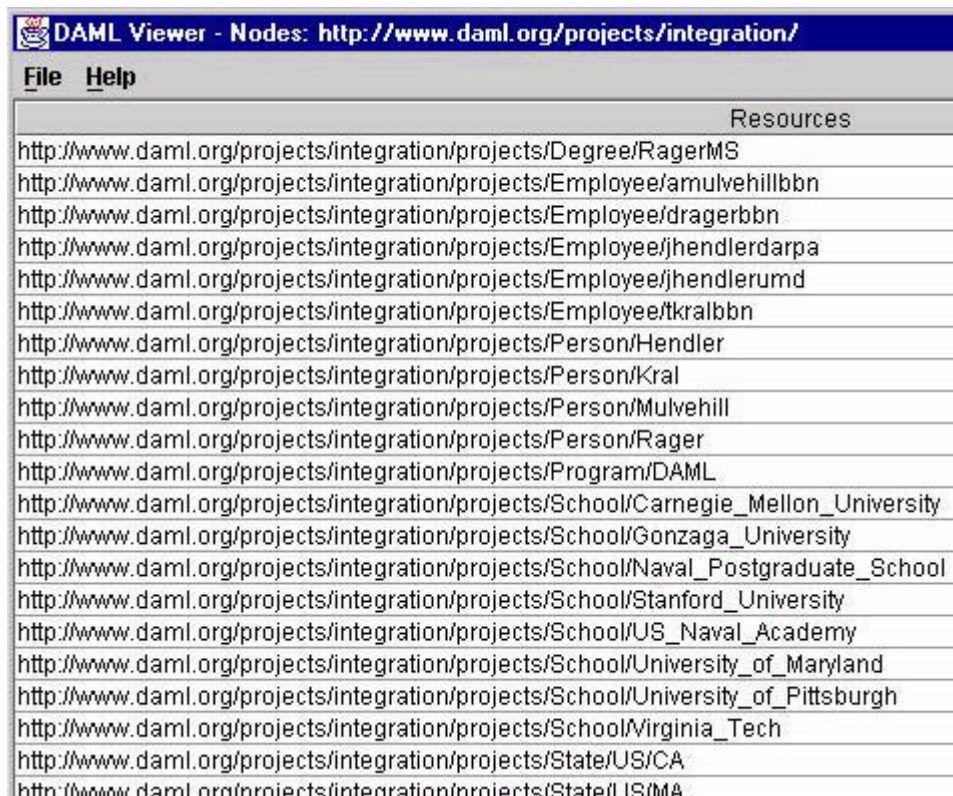
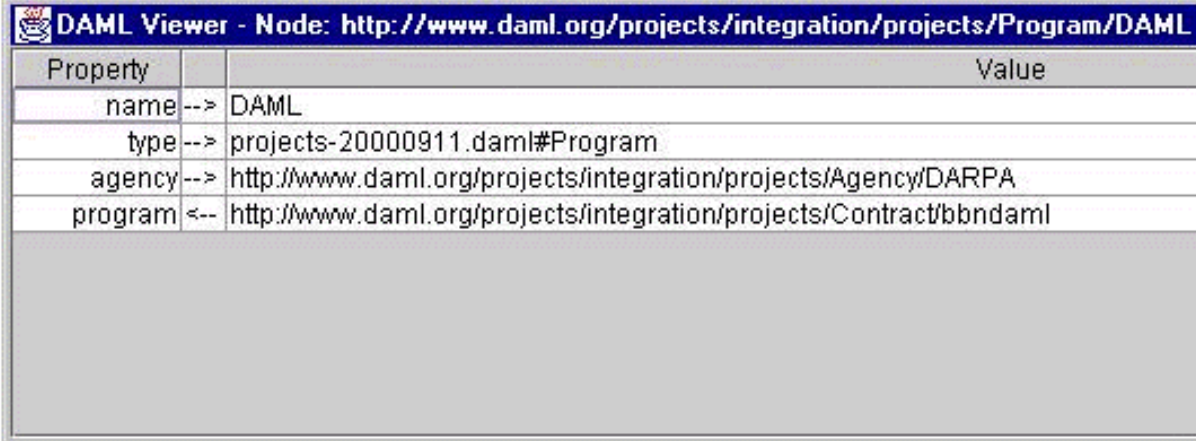


Abb. 11: Ausgabe der gefundenen Ressourcen (DAML-Homepage)

Das „File“-Menü bietet einige Optionen, von denen hier nur die wichtigen aufgeführt werden. Mit „Properties“ erhält man zur eingegebenen URI die Anzahl der Ressourcen und der gefundenen DAML-Statements. Die Option „View Source“ liefert den Quellcode zu einem auswählbaren Knoten. Durch das Anklicken eines Knotens wird darüber hinaus noch ein weiteres Fenster geöffnet, das alle verwendeten Properties auflistet:



The screenshot shows a window titled "DAML Viewer - Node: http://www.daml.org/projects/integration/projects/Program/DAML". Below the title bar is a table with two columns: "Property" and "Value".

| Property    | Value  |
|-------------|--|
| name -->    | DAML   |
| type -->    | projects-20000911.daml#Program                                     |
| agency -->  | http://www.daml.org/projects/integration/projects/Agency/DARPA     |
| program <-- | http://www.daml.org/projects/integration/projects/Contract/bbndaml |

Abb. 12: Properties in einem Knoten (DAML-Homepage)

Die Pfeile definieren wie schon beim PalmDAML die Subjekt-Objekt-Beziehung zwischen der Resource-URI und den Property-Werten. So sind in diesem Beispiel der Name „DAML“, der Typ „projects-20000911.daml#Program“ und die Organisation „.../Agency/DARPA“ Objekte des Subjektes „.../projects/program/DAML“ aus der Titelzeile. Andererseits ist dieses jedoch das Objekt des übergeordneten Programms „.../Contract/bbndaml“, das demnach das Subjekt ist.

## 5.6 HyperDAML

HyperDAML ist eine kleine JAVA-Anwendung, die ein DAML-Dokument in ein HTML-Dokument konvertiert. Eine derartige Anwendung ist sehr nützlich, wenn z.B. ein Agent ein in DAML-Notation vorliegendes Dokument präsentieren soll. Der Anwender hat nun direkt die Möglichkeit, die Links auf verwendete Klassen, Properties etc. direkt anzuwählen. Die mehrfarbige Darstellung erhöht außerdem die Lesbarkeit. Betrachtet man z. B. folgenden Auszug aus der DAML-Datei gedcom (s. PalmDAML):



```
<rdfs:subClassOf>
  <daml:Restriction>
    <daml:onProperty rdf:resource="#surname"/>
    <daml:toClass
      rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:Restriction>
</rdfs:subClassOf>
```

Dann erhält man nach der Konvertierung eine HTML-Ausgabe der Form:

```
<rdfs:subClassOf>
  <daml:Restriction>
    <daml:onProperty rdf:resource="#surname"/>
    <daml:toClass rdf:resource="http://www.daml.org/cgi-b
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:Restriction>
</rdfs:subClassOf>
```

## 6 Zusammenfassung

Der DAML-Standard vereinigt in seiner Anwendbarkeit die Standards RDF, XML, OIL und DAML selbst, so dass DAML die besten Chancen hat, die anderen Standards zu verdrängen. Mit der DARPA hat DAML sicherlich auch einen hochwertigen Schirmherrn, der ein überdurchschnittliches Interesse an diesem Projekt hat. Es ist bereits in der jetzigen frühen Entwicklungsstufe beeindruckend, wie detailliert DAML-Informationen und DAML-Statistiken selbständig von Agenten aus dem Internet zusammengetragen werden. Aus dem bisherigen HTML-Datenfluss soll nun ein intelligent strukturierter und semantisch eindeutiger Informationsaustausch werden. Es ist jedoch fraglich, ob bis zum geschätzten Ende des DAML-Projektes um 2004 herum die Agenten nicht mehr auf die vorhergehende Eingabe der URIs angewiesen sein werden oder ob wenigstens eine Informationsrecherche mit Aktualisierung in Echtzeit möglich sein wird. Das Semantic Web wird jedoch greifbarer, und die Möglichkeiten, die sich durch ein intelligentes semantisches Internet eröffnen, können zum heutigen Zeitpunkt wohl nur erahnt werden. Der militärische Nutzen des agenten-basierten DAMLs wird sicherlich enorm sein und könnte z.B. im umstrittenen neuen US-Raketenabwehrsystem Anwendung finden.

## 7 Literatur

Als Literatur wurden ausschließlich folgende Internet-Links verwendet:

1. [www.daml.org](http://www.daml.org) : Offizielle DAML-Homepage (Betreiber DARPA), Stand 27.11.2001
2. [www.darpa.mil](http://www.darpa.mil) : Defense Advanced Research Projects Agency, Stand 1.1.2002