UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

P3P-Policy-Generator für Rheinland-Pfalz

Diplomarbeit

zur Erlangung des Grades einer Diplom-Informatikerin
im Studiengang Computervisualistik

vorgelegt von

Carmen Kölbl

200210265

Erstgutachter:	Prof. Dr. Rüdiger Grimm, Institut für Wirtschafts- und Verwaltungsinformatik, Arbeitsgruppe Grimm
Zweitgutachter:	Dipl.-Ing. Helge Hundacker, Institut für Wirtschafts- und Verwaltungsinformatik, Arbeitsgruppe Grimm
Weitere Betreuer:	Ministerialrat Helmut Eiermann, Der Landesbeauftragte für den Datenschutz Rheinland-Pfalz, Leiter der Gruppe IV: Technisch-organisatorischer Datenschutz

Koblenz, im März 2008

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und die direkt oder indirekt aus fremden Quellen übernommenen Gedanken als solche kenntlich gemacht habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

.....

(Ort, Datum) (Unterschrift)

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die diese Arbeit erst ermöglicht haben.

Besonderer Dank gebührt Prof. Dr. Rüdiger Grimm für die Ausschreibung dieser Diplomarbeit, die meinen Interessen sehr entgegenkam und es mir ermöglichte, mich mit dem Thema Datenschutz zu beschäftigen und selbst ein Programm zu schreiben. Danken möchte ich ihm auch für den großen Freiraum, den ich bei der Anfertigung der Arbeit hatte.

Ganz herzlich möchte ich mich bei Ministerialrat Helmut Eiermann bedanken, der die Arbeit seitens des Landesbeauftragten für den Datenschutz Rheinland-Pfalz betreute. Er stand stets für Fragen zur Verfügung und seine konstruktiven Anregungen halfen mir – besonders auf juristischem Gebiet – den Überblick zu behalten und die richtigen Prioritäten zu finden.

Dipl.-Ing. Helge Hundacker danke ich für die engagierte Betreuung meiner Arbeit, für viele hilfreiche Hinweise und Diskussionen und für die Möglichkeit, das Thema eigenständig und mit viel Entscheidungsfreiheit zu bearbeiten.

Bei meinen Eltern möchte ich mich besonders bedanken, da sie mich in jeglicher Hinsicht unterstützt und mich zu der Entscheidung für einen technischen Studiengang ermutigt haben. Weiterhin möchte ich meinen Freunden und Verwandten für ihre moralische Unterstützung danken.

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
APPEL	A P3P Preference Exchange Language
DTD	Document Type Definition
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JFC	Java Foundation Classes
P3P	Platform for Privacy Preferences
PET	Privacy Enhancing Technology
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Abbildungsverzeichnis

Listings

1 Einleitung

In den folgenden Abschnitten werden Inhalte und Zielsetzung der Diplomarbeit zusammengefaßt (vgl. 1.1 und 1.2) und die Organisation der Arbeit vorgestellt (vgl. 1.3).

1.1 Gegenstand der Diplomarbeit

Gegenstand dieser Arbeit ist die Entwicklung eines Generators für Datenschutzpolicies. Diese werden in der P3P-Sprache formuliert. P3P steht für *Platform for Privacy Preferences* und ist ein Standard zum Schutz der Privatsphäre im Internet, der vom *World Wide Web Consortium* (W3C) entwickelt wurde. Er dient zum Austausch von Datenschutzinformationen. Die Datenschutzpraktiken einer Website können durch P3P in maschinenlesbarer Form durch sogenannte Policies beschrieben werden. So wird ein automatisierter Vergleich mit den entsprechenden Präferenzen auf Nutzerseite möglich. Der im Rahmen dieser Arbeit entwickelte Generator soll das Erstellen der Policies auf Anbieterseite erleichtern. Er entstand in Zusammenarbeit mit dem Landesbeauftragten für den Datenschutz Rheinland-Pfalz und geht auf die Bedürfnisse von Behörden ein, die P3P-Policies auf ihren Webseiten verwenden wollen.

1.2 Zielsetzung der Diplomarbeit

Ziel der Diplomarbeit ist es, einen P3P-Policy-Generator zu entwickeln, der es öffentlichen Dienstleistern erleichtert, auf ihren Webseiten P3P zu verwenden. Als Grundlage sollen hierfür die Datenschutzanforderungen von Behörden in Rheinland-Pfalz durch geeignete Policy-Blöcke formuliert werden. Zudem soll eine einfache Methode zum Zusammenstellen dieser Blöcke entworfen werden. Es soll weiterhin untersucht werden, welche Tools auf Nutzerseite es ermöglichen, Datenschutzpräferenzen im Bezug auf P3P zu definieren, und inwiefern ein Abgleich von Präferenzen und Anbieterpraktiken in diesen Tools möglich ist.

1.3 Organisation der Arbeit

In Kapitel 2 werden Grundlagen erläutert, die zum Verständnis der Arbeit nötig sind. Hier werden zunächst Grundlagen im Bezug auf XML dargestellt (vgl. Abschnitt 2.1). Anschließend werden die grundlegenden Konzepte sowie die Funktionsweise von P3P vorgestellt (vgl. Abschnitt 2.2 und 2.3). Einen

Überblick über das Vokabular zur Formulierung einer Policy gibt Abschnitt 2.4. Die Kurzform einer P3P-Policy, die sog. Compact Policy, stellt Abschnitt 2.5 vor. Die Motivation für den Einsatz von P3P beleuchtet Abschnitt 2.6 näher, dabei wird ausgeführt, welche Kritik an P3P geübt wird, was P3P momentan leistet und was nicht.

Die in Zusammenarbeit mit dem Landesbeauftragten für den Datenschutz Rheinland-Pfalz ermittelten Anforderungen an den Generator werden in Kapitel 3.1 beschrieben. Der Entwurf des Systems wird in Kapitel 3.2 vorgestellt. Nach einer Beschreibung der einzelnen Teile des Systems werden diese Bereiche und die ihnen zugeteilten Aufgaben beschrieben. Die Basis des Systems ist das Datenmodell, dieses bildet die für die Template-Policies benötigten Konzepte des P3P-Standards ab (vgl. Abschnitt 3.2.1). Der Teil der Systemarchitektur, der von der Benutzerschnittstelle verwendet wird, um den aktuellen Zustand des Datenmodells zu erfragen und ggf. Änderungen daran vorzunehmen, wird in Abschnitt 3.2.2 vorgestellt. Abschnitt 3.2.3 beschreibt den Entwurf der Benutzerschnittstelle. Abschnitt 3.2.4 erläutert im Hinblick auf zentrale Abläufe, wie die Komponenten des Systems miteinander interagieren.

Die Implementation wird in Kapitel 4 vorgestellt, dabei wird zunächst ausgeführt, welche Komponenten in dem Generator zum Einsatz kommen (Programmiersprache, GUI-Bibliothek, XML-Parser, etc.), und warum diese ausgewählt wurden (vgl. Abschnitt 4.1). Welche Funktionalität des Generators auf welche P3P-Version zurückgreift, erläutert Abschnitt 4.2. Inwieweit Anforderungen im Bereich der Softwareergonomie und der Barrierefreiheit umgesetzt wurden, beschreiben die Abschnitte 4.3 und 4.4. Bedeutsam für die Robustheit des Systems sind Mechanismen der Fehlerbehandlung und der Validierung, ihre Umsetzung wird in den Abschnitten 4.5 und 4.6 dargestellt. Abschnitt 4.7 erläutert, wie weitere Templates in den Generator integriert werden können.

Kapitel 5 stellt die für den Generator entwickelten Muster-Policies vor, die auf die Anforderungen des Landesbeauftragten für den Datenschutz Rheinland-Pfalz zugeschnitten sind. Abschnitt 5.1 gibt eine Einführung zum Datenschutzrecht in Deutschland, diskutiert P3P aus datenschutzrechtlicher Sicht und untersucht, inwiefern P3P die Anforderungen des Telemediengesetzes erfüllt. Die einzelnen Musterblöcke, die sog. Templates, werden in Abschnitt 5.2 vorgestellt.

Eine Bedienungsanleitung für den Generator ist in Kapitel 6 enthalten. Hier wird beschrieben, wie die einzelnen Funktionalitäten über die Benutzerschnittstelle erreicht werden können.

Kapitel 7 geht auf die Tools ein, die aktuell P3P auf der Seite des Nutzers unterstützen. Dabei wird der Umfang erläutert, in dem die Möglichkeiten des P3P-Standards in den einzelnen Tools genutzt werden und es werden Empfehlungen zur Konfiguration der Tools gegeben.

Kapitel 8 reflektiert die gewonnenen Erkenntnisse sowie das Maß, in dem der Generator die erhobenen Anforderungen erfüllt. In einem Ausblick werden für die Zukunft sinnvolle Erweiterungsmöglichkeiten des Generators diskutiert, die aufgrund der zeitlichen Rahmenbedingungen dieser Arbeit nicht realisiert wurden.

2 Grundlagen

Nach einer Vorstellung der Grundlagen für P3P (vgl. Abschnitt 2.1) sowie einer Einführung in den P3P-Standard (vgl. Abschnitt 2.2) geht es in Abschnitt 2.3 um dessen Funktionsweise. In Abschnitt 2.4 geht es um das Vokabular, das P3P zur Formulierung von Policies und Policy-Referenzen bietet. Abschnitt 2.5 beschreibt die Kurzfassung einer Policy, die sog. Compact Policy. Die Motivation für die Verwendung von P3P sowie die Kritik an der aktuellen Version werden in Abschnitt 2.6 vorgestellt.

2.1 XML-Grundlagen

Da P3P eine XML-Applikation ist, werden im folgenden Abschnitt Grundlagen zu XML vorgestellt, die für das Verständnis von P3P und von Komponenten, die vom P3P-Policy-Generator eingesetzt werden, nötig sind. Abschnitt 2.1.1 gibt eine Einführung in XML, Abschnitt 2.1.2 stellt die Grundlagen zu XML Schema – der Schemasprache, die die Grammatik von P3P-Dokumenten formuliert – vor. Um auf einzelne Bestandteile eines XML-Dokuments zuzugreifen, verwendet der P3P-Generator die Anfragesprache XPath, diese wird in Abschnitt 2.1.3 vorgestellt. Die Darstellung dieser Sprachen beschränkt sich auf grundlegende Aspekte und gibt lediglich eine Einführung in die einzelnen Gebiete.

2.1.1 XML

Die *Extensible Markup Language*, kurz XML, ist eine Metasprache, d.h. eine Sprache, mit der andere Sprachen definiert werden können. XML dient zur Definition von Auszeichnungssprachen. Durch diese Sprachen können strukturierte Daten wie z.B. Textdokumente, Vektorgrafiken oder Daten aus Datenbanken beschrieben werden (vgl. [Kersken04]).

Die sogenannten Auszeichnungen beschreiben die Struktur eines Dokuments. Der XML-Standard definiert die Syntax, der eine Auszeichnung entsprechen muss. Die gebräuchlichste Form der Auszeichnung ist das Tag, diese Tags sind nicht von XML vorgegeben, sondern können bei Bedarf vom Entwickler selbst definiert werden. So kann die Sprache erweitert und an unterschiedliche Bedürfnisse angepaßt werden. Auszeichnungen können deshalb auch die Semantik eines Dokuments beschreiben, wenn die Benennung der Auszeichnungen passend gewählt wird.

Da XML-Dokumente einfache Textdateien sind, die sowohl die Daten selbst als auch die Auszeichnungen in Form von Text enthalten, bietet XML die Möglichkeit, Daten plattformübergreifend vorzuhalten und auszutauschen (vgl. [Harold03]).

Das XML-Dokument

Im Folgenden wird der Aufbau eines XML-Dokuments durch ein Beispiel veranschaulicht (siehe Listing 2.1):

Listing 2.1: XML-Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<buch isbn="978-3453146976">
    <titel>Per Anhalter durch die Galaxis</titel>
    <autor>
        <vorname>Douglas</vorname>
        <nachname>Adams</nachname>
    </autor>
    <verlag>Heyne</verlag>
    <seiten>204</seiten>
</buch>
```

Die XML-Deklaration Die erste Zeile des Beispieldokuments in Listing 2.1 enthält die XML-Deklaration. Diese gibt die verwendete XML-Version sowie die Kodierung des Dokuments an. Die Verwendung der XML-Deklaration wird von der XML-Spezifikation empfohlen (vgl. [Bray06b]). Falls sie verwendet wird, muss sie an erster Stelle im Dokument stehen.

Elemente Ein Element ist der wichtigste Bestandteil eines XML-Dokuments. Es besteht aus einem Start-Tag und einem End-Tag und enthält Inhalte wie Text oder weitere Elemente. Das Dokument in Listing 2.1 enthält ein Element *buch*, das Start-Tag ist hier <buch>, das entsprechende End-Tag wird durch einen Schrägstrich nach der öffnenden Klammer markiert, im Beispiel also </buch>. Das Beispielelement enthält vier Kindelemente: *titel*, *autor*, *verlag* und *seiten*. Das Element *autor* enthält wiederum zwei Kindelemente *vorname* und *nachname*.

Es gibt im vorliegenden Beispiel Elemente, die nur Kindelemente enthalten, wie z.B. *autor*. Zudem gibt es Elemente, die nur Text enthalten, ein Beispiel hierfür ist *titel*. Es ist auch möglich, leere Elemente zu definieren. Für ein leeres Element gibt es neben der bisher vorgestellten Schreibweise <elementName> </elementName> noch eine Kurzform, in der der „schließende“ Schrägstrich in das Start-Tag gezogen wird: <elementName/>.

In jedem XML-Dokument gibt es ein Wurzelelement. Dieses Element ist das erste Element im Dokument und enthält alle anderen Elemente (vgl. [Harold03]). Im obigen Beispiel ist das Element *buch* das Wurzelelement.

Attribute Für ein XML-Element können beliebig viele Attribute definiert werden. Attribute werden im Start-Tag eines Elements plaziert. Ein Attribut besteht aus einem Namen und einem Wert, der diesem Namen zugeordnet wird (vgl. [Harold03]). Im Beispiel in Listing 2.1 hat das Element *buch* ein Attribut *isbn*.

Kommentare XML-Dokumente können Kommentare enthalten. Ein Kommentar darf an jeder Stelle eines XML-Dokuments plaziert werden, außer innerhalb eines Tags oder innerhalb eines anderen Kommentars. Ein Kommentar hat folgende Syntax (vgl. [Harold03]):

Listing 2.2: Beispiel für einen Kommentar

```
<!-- Deklaration des Wurzelelements -->
```

Verarbeitungsanweisungen Verarbeitungsanweisungen bieten in XML die Möglichkeit, Informationen an bestimmte Anwendungen, die das Dokument lesen, zu übergeben. Eine Verarbeitungsanweisung beginnt mit <?, danach folgt das sogenannte *Ziel*, ein String, der z.B. einen Identifikator für diese Verarbeitungsanweisung enthält oder den Namen der Anwendung, für die diese Anweisung gedacht ist. Auf das Ziel folgt ein weiterer String, der Anweisungen oder Informationen für die verarbeitende Anwendung enthält. Das Ende der Verarbeitungsanweisung wird durch ?> markiert (vgl. [Harold03]).

Eine Verarbeitungsanweisung kann beliebige Informationen wie beispielsweise Pseudoattribute (name=„wert“) oder auch Funktionsaufrufe enthalten (vgl. [Niedermeier06]). Ein Beispiel für eine Verarbeitungsanweisung, die einen Funktionsaufruf definiert, zeigt Listing 2.3.

Listing 2.3: Beispiel für eine Verarbeitungsanweisung

```
<?Zielprogramm zeigeUebersicht("Nutzereingaben");?>
```

Es gibt noch weitere Bestandteile, die ein XML-Dokument enthalten kann, die aber für das Verständnis dieser Arbeit nicht benötigt werden. Eine ausführliche Darstellung aller Komponenten findet sich in [Bray06b].

XML-Konzepte

XML-Applikationen Als XML-Applikation wird nach [Harold03] die Anwendung von XML auf ein bestimmtes Problemfeld bezeichnet. Eine XML-Applikation legt eine Menge von Tags fest, die zur Beschreibung der Sachverhalte auf diesem Gebiet verwendet werden dürfen. Ziel solcher Beschränkungen ist es, die Austauschbarkeit und Verarbeitungsfähigkeit von Dokumenten zu verbessern. Ein Beispiel für eine XML-Applikation ist P3P.

Welche Auszeichnungen in einer konkreten XML-Applikation erlaubt sind, und wie diese Auszeichnungen zu verwenden sind, kann mit Hilfe eines Schemas definiert werden. Ein Schema drückt demnach die Grammatik der zugehörigen Instanz-Dokumente aus. Zu diesem Zweck gibt es viele Schemasprachen, die beiden bekanntesten sind die DTD (*Document Type Definition*) und XML Schema. Da XML Schema auch bei P3P Verwendung findet, wird diese Sprache in Abschnitt 2.1.2 vorgestellt.

Namensräume Nach [Harold03] erfüllen Namensräume in XML zwei Aufgaben:

1. Elemente und Attribute zu unterscheiden, die aus verschiedenen Vokabularen stammen und den gleichen Namen haben. Als *Vokabular* werden Elemente und Attribute einer XML-Applikation bezeichnet, die mit dem Ziel entwickelt wurden, von mehreren Software-Modulen verwendet zu werden. Die Definition solcher Vokabulare ist motiviert durch den Wunsch nach Modularität: Ein Vokabular, das weithin verstanden wird und für das verarbeitende Software vorhanden ist, kann wiederverwendet werden und muss nicht neu erfunden werden ([Bray06a]).

2. Alle Elemente und Attribute einer XML-Applikation zu kennzeichnen, um sie für Software leicht erkennbar zu machen.

Ein Namensraum wird durch einen URI (*Uniform Resource Identifier*) identifiziert. URIs sind Zeichenfolgen, die eine abstrakte oder physische Ressource identifizieren. URIs können z.B. die Adressen von Webseiten oder sonstigen Ressourcen in einem Netzwerk sein, der Begriff des URIs ist aber sehr weit gefaßt und nicht auf elektronische Dokumente beschränkt (vgl. [Berners-Lee05]).

Der URI eines Namensraums wird jedem Element und Attribut zugeordnet. Dies geschieht, indem jedem Element und Attribut ein Präfix vorangestellt wird. Dieses Präfix wird durch ein xmlns:präfix-Attribut auf einen URI abgebildet. Es ist auch möglich, für Elemente ohne Präfix einen Default-URI anzugeben.

Alle Elemente und Attribute, die einem bestimmten URI zugeordnet sind, gehören zum gleichen Namensraum. In den meisten Fällen werden alle Elemente einer XML-Applikation einem URI zugeordnet, einige Applikationen verwenden auch mehrere Namensräume, um verschiedene Teile der Applikation zu kennzeichnen (vgl. [Harold03]).

Wohlgeformtheit Alle XML-Dokumente müssen wohlgeformt sein. Wann die Eigenschaft der Wohlgeformtheit erfüllt ist, wird durch eine Reihe von syntaktischen Regeln definiert. Die wichtigsten dieser Regeln sind im folgenden aufgeführt (vgl. [Bray06b]):

- Zu jedem Start-Tag muss ein dazugehörendes End-Tag vorhanden sein.
- Überlappungen von Elementen sind nicht erlaubt, ein Element muss auf gleicher Ebene geöffnet und geschlossen werden.
- Es gibt genau ein Wurzelelement.
- Innerhalb eines Start-Tags müssen Attribute eindeutig spezifiziert sein, es darf kein Attributname mehrmals vorkommen.
- Attributwerte müssen in Anführungszeichen gesetzt sein.

Zum Einlesen und Verarbeiten von XML-Dokumenten werden sogenannte XML-Prozessoren verwendet. Verstöße gegen die Regeln der Wohlgeformtheit haben zur Folge, dass XML-Prozessoren das entsprechende XML-Dokument nicht verarbeiten können, sie geben eine Fehlermeldung aus.

Gültigkeit Unter der Voraussetzung, dass die Prüfung auf Wohlgeformtheit bestanden ist, kann der Inhalt eines XML-Dokuments auch daraufhin überprüft werden, ob er einem bestimmten Schema entspricht. Diese Überprüfung wird als Validierung bezeichnet. Hält das XML-Dokument die Regeln des Schemas ein, wird es als gültig bezeichnet.

Ein validierender XML-Parser überprüft ein XML-Dokument also zunächst auf Wohlgeformtheit, falls diese vorliegt, wird das Dokument mit dem zugehörigen Schema verglichen. Ist das Dokument ungültig, listet der Parser alle Stellen auf, die nicht mit den Vorgaben des Schemas übereinstimmen (vgl. [Harold03]).

2.1.2 XML Schema

XML Schema ist eine vom W3C entwickelte Schemasprache, die dazu dient, ein gültiges Dokument formal zu beschreiben. Ein Schema-Dokument ist selbst ein XML-Dokument, die XML-Elemente, die das Schema bilden, liegen im XML-Schema-Namensraum *http://www.w3.org/2001/XMLSchema*.

Mit Hilfe von XML Schema können komplexe Beschränkungen für die Verwendung von Elementen und Attributen formuliert werden. Das Wurzelelement eines Schema-Dokuments ist *schema*, dieses Element enthält alle Element- und Attributdeklarationen (vgl. [Harold03]).

Das Schema-Dokument

Der Aufbau und die Bestandteile eines Schema-Dokuments sollen anhand eines Beispiels erklärt werden. Listing 2.4 zeigt ein XML Schema-Dokument, das eine Grammatik für das XML-Beispiel im vorigen Abschnitt angibt (siehe Listing 2.1).

Listing 2.4: XML Schema-Beispiel

```
1  <?xml version="1.0"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4  <xs:element name="buch" type="buchtyp"/>
5
6  <xs:complexType name="buchtyp">
7    <xs:sequence>
8      <xs:element name="titel" type="xs:string"/>
9      <xs:element name="autor" type="autortyp"/>
10     <xs:element name="verlag" type="xs:string"/>
11     <xs:element name="seiten" type="xs:positiveInteger"
12       minOccurs="0"/>
```

```
13    </xs:sequence>
14    <xs:attribute name="isbn" type="isbnTyp" use="optional"/>
15  </xs:complexType>
16
17  <xs:complexType name="autortyp">
18    <xs:sequence>
19      <xs:element name="vorname" type="xs:string"/>
20      <xs:element name="nachname" type="xs:string"/>
21    </xs:sequence>
22  </xs:complexType>
23
24  <xs:simpleType name="isbnTyp">
25      <xs:restriction base="xs:string">
26      <xs:pattern value="\d{3}\-\d{9}[0-9X]"/>
27      </xs:restriction>
28  </xs:simpleType>
29  </xs:schema>
```

Im obigen Beispiel fällt zunächst auf, dass das *schema*-Element und seine Kindelemente im XML-Schema-Namensraum liegen, was durch die Zuordnung dieses Namensraumes zu dem Präfix xs: verdeutlicht wird. Aus der XML-Deklaration in der ersten Zeile des Beispiels geht hervor, dass ein Schema-Dokument ein XML-Dokument ist.

Im vorliegenden Beispiel wird in Zeile 4 mit Hilfe des Elements xs:element ein Element deklariert. Das Attribut *name* gibt den Namen des Elements an, das Attribut *type* gibt den Typ des Inhalts an. Hier unterscheidet man einfachen und komplexen Inhalt. Das *buch*-Element in Zeile 4 hat einen komplexen Inhaltstyp, es kann daher eingebettete Elemente enthalten und Attribute haben. Im Gegensatz dazu gibt es in der XML Schema-Sprache grundlegende, einfache Datentypen, die den aus modernen Programmiersprachen bekannten Datentypen entsprechen. Diese einfachen Inhaltstypen dürfen keine eingebetteten Elemente enthalten.

Die Definition des komplexen Typs *buchtyp*, der in der Deklaration des *buch*-Elements explizit referenziert wird, erfolgt in Zeile 6. Der Inhalt des Typs *buchtyp* ist eine Sequenz aus vier Elementen: *titel*, *autor*, *verlag* und *seiten*. Eine Sequenz wird durch das xs:sequence-Element definiert, dieses Element legt die genaue Reihenfolge fest, in der die enthaltenen Elemente im Ziel-Dokument erscheinen müssen.

Das *autor*-Element hat einen komplexen Inhalt, die anderen Elemente der Sequenz haben einfache Inhaltstypen wie z.B. xs:string oder xs:positiveInteger. Das Element *seiten* enthält das Attribut *minOccurs*, das besagt, wie oft das Element mindestens auftreten darf. Analog dazu kann durch das Attribut *maxOccurs* eine Obergrenze für das Auftreten eines Elements gesetzt werden. Sowohl für *minOccurs* als auch für *maxOccurs* ist der Standardwert 1. Daher besagt die Deklaration des Elements *seiten* in Zeile 11, dass dieses Element 0 bis 1 mal auftreten darf.

Zudem kann das *buch*-Element ein Attribut *isbn* haben. Dass dieses Attribut optional ist, wird durch das Attribut *use* des xs:attribute-Elements ausgedrückt. Der Inhaltstyp dieses Attributs wird in Zeile 24 definiert, er leitet sich vom vordefinierten Typ xs:string ab. Durch das xs:simpleType-Element wird ein neuer einfacher Typ definiert und benannt. Ein einfacher Typ darf im Gegensatz zu komplexen Typen keine Elemente enthalten oder Attribute tragen. Das xs:restriction-Element dient dazu, Einschränkungen des Wertebereichs vorzunehmen.

Da für eine ISBN-Nummer nicht jeder beliebige String in Frage kommt, wird hier durch Angabe eines regulären Ausdrucks im Element xs:pattern der zulässige Wertebereich eingeschränkt. Der in Zeile 26 angegebene reguläre Ausdruck erlaubt folgende Zeichenfolge: Der String beginnt mit drei Ziffern, es folgt ein Bindestrich, darauf folgen neun weitere Ziffern, am Ende steht eine Prüfziffer, die im Bereich zwischen 0 und 10 liegen kann. Die 10 wird als X dargestellt (vgl. [Fallside04]).

Die Verwendung des Elements xs:pattern ist nur eine von vielen Möglichkeiten, den Wertebereich eines einfachen Datentyps einzuschränken. Elemente wie xs:pattern, die innerhalb von xs:restriction für derartige Einschränkungen eingesetzt werden können, werden Facetten genannt. Eine komplette Liste aller möglichen Facetten findet sich in [Fallside04].

XML Schema-Konzepte

Kompositoren Ein Kompositor legt die Anordnung von Elementen fest. Ein möglicher Kompositor ist xs:seqence, durch dieses Element wird die Reihenfolge der Elemente definiert. Eine Sequenz wird im obigen Beispiel zur Definition des *buch*-Elements verwendet (siehe Listing 2.4).

XML Schema definiert zwei weitere Kompositoren: xs:choice und xs:all. xs:choice steht für eine Disjunktion, d.h. von den enthaltenen Elementen muss genau eines im Ziel-Dokument erscheinen. Das Element xs:all repräsentiert eine Konjunktion: Von den enthaltenen Elementen dürfen alle in beliebiger Reihenfolge im Ziel-Dokument auftreten. Jedes der Elemente darf dabei 0 oder 1 mal vorkommen (vgl. [Thompson04]). Diese Elemente können verschachtelt werden, um komplexe Elementstrukturen zu erzeugen [Harold03].

Inhaltsmodelle Ein Element kann verschiedene Arten von Inhalten haben, in XML Schema gibt es vier Inhaltsmodelle (vgl. [Harold03]):

- leeres Inhaltsmodell
- einfacher Inhalt
- gemischter Inhalt
- uneingeschränkter Inhalt

Leeres Inhaltsmodell Die zu einem Element gehörenden Informationen werden nicht immer durch dessen Kindelemente modelliert, sondern können ihren Informationsgehalt auch vollständig über Attribute oder über ihre Position im Verhältnis zu anderen Elementen vermitteln. Ein Beispiel für ein Element, das die Information in Attributen enthält, ist in Listing 2.5 ersichtlich:

Listing 2.5: Beispiel für ein Element mit leerem Inhaltsmodell

```
<person vorname="Charlie" nachname="Brown"/>
```

Um ein Element mit leerem Inhaltsmodell zu definieren, definiert man einen Typ, in dem nur Elemente enthalten sein dürfen, ohne Elemente für dessen Inhalt zu deklarieren. Dadurch entsteht ein leeres Element (vgl. [Fallside04]). Das Beispielelement *person* aus Listing 2.5 kann in XML Schema folgendermaßen modelliert werden:

Listing 2.6: Deklaration eines Elements mit leerem Inhaltsmodell in XML Schema

```
1 <xs:element name="person">
2   <xs:complexType>
3     <xs:attribute name="vorname" type="xs:string"/>
4     <xs:attribute name="nachname" type="xs:string"/>
5   </xs:complexType>
6 </xs:element>
```

Da der im Element *person* enthaltene xs:complexType kein xs:element enthält, hat *person* ein leeres Inhaltsmodell.

Einfacher Inhalt In dem einführenden Beispiel in Abschnitt 2.1.2 wurde schon eine Möglichkeit vorgestellt, um Elemente mit einfachem Inhalt zu definieren: dafür kann das Element xs:simpleType verwendet werden. Die so deklarierten Elemente können aber weder Kindelemente noch Attribute haben. Daher kann das Element xs:complexType auch eingesetzt werden, um einen neuen komplexen Typ zu definieren, der von einem einfachen Typ abgeleitet ist. Ein Beispiel ist ein Element, das im Instanz-Dokument folgende Form haben soll:

Listing 2.7: Element mit Attribut und einfachem Inhalt

```
<flaschenInhalt masseinheit="liter">1.5</flaschenInhalt>
```

Dieses Element hat einen einfachen Inhalt vom Typ xs:decimal, gleichzeitig aber auch ein Attribut. Das entsprechende XML Schema-Element kann so definiert werden:

Listing 2.8: Deklaration eines Elements mit Attribut und einfachem Inhalt

```
1 <xs:element name="flaschenInhalt">
2   <xs:complexType>
3     <xs:simpleContent>
4       <xs:extension base="xs:decimal">
5         <xs:attribute name="masseinheit" type="xs:string"/>
```

```
6         </xs:extension>
7       </xs:simpleContent>
8     </xs:complexType>
9 </xs:element>
```

Das Element xs:complexType definiert einen neuen Typ. Dieser Typ enthält nur Zeichendaten und keine Elemente, das wird durch das Element xs:simpleContent ausgesagt. Das xs:extension-Element drückt aus, dass der im Attribut *base* genannte Datentyp erweitert wird, im Beispiel wird der Typ dadurch erweitert, dass ein Attribut hinzugefügt wird.

Gemischter Inhalt XML Schema erlaubt die Deklaration von Elementen, die Kindelemente und Text auf gleicher Ebene enthalten. Eine solche Mischung von Text und Kindelementen zeigt das Beispiel in Listing 2.9.

Listing 2.9: Element mit gemischtem Inhalt

```
1 <ankuendigung>
2   Die Veranstaltung
3   <name>Kinderuni</name>
4   findet am
5   <datum>2007-06-20</datum>
6   <adresse>
7     in der
8     <strasse>Universitätsstrasse</strasse>
9     <hausnummer>1</hausnummer>
10    in
11    <ort>Koblenz</ort>
12  </adresse>
13  statt.
14 </ankuendigung>
```

Eine Schema-Deklaration für dieses Element ist in Listing 2.10 aufgeführt. Die Elemente *ankuendigung* und *adresse* erlauben Zeichendaten zwischen ihren Kindelementen, da das *mixed*-Attribut in der zugehörigen Typdefinition auf true gesetzt ist.

Listing 2.10: Schema-Deklaration für Element mit gemischtem Inhalt

```
1 <xs:element name="ankuendigung">
2   <xs:complexType mixed="true">
3     <xs:sequence>
4       <xs:element name="name" type="xs:string"/>
5       <xs:element name="datum" type="xs:date"/>
6       <xs:element name="adresse">
7         <xs:complexType mixed="true">
8           <xs:sequence>
9             <xs:element name="strasse" type="xs:string"/>
10            <xs:element name="hausnummer"
11              type="xs:positiveInteger"/>
```

```
12              <xs:element name="ort" type="xs:string"/>
13            </xs:sequence>
14          </xs:complexType>
15        </xs:element>
16      </xs:sequence>
17    </xs:complexType>
18  </xs:element>
```

Uneingeschränkter Inhalt Die am wenigsten restriktive Möglichkeit, den Inhaltstyp eines Elements zu definieren, ist die Verwendung des Typs *anyType*. Dieser Typ schränkt den Inhalt des zugehörigen Elements nicht ein, im Instanz-Dokument kann dieses Element also beliebige Werte enthalten (vgl. [Fallside04]).

Listing 2.11: Element mit beliebigem Inhalt

```
<xsd:element name="zusatzInformation" type="xs:anyType"/>
```

Daneben gibt es auch sogenannte Wildcard-Schemakomponenten, darunter fallen das Element xs:any sowie das Attribut xs:anyAttribute. An der Stelle des xs:any-Elements darf im Instanz-Dokument ein beliebiges wohlgeformtes XML-Element stehen. Dieses Element kann z.B. eingesetzt werden, um Erweiterungen zuzulassen.

Das Einfügen beliebiger Attribute zu einem gegebenen Element kann durch Angabe des xs:anyAttribute-Elements erlaubt werden (vgl. [Fallside04]).

2.1.3 XPath

XPath ist eine Anfragesprache des W3C zur Adressierung von Teilen eines XML-Dokuments. XPath ermöglicht die Navigation durch die hierarchische Struktur eines XML-Dokuments. Die Basis für diese Sprache bildet die Betrachtung eines XML-Dokuments als Baum, der aus Knoten besteht. XPath kennt sieben Knotentypen (vgl. [Clark99]):

- **Wurzelknoten:** Es gibt in der Baumrepräsentation genau einen Wurzelknoten, dieser enthält als Kindknoten das Wurzelelement des XML-Dokuments sowie Knoten für alle Verarbeitungsanweisungen und Kommentare, die vor oder nach dem Wurzelelement des XML-Dokuments auftreten.

- **Elementknoten:** Für jedes Element in einem XML-Dokument gibt es einen Elementknoten. Sind in dem Element des XML-Dokuments Elemente, Kommentare, Verarbeitungsanweisungen oder Text enthalten, so werden entsprechende XPath-Knoten als Kinder des Elementknotens eingefügt.

- **Textknoten:** In Textknoten werden in einem Element enthaltene Zeichendaten zusammengefasst. Nicht für alle Zeichendaten werden Textknoten erzeugt: Zeichen, die in Kommentaren, Verarbeitungsanweisun-

gen oder Attributwerten enthalten sind, werden nicht durch Textknoten repräsentiert.

- **Attributknoten:** Attributknoten werden für Attribute, die in einem Element enthalten sind, erzeugt. Ein Ausnahmefall sind Attribute, die Namensräume deklarieren, für diese Attribute werden Namensraumknoten erzeugt.
- **Namensraumknoten:** Jedem Element ist eine Menge von Namensraumknoten zugewiesen. Diese Menge setzt sich zusammen aus Namensraumknoten für jeden Namensraum, dem das Element zugeordnet ist, unabhängig davon, ob diese Zuordnung im Element selbst oder über ein Vorfahrenelement des Elements erfolgt.
- **Verarbeitungsanweisungsknoten:** Für jede Verarbeitungsanweisung wird ein entsprechender Verarbeitungsanweisungsknoten erzeugt, sofern sie nicht innerhalb einer DTD erscheint.
- **Kommentarknoten:** Für jeden Kommentar wird ein entsprechender Kommentarknoten erzeugt, sofern er nicht innerhalb einer DTD erscheint.

Lokalisierungspfade Um Knoten zu selektieren, dient in XPath ein sogenannter Lokalisierungspfad. Dieser Pfad kann relativ oder absolut angegeben werden. Ein absoluter Lokalisierungspfad beginnt mit dem Zeichen /, welches den Wurzelknoten auswählt. Darauf kann optional ein relativer Lokalisierungspfad folgen, der den Pfad zum gesuchten Element relativ zum Wurzelknoten angibt.

Ein Beispiel für einen absoluten Lokalisierungspfad, der einen Elementknoten lokalisiert, ist folgender XPath-Ausdruck:

```
/buch/autor/vorname
```

Dieser Pfad wählt den Elementknoten, der das Element *vorname* repräsentiert. Das zu diesem Beispielpfad passende XML-Dokument ist in Listing 2.1 in Abschnitt 2.1.1 aufgeführt.

Ein Lokalisierungspfad wird durch eine Folge von Lokalisierungsschritten erzeugt, die durch das Zeichen / voneinander getrennt sind. Jeder Lokalisierungsschritt wählt eine Knotenmenge relativ zu einem Kontextknoten aus. Bei absoluten Pfaden ist der Wurzelknoten der Kontextknoten. Vom Wurzelknoten aus wird der erste Lokalisierungsschritt ausgewertet. Dieser selektiert im vorliegenden Beispiel alle Elemente mit dem Namen *buch*. Jeder Knoten dieser Ergebnismenge wird als Kontextknoten für den folgenden Schritt verwendet. So wird für alle Lokalisierungsschritte verfahren. Das Ergebnis dieser Auswertung ist eine Menge von Knoten. Hätte das Element *autor* des obigen Beispiels mehrere *vorname*-Elemente, so würde der Beispielpfad alle diese Elemente auswählen. Ein relativer Lokalisierungspfad könnte z.B. so aussehen:

```
autor/vorname
```

Dieser Pfad würde vom aktuellen Kontextknoten aus ausgewertet und würde daher die *vorname*-Kindelemente der *autor*-Kindelemente dieses Kontextknotens selektieren.

Um nicht nur die Kindelemente, sondern alle Abkömmlinge des Kontextknotens zu selektieren, kann der doppelte Schrägstrich verwendet werden. Der XPath-Ausdruck `//nachname` wählt z.B. alle *nachname*-Elemente im gesamten Dokument, da hier der Kontextknoten der Wurzelknoten ist.

Auch Attribute können durch Lokalisierungspfade ausgewählt werden. Ein Attributname wird durch ein vorangestelltes `@`-Zeichen gekennzeichnet. Der folgende Pfad wählt das *isbn*-Attribut des Elements *buch* aus:

```
/buch/@isbn
```

Andere Knotentypen können mit speziellen Lokalisierungsschritten ausgewählt werden: Textknoten können durch `text()`, Kommentarknoten durch `comment()` und Verarbeitungsanweisungen durch `processing-instruction()` selektiert werden.

In einen Lokalisierungspfad können auch Wildcards eingebaut werden, die unterschiedliche Element- und Knotentypen ansprechen. XPath kennt drei Wildcards: `*`, `node()` und `@*`. Der Asterisk (`*`) steht für alle Elementknoten, spricht aber keine Attribute, Textknoten, Kommentare oder Verarbeitungsanweisungen an.

Die Wildcard `node()` wählt nicht nur Elementtypen aus, sondern auch Text-, Verarbeitungsanweisungs-, Namensraum-, Attribut- und Kommentarknoten. Die Wildcard `@*` selektiert alle Attributknoten (vgl. [Harold03]).

Prädikate Um die Knotenmenge, die ein Lokalisierungsschritt selektiert, weiter einzuschränken, kann in jedem Lokalisiserungsschritt ein Prädikat angegeben werden. Das Prädikat enthält einen Booleschen Ausdruck. Dieser Ausdruck wird für jeden in der Kontextknotenliste enthaltenen Knoten ausgewertet. Nur wenn der Ausdruck wahr ist, wird der Knoten in der Menge der Knoten belassen, die im nächsten Lokalisierungsschritt verarbeitet werden (vgl. [Harold03]).

Prädikate werden in eckigen Klammern angegeben. Der XPath-Ausdruck `//autor [vorname="Douglas"]` selektiert alle *autor*-Elemente, die ein *vorname*-Element besitzen, das den Wert „Douglas“ hat. Auch Attributwerte können zur Filterung angegeben werden, der Ausdruck `//buch [@isbn="978-345314-6976"]` liefert das *buch*-Element zurück, das die angegebene ISBN-Nummer im *isbn*-Attribut beinhaltet.

Neben dem Gleichheitszeichen können weitere Vergleichszeichen verwendet werden. Der Ausdruck `/buch[seiten<500]` selektiert alle *buch*-Elemente, die ein Kindelement *seiten* mit einem numerischen Wert kleiner 500 haben. XPath unterstützt die folgenden Vergleichsoperatoren: `=`, `!=`, `<`, `>`, `<=` und `>=`. Auch die Booleschen Operatoren `and` und `or` können verwendet werden (vgl. [Harold03]). Der Ausdruck `//autor [nachname="Adams" or nachname="Davis"]` wählt alle *autor*-Elemente, die ein *nachname*-Kindelement mit den Werten `Adams` oder `Davis` haben.

Funktionen In XPath-Ausdrücken können auch Funktionen zum Einsatz kommen. XPath stellt eine große Anzahl von Funktionen bereit. Jede dieser Funktionen gibt einen der folgenden vier Typen zurück (vgl. [Harold03]):

- Boolescher Wert
- Zahl
- Knotenmenge
- String

Ein Beispiel für eine Funktion, die einen Wahrheitswert zurückgibt, ist `starts-with(string1, string2)`. Falls das erste Argument `string1` mit dem zweiten Argument `string2` beginnt, liefert die Funktion `true` zurück. Einen String liefert beispielsweise die Funktion `concat(string1, string2, ...)` zurück. Die Funktion konkateniert die übergebenen Argumente.

Eine Funktion, die eine Zahl zurückgibt, ist z. B. `round(zahl)`. Die Funktion rundet das Argument `zahl` auf die nächstgelegene ganze Zahl. Eine vollständige Liste aller verfügbaren XPath-Funktionen enthält [Clark99].

2.2 Einführung in P3P

Beim Surfen im Internet scheint der Nutzer bei oberflächlicher Betrachtung anonym zu sein. In der Regel hinterläßt er aber Datenspuren bei den besuchten Web-Angeboten, z.B. in den Log-Dateien eines Webservers, durch Informationen, die sein Browser bei einer Anfrage mitsendet, durch das Annehmen von Cookies etc. (vgl. [Lindskog03]). In [Cranor02a] wird in diesem Zusammenhang auf eine wachsende Besorgnis der Internetnutzer über Art und Umfang online hinterlassener Daten sowie über deren Verwendung hingewiesen. Als problematisch wird auch der Aufwand empfunden, der mit dem Suchen, Lesen und besonders mit dem Verstehen der Datenschutzerklärungen eines Web-Angebots verbunden ist. Oft fällt es Nutzern schwer, herauszufinden, wie sie einen datensparsamen Umgang mit ihren personenbezogenen Daten einfordern könnten. Medienberichte über besonders „wissbegierige“ Unternehmen oder Institutionen tun ein Übriges, um das Vertrauen der Nutzer in Web-Angebote zu schmälern. Daher ist es auch aus Sicht der Anbieter sinnvoll, dem Misstrauen von Nutzern und damit potentiellen Kunden mit möglichst großer Transparenz zu begegnen. Ein Ansatz, der versucht, mehr Transparenz hinsichtlich der Datenschutzpraxis eines Web-Angebots zu schaffen, ist P3P.

P3P ist die Abkürzung für *Platform for Privacy Preferences* und ist ein vom W3C entwickelter Standard, der einen Abgleich zwischen den Datenschutzpraktiken eines Web-Angebots und den entsprechenden Präferenzen eines Nutzers ermöglicht. Damit dieser Abgleich automatisiert erfolgen kann, bietet P3P die Möglichkeit, beides in maschinenlesbarer Form zu formulieren. Dies ermöglicht es dem Nutzer, eine schnelle Übersicht über die Policy eines Anbieters zu erhalten. Diese enthält Angaben über die Datenschutzpraktiken

des Web-Angebots. Dazu zählen laut Spezifikation auf die Daten bezogene Angaben wie z.B. die Art der erhobenen Daten, der Zweck, für den sie erhoben werden, und ob die Daten an Dritte weitergegeben werden. Zudem werden allgemeinere Aussagen gemacht, diese betreffen u.a. Möglichkeiten, den Anbieter zu kontaktieren, oder Anlaufstellen für Beschwerden und die Lösung von Streitigkeiten (vgl. [Wenning06]). Auf der Grundlage des automatisierten Abgleichs sowie der durch den Standard strukturierten Übersicht über die Praktiken des Anbieters kann der Nutzer über sein weiteres Verhalten entscheiden.

Hat ein Anbieter sich dafür entschieden, eine P3P-Policy zu veröffentlichen, so muss er neben dieser Policy auch eine menschenlesbare Datenschutzerklärung anbieten, die für juristische Belange verwendbar ist (vgl. [Cranor02a]). Dies hat zudem den Vorteil, dass Details, die durch das vom P3P-Standard vorgegebene Vokabular nicht ausgedrückt werden können, hier näher erläutert werden können.

2.3 Funktionsweise von P3P

P3P funktioniert nach dem folgenden Prinzip (vgl. [Cranor02a]): Ein Anbieter veröffentlicht eine maschinenlesbare Policy sowie eine maschinenlesbare Policy-Referenzdatei. Ein Nutzer, der auf dieses Web-Angebot zugreifen möchte, verwendet einen sogenannten Nutzeragenten, damit ist die P3P-Implementation auf Client-Seite gemeint. Diese kann in einen Browser oder in andere Tools integriert sein, oder aber als eigenständiges Programm arbeiten. Hier können die Nutzerpräferenzen festgelegt werden.

Die Policy-Referenzdatei dient dem Nutzeragenten zum Auffinden der Policy. In dieser Datei wird definiert, für welche Bereiche eines Webangebots welche Policy gelten soll. Eine Policy kann für Webseiten, aber auch für Cookies gelten. Demnach fordert der Nutzeragent zuerst die Policy-Referenzdatei an. Aus dieser Datei entnimmt er, wo er die Policy für die Webseite findet, die er anfordern möchte. Nun kann er die geeignete Policy anfordern. Daraufhin kann er die Policy parsen, sie mit den Nutzerpräferenzen vergleichen und dem Nutzer das Ergebnis mitteilen. Der Nutzer kann auf dieser Basis entscheiden, ob er die Webseite anfordern möchte, oder nicht. Dieser Ablauf ist in Abb. 2.1 dargestellt.

Diese Mechanismen basieren auf dem Zusammenspiel der einzelnen Komponenten von P3P (vgl. [Langheinrich01]):

- Daten-Schema: Das sogenannte *P3P Base Data Schema* definiert verschiedene Arten von Daten, die ein Web-Angebot sammeln könnte, Beispiele hierfür sind Name, Mailadresse, IP-Adresse, etc.

- Praktiken-Vokabular: Mit diesem Vokabular kann ein Anbieter definieren, wie er mit obigen Daten verfährt, d.h. welche der obigen Daten wann und für wie lange gespeichert werden, für welchen Zweck sie verwendet werden und an wen sie weitergeleitet werden.

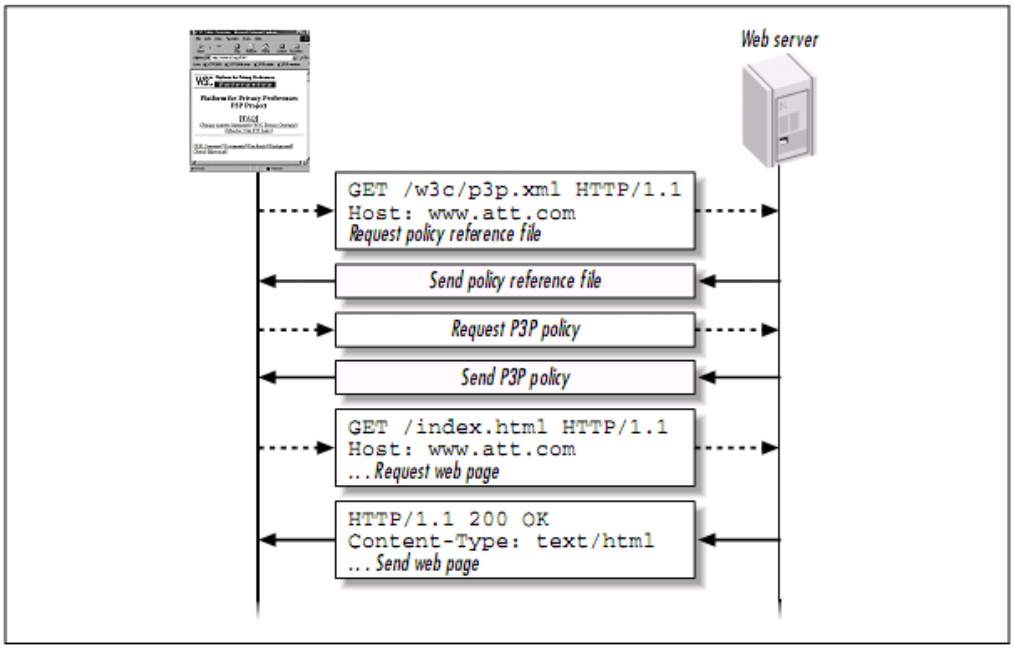

Abbildung 2.1: Protokoll zum Einholen einer Policy (Quelle: [Cranor02a], S.5)

- XML-Syntax: Um Maschinenlesbarkeit gewährleisten zu können, definiert P3P eine XML-Syntax zur Formulierung von Policy- und Policy-Referenzdateien. Die Elemente der Syntax sind im XML-Schema-Namensraum `http://www.w3.org/2006/01/P3Pv11` festgelegt.
- Protokoll: Die Datenschutzpolicies werden über ein effizientes Protokoll (siehe Abb. 2.1) mit Web-Inhalten wie z.B. HTML-Seiten oder Grafiken verknüpft. Welcher Inhalt mit welcher Policy verknüpft ist, wird innerhalb der Policy-Referenzdatei festgelegt. Das P3P-Protokoll dient dazu, die Referenzdatei anzufordern, die passende Policy zu übermitteln und – bei Übereinstimmung mit den Nutzerpräferenzen – die angeforderte Webseite zu übertragen. Das P3P-Protokoll basiert auf HTTP.
- Präferenzsprache (APPEL): Diese separate, ebenfalls vom W3C entwickelte Sprache ermöglicht die Formulierung von Präferenzen auf Nutzerseite.

Ein Anbieter kann seine Policy-Referenzdatei auf verschiedene Arten veröffentlichen. Die vom W3C empfohlene Methode ist das Ablegen der Referenzdatei an der sog. „well-known location“, d.h. unter dem Pfad */w3c/p3p.xml*. Für das Webangebot *http://www.beispiel.de* wäre die Referenzdatei also unter *http://www.beispiel.de/w3c/p3p.xml* zu finden.

Nutzeragenten werden zunächst an diesem Ort nach der Referenzdatei suchen, was den Vorteil hat, dass die Datei vor anderen Inhalten des Webangebots angefordert wird. So kann die Evaluation der Policy vor der Nutzung des Angebots erfolgen (vgl. [Cranor02a]).

Eine Alternative zur „well-known location“ ist die Verwendung des P3P-Headers. Dies ist ein erweiterter HTTP-Header, der die URL der Referenzdatei enthält. Der P3P-HTTP-Header hat das folgende Format:

```
P3P: policyref="http://www.beispiel.de/p3p.xml"
```

Wird zudem eine Compact Policy mitgesendet, hat der Header folgendes Format (Näheres zu Compact Policies in Abschnitt 2.5):

```
P3P: policyref="http://www.beispiel.de/p3p.xml", CP="Compact
Policy"
```

Werden P3P-Header verwendet, um auf die Referenzdatei hinzuweisen, sollten diese Header in den Antworten auf alle Anfragemethoden verwendet werden (vgl. [Wenning06]).

Die dritte Möglichkeit, auf die Referenzdatei hinzuweisen, ist die Einbettung des Verweises in ein HTML- oder XHTML-Link-Tag. Ein Beispiel für eine einfache HTML-Seite, die ein solches Link-Tag enthält, ist in Listing 2.12 aufgeführt (Quelle: [Cranor02a], S.148).

Listing 2.12: Beispiel für ein HTML-Dokument mit Link-Tag

```
<HTML>
<HEAD>
    <LINK rel="P3Pv1" href="http://www.example.com/
          p3p-prf.xml">
    <TITLE>Sample Page</TITLE>
</HEAD>
<BODY>
    <P>This is a sample page.</P>
</BODY>
</HTML>
```

Diese Methode bietet sich an, wenn der Anbieter keinen Zugriff auf die „well-known location“ hat, und die Serverkonfiguration nicht ändern kann. Wird dieses Verfahren gewählt, müssen alle HTML-Seiten des Webangebots, für die eine P3P-Policy gilt, das Link-Tag enthalten (vgl. [Cranor02a]).

2.4 P3P-Vokabular

Das Element META enthält die Policy-Referenzdatei. Diese wird durch das Element POLICY-REFERENCES definiert. Optional kann META auch das Element POLICIES enthalten, das Policies in Form von POLICY-Elementen beinhaltet. Die beiden Elemente POLICIES und POLICY-REFERENCES sind die zentralen Bestandteile der P3P-Templates, die im Rahmen dieser Arbeit entwickelt wurden. Ihr Inhalt wird in den folgenden Abschnitten vorgestellt, diese Darstellung orientiert sich an der P3P-Spezifikation und bietet einen Überblick über die wesentlichen Aspekte der einzelnen Elemente. Eine detaillierte Erläuterung der einzelnen Attribute und Subelemente ist in der Spezifikation zu finden (vgl. [Wenning06]).

2.4.1 Vokabular für Policy-Referenzen

Mit Hilfe einer Policy-Referenzdatei wird festgelegt, für welchen Bereich eines Webangebots welche Policy gilt. Das Element POLICY-REFERENCES kann folgende Subelemente enthalten:

- EXPIRY: Dieses Element wird zur Definition der Geltungsdauer von POLICY-REFERENCES und POLICIES verwendet. Der Standardwert für dieses Element ist eine relative Geltungsdauer von einem Tag. Diese Zeitspanne beginnt, wenn der Nutzer die entsprechende P3P-Datei angefragt hat. Alternativ zu einer relativen Dauer kann die Geltungsdauer auch durch die Angabe eines absoluten Datums definiert werden.

- POLICY-REF: Dieses Element enthält Informationen zum Geltungsbereich einer Policy. Im Attribut *about* muss die Policy, für die der Geltungsbereich definiert wird, durch einen URI referenziert werden. POLICY-REF kann folgende Subelemente enthalten:
  - INCLUDE und EXCLUDE: Mit diesen Elementen können URIs in den Geltungsbereich einbezogen bzw. vom Geltungsbereich ausgeschlossen werden. Die URIs müssen relativ zu dem Host angegeben werden, auf dem die Policy-Referenzdatei liegt. Mit der Wildcard * können Mengen von URIs definiert werden, mit dem relativen URI /* ist beispielsweise das gesamte Webangebot gemeint.
  - COOKIE-INCLUDE und COOKIE-EXCLUDE: Mit diesen beiden Elementen können Cookies in den Geltungsbereich einbezogen bzw. von diesem ausgeschlossen werden. Cookies können mit den Attributen *name*, *value*, *domain* und *path* definiert werden. Auch für diese Attribute darf die Wildcard * verwendet werden.
  - METHOD: Durch dieses Element kann der Geltungsbereich auf bestimmte HTTP-Anfragemethoden beschränkt werden.

- HINT: Dieses Element dient zur Angabe von Verweisen auf die Policy-Referenzdateien anderer Anbieter. Es kann z.B. für Hyperlinks oder eingebettete Inhalte eingesetzt werden, über die auf das Angebot eines weiteren Anbieters zugegriffen werden kann.

- EXTENSION: Mit Hilfe dieses Elements kann die P3P-Syntax flexibel erweitert werden. Das Element EXTENSION kann in den meisten P3P-Elementen als Kindelement enthalten sein. Die XML Schema-Definition des Elements läßt beliebigen Inhalt zu. Eine Erweiterung kann als optional oder zwingend erforderlich deklariert werden. Dazu dient das Attribut *optional*, das mit „yes“ oder „no“ belegt werden kann. Dieses Attribut ist für Tools auf Nutzerseite von Bedeutung: Ein Nutzeragent, der eine zwingend erforderliche Erweiterung nicht versteht, muss daraus folgern, dass er die das Element umgebende Policy oder Policy-Referenzdatei nicht versteht. Ist dies der Fall, behandelt der Nutzeragent das Webangebot so, als hätte es keine P3P-Policy. Es wird empfohlen, dass die in einer

Erweiterung enthaltenen Elemente durch einen Namensraum eindeutig identifiziert werden, dieser kann im *xmlns*-Attribut des Elements angegeben werden (vgl. [Cranor02a]).

2.4.2 Vokabular für eine Policy

Das für die Formulierung einer Policy benötigte Vokabular läßt sich in zwei Kategorien aufteilen: Elemente für allgemeine Aussagen (vgl. 2.4.2) sowie für Angaben, die sich auf die erhobenen Daten beziehen (vgl. 2.4.2).

Allgemeine Angaben

Eine Policy ist im Element POLICIES enthalten und wird eingeleitet mit dem POLICY-Element. In den dazugehörigen Attributen muss der Policy unter *name* ein eindeutiger Name sowie unter *discuri* der URI zugeordnet werden, unter dem die entsprechende menschenlesbare Datenschutzerklärung eingesehen werden kann. Sofern der Nutzer die Möglichkeit hat, der Verwendung seiner Daten zu einem bestimmten Zweck zuzustimmen (sog. „Opt-In“) oder diese abzulehnen (sog. „Opt-Out“), muss im Attribut *opturi* ein URI angegeben werden, der diesbezüglich eine natürlichsprachliche Anleitung enthält. Ein POLICY-Element hat die folgenden Subelemente:

- ENTITY: Dieses Element dient zur Beschreibung der juristischen Person, die für das Web-Angebot verantwortlich ist. Deren Name und mindestens eine Kontaktinformation wie z.B. Postanschrift, Telefonnummer oder E-Mailadresse müssen hier angegeben werden.
- ACCESS: Mit diesem Element definiert ein Anbieter, zu welchen über ihn gespeicherten personenbezogenen Daten er dem Nutzer Zugang gewährt. Die Zugangsmethode wird dabei nicht spezifiziert. Daher sollte ein Anbieter, der solche Zugangsmöglichkeiten bietet, Näheres dazu in seiner natürlichsprachlichen Datenschutzerklärung erläutern (vgl. [Cranor02a]).
- DISPUTES: Hier können eine oder mehrere Vorgehensweisen angegeben werden, um Streitigkeiten im Bezug auf die Datenschutzpraktiken eines Anbieters zu lösen. Dieses Element ist optional, die P3P-Spezifikation empfiehlt aber, es zu verwenden. Einer der folgenden Wege zur Konfliktlösung muss im Attribut *resolution-type* verwendet werden:
  - der Kundendienst des Anbieters (*service*),
  - eine unabhängige Organisation (*independent*), von der dem Anbieter möglicherweise ein Siegel oder Zertifikat ausgestellt wurde,
  - das zuständige Gericht (*court*),
  - das geltende Gesetz (*law*).

  Ein weiteres verpflichtendes Attribut ist *service*, hier muss der URI der in *resolution-type* genannten Organisation angegeben werden. Zudem können im Rahmen des DISPUTES-Elements längere natürlichsprachliche

Erläuterungen sowie Logos von Siegeln oder Zertifikaten eingebunden werden.

  - REMEDIES: Dies ist ein Subelement von DISPUTES, und sollte verwendet werden, um darzulegen, wie im Falle eines Verstoßes des Anbieters gegen seine Policy verfahren werden soll. Folgende Möglichkeiten sind verfügbar:
    * Beheben des Fehlers (*correct*)
    * Finanzielle Entschädigung (*money*)
    * Regelung durch Gesetze, die in der natürlichsprachlichen Policy genannt werden (*law*)

- TEST: Dieses Element ist optional. Ist es in eine Policy eingebunden, so signalisiert dies dem Nutzeragenten, dass die Policy nur Testzwecken dient, nicht gültig ist und daher ignoriert werden muss.

- STATEMENT-GROUP-DEF: Mit diesem Element können Statement-Gruppen definiert werden. Ein Statement enthält datenspezifische Angaben (siehe Abschnitt 2.4.2). In einer Policy können mehrere Statements enthalten sein, diese können Statement-Gruppen zugeordnet werden, um dem Nutzer die Übersicht zu erleichtern. Diesen Gruppierungsmechanismus gibt es seit P3P Version 1.1, seine Verwendung ist optional. Das Element STATEMENT-GROUP-DEF enthält folgende Attribute (vgl. [Wenning06]):
  - *id*: Mit diesem Attribut wird ein eindeutiger Name für die Statement-Gruppe vergeben.
  - *short-description*: Eine kurze natürlichsprachliche Beschreibung kann durch dieses Attribut angegeben werden (optional).
  - *consent*: Dieses optionale Attribut gibt an, ob der Nutzer die Möglichkeit hat, alle Statements dieser Gruppe – im Hinblick auf die durch sie definierten Verwendungszwecke und Datenempfänger – geschlossen zu akzeptieren bzw. abzulehnen. Diese Aussage hängt mit dem *required*-Attribut der Zweck- und Empfängerelemente zusammen (siehe PURPOSE und RECIPIENT in Abschnitt 2.4.2). Den möglichen Werten für dieses Attribut entsprechen die für das *consent*-Attribut wählbaren Werte:
    * *opt-in*: Der Nutzer kann alle Statements der Gruppe geschlossen akzeptieren (explizite Einwilligung).
    * *opt-out*: Der Nutzer kann alle Statements der Gruppe geschlossen ablehnen (expliziter Widerspruch).
    * *always*: Dieser Wert sagt aus, dass der Nutzer weder explizit einwilligen, noch explizit ablehnen kann, sondern dass die Erhebung der deklarierten Daten erforderlich ist.

  - *mixed*: Dieser Wert wird verwendet, wenn es zwar für einzelne Statements der Gruppe „opt-in“- bzw. „opt-out“-Möglichkeiten gibt, diese Zustimmung oder Ablehnung aber nicht geschlossen für die ganze Gruppe vorgenommen werden kann.

Datenspezifische Angaben

Zur Formulierung der datenspezifischen Angaben werden pro Policy ein oder mehrere STATEMENT-Elemente verwendet. Jedes Statement sollte das Datenschutzverhalten des Anbieters im Bezug auf eine bestimmte Gruppe von Daten repräsentieren (vgl. [Cranor02a]). Beispielhaft für solche Gruppen sind Daten für administrative Zwecke (wie die Logdatei des Web-Servers) oder Daten, die für die Zustellung einer Bestellung benötigt werden. In einem STATEMENT-Element sind die folgenden Subelemente enthalten:

- STATEMENT-GROUP: Dieses optionale Element wird in P3P Version 1.1 definiert und kann verwendet werden, um ein Statement einer Statement-Gruppe zuzuordnen. Zur Definition der Gruppe dient das Element STATEMENT-GROUP-DEF, dort wird jeder Statement-Gruppe ein eindeutiger Bezeichner zugewiesen (siehe Abschnitt 2.4.2). Durch einen solchen Bezeichner verweist das Attribut *id* des Elements STATEMENT-GROUP auf die Definition der Gruppe.

- NON-IDENTIFIABLE: Dieses Element ist optional und kann verwendet werden, um auszudrücken, dass unter diesem Statement entweder keine Daten erhoben werden, oder aber alle Daten bei der Erhebung anonymisiert werden, so dass es für den Anbieter oder Dritte nicht möglich ist, diese Daten einer Person zuzuordnen. Die Transformation zur Anonymisierung dieser Daten darf nicht umkehrbar sein, und es muss in der natürlichsprachlichen Policy erläutert werden, wie die Daten anonymisiert werden. Falls dieses Element verwendet wird, sind alle folgenden Elemente optional.

- CONSEQUENCE: Dieses Element ist optional und dient dazu, natürlichsprachliche Erläuterungen zu den Datenschutzpraktiken bereitzustellen, die in diesem Statement definiert werden.

- PURPOSE: Mit diesem Element werden ein oder mehrere Zwecke angegeben, für die die Daten gesammelt werden. Jeder der aufgeführten Zwecke kann das Attribut *required* spezifizieren, das aussagt, ob dieser Zweck erforderlich ist, oder ob der Nutzer diesbezüglich eine Wahl hat. Für dieses Attribut gibt es folgende Werte:
  - always: Der Nutzer hat keine Wahlmöglichkeit. Dies ist der Standardwert, falls required nicht gesetzt ist.
  - opt-in: Für diesen Zweck dürfen die Daten nur dann verwendet werden, wenn der Nutzer ausdrücklich in diese Verwendung zustimmt. In diesem Fall muss der Nutzer die Möglichkeit haben, seine Zustimmung zu widerrufen. Wie er einen solchen Widerruf geltend

machen kann, muss durch den URI, der dem Attribut *opturi* des POLICY-Elements zugeordnet wird, natürlichsprachlich erläutert werden (vgl. 2.4.2).

– opt-out: Für diesen Zweck dürfen die Daten verwendet werden, sofern der Nutzer dieser Verwendung nicht ausdrücklich widerspricht. Instruktionen für einen solchen Widerspruch müssen ebenfalls durch das *opturi*-Attribut bereitgestellt werden.

Durch das Subelement PPURPOSE kann der primäre Grund für die Erhebung der Daten hervorgehoben werden (ab P3P Version 1.1).

- RECIPIENT: Durch dieses Element werden der oder die Empfänger der erhobenen Daten aufgeführt. Hier wird deklariert, ob diese Daten an Dritte weitergegeben werden, und welchen Datenschutzpraktiken sie bei diesen Dritten unterliegen. Dabei müssen alle Empfänger angegeben werden, die Zugriff auf die Daten erlangen können. Analog zum Element PURPOSE kann auch für RECIPIENT das *required*-Attribut verwendet werden, das angibt, ob die Weitergabe der Daten an diesen Empfänger erforderlich ist, oder ob der Nutzer dieser Verwendung explizit zustimmen oder diese ablehnen kann. Im Subelement JURISDICTION kann die für einen Empfänger zuständige Gerichtsbarkeit genannt werden (ab P3P Version 1.1).

- RETENTION: In diesem Element wird definiert, wie lange die erhobenen Daten gespeichert werden.

- DATA-GROUP: Dieses Element dient zur Beschreibung der Datenarten, die ein Anbieter sammelt. Hier können die Elemente benutzt werden, die durch das sog. *Base Data Schema* definiert werden. Die Elemente dieses Schemas sind hierarchisch aufgebaut, was sowohl präzise als auch allgemeine Deklarationen zuläßt. Ein Anbieter, der z.B. das Element *user.online.email* verwendet, macht eine sehr genaue Angabe, wohingegen die Verwendung des Wurzelelements *user* automatisch alle Subelemente von *user* mit einschließt, wie z.B. Name, Heimat- und Geschäftsadresse, Geburtsdatum, Geschlecht, Beruf, Arbeitgeber, Kennung, Paßwort, etc.

  Den einzelnen Datenelementen sind Kategorien zugeordnet, die dem Nutzeragenten die Interpretation der Elemente erleichtern sollen. Meist existiert im *Base Data Schema* eine feste Zuordnung eines Elements zu einer oder mehreren Kategorien, einige wenige sind auch — abhängig von der Situation — variablen Kategorien zuzuteilen. Dazu dient das Element CATEGORIES.

Mit diesen Elementen können die wichtigsten Aussagen zum Umgang mit den erhobenen Daten maschinenlesbar gemacht werden. Auch eine natürlichsprachliche Präzisierung der durch P3P gemachten Angaben kann durch das CONSEQUENCE-Element transportiert werden.

2.5 Compact Policies

Neben den bisher beschriebenen ausführichen Policies enthält der P3P-Standard eine Syntax für sogenannte Compact Policies. Eine Compact Policy fasst die Policy, die für ein Cookie gilt, in einem String zusammen. In diese Kurzfassung fließen folgende Elemente der Policy ein: ACCESS, CATEGORIES, DISPUTES, NON-INDENTIFIABLE, PURPOSE, RECIPIENT, REMEDIES, RETENTION, TEST.

Die einzelnen Elemente der Policy werden jeweils durch eine kurze Zeichenfolge repräsentiert. Elemente, die beschreibenden Text enthalten, wie zum Beispiel CONSEQUENCE, sind in einer Compact Policy nicht vertreten. Eine Policy, die mehrere Statements enthält, wird nach P3P-Version 1.0 so in eine Compact Policy umgeformt, dass die Kindelemente der Statements „aggregiert“ sind, d.h. es ist in der Compact Policy nicht ersichtlich, welches Element zu welchem Statement gehört. P3P-Version 1.1 führt eine Syntax ein, die die zu einem Statement gehörenden Elemente zusammen gruppiert.

Der String, der die Compact Policy enthält, wird in den P3P-HTTP-Header gesetzt (siehe Abschnitt 2.3). Falls durch eine HTTP-Antwort mehrere Cookies gesetzt werden, muss die Compact Policy für alle diese Cookies gelten (vgl. [Cranor02a]).

Die P3P-Spezifikation sieht Compact Policies als optionale Optimierung: Die Verarbeitung von Cookies kann so möglicherweise performanter sein, sofern in der Compact Policy alle Informationen enthalten sind, die für den Abgleich mit den Nutzerpräferenzen benötigt werden. Ist dies nicht der Fall, sollte der Nutzeragent die vollständige Policy einholen (vgl. [Wenning06]). Einige Nutzeragenten verlassen sich allerdings bei der Entscheidung, ob ein Cookie angenommen werden soll, ausschließlich auf Compact Policies (vgl. Kapitel 7). Dies ist sinnvoll im Hinblick auf Server, die so konfiguriert sind, dass sie automatisch bei der ersten HTTP-Anfrage Cookies setzen. In diesem Fall kommt das Cookie vor der vollständigen Policy beim Nutzer an. Eine solche Konfiguration entspricht allerdings nicht der Empfehlung des W3C, eine „Safe Zone“ für das Einholen von Policies einzurichten. Innerhalb der „Safe Zone“ soll die Datenerhebung auf ein Minimum begrenzt und auf nicht personenbeziehbare Daten beschränkt werden. Daher sollen Cookies hier nicht eingesetzt werden (vgl. [Wenning06]).

2.6 Motivation für den Einsatz von P3P

Der P3P-Standard wurde in der Vergangenheit aus vielfachen Gründen kritisiert. Während der Entstehungsgeschichte des Standards bis zur Veröffentlichung der Version 1.0 wurde die ursprünglich geplante Funktionalität reduziert. In [Grimm00] werden die ersten Entwürfe beschrieben, die eine möglichst freie und gleichberechtigte Aushandlung des Datenschutzverhaltens eines Web-Angebots zwischen Anbieter und Nutzer enthielten. Dies hätte bedeutet, dass der Nutzer einen Vorschlag des Anbieters ablehnen kann, daraufhin dem Anbieter seine Vorstellungen übermittelt, dieser wiederum mit einem neuen Vorschlag reagieren kann usw., und diese Verhandlungen dadurch zu einem Ende kommen, dass entweder beide sich einigen, oder aber der Nutzer die Verbindung abbricht. Diese Aushandlung wurde im W3C als zu aufwändig betrachtet und daher nicht umgesetzt. Ebenfalls nicht verwirklicht wurde der Transfer von personenbezogenen Daten sowie eine explizite Einwilligung des Nutzers, dieser kann nach Sichtung der Policy die Dienste eines Web-Angebots lediglich nutzen und damit akzeptieren oder darauf verzichten (sog. „notice and choice"-Prinzip), nicht aber eigene Vorschläge machen.

Die letztendlich umgesetzte Funktionalität geht vielen Kritikern nicht weit genug, da sie dem Nutzer keine echte Wahlmöglichkeit zwischen mehreren Alternativen gibt, es besteht nur die Wahl zwischen „Ja" (Akzeptanz der Bedingungen) oder „Nein" (Verzicht auf Nutzung des Angebots). Dem Anbieter wird somit auch keine Rückmeldung über die Vorstellungen der Nutzer gegeben (vgl. [Cranor02a]).

Kritisiert wird auch, dass der P3P-Standard keinen Kontrollmechanismus vorsieht, der überprüft, ob die in einer Policy gemachten Aussagen der Wahrheit entsprechen (vgl. [Epi00]). P3P ist nur ein technischer Standard, daher betont der Bundesdatenschutzbeauftragte Schaar die Notwendigkeit einer ergänzenden Datenschutzkontrolle und präzisen Rechtsnormen zum Schutz der Privatsphäre (vgl. [Schaar02]). Auch die Voreinstellungen in den Tools, die P3P auf Nutzerseite unterstützen, sind Kritikern ein Dorn im Auge, da die Defaulteinstellungen oftmals ein sehr niedriges Datenschutzniveau repräsentieren. Dies könnte zur unbeabsichtigten Preisgabe von Daten und ungerechtfertigtem Vertrauen in die Praktiken eines Web-Angebots führen (vgl. [Epi00]). Schwierigkeiten kann auch das P3P-Vokabular bereiten, das Datenschutzpraktiken zwangsläufig unpräzise beschreibt, um Maschinenlesbarkeit ermöglichen zu können (vgl. [Ulber04]).

All diese Kritikpunkte zeigen, dass P3P allein nicht ausreicht, um die Privatsphäre des Nutzers zu schützen, dennoch gibt es positive Aspekte, die für einen Einsatz von P3P sprechen. Zunächst erhöht P3P die Transparenz und erleichtert es einem Nutzer, sich über die Datenschutzpraktiken eines Anbieters zu informieren (vgl. [Ulber04]). Schaar sieht den Einsatz von P3P und weiteren Datenschutz-Tools als möglichen Wettbewerbsvorteil für Unternehmen, dieser könnte besonders dann zum Tragen kommen, wenn er mit Datenschutz-Gütesiegeln verbunden wird (vgl. [Schaar02]). In [Grimm00] wird die Mög-

lichkeit der Herausbildung einer Datenschutzkultur beschrieben: P3P bewirkt sowohl auf Anbieter- als auch auf Nutzerseite eine Erhöhung des Bewußtseins für das eigene Datenschutzverhalten. Anbieter können durch P3P erklären, inwieweit sie mit ihrem Verhalten geltendem Datenschutzrecht entsprechen und über P3P Gütesiegel, Datenschutzauditzeichen o.ä. veröffentlichen. Verbraucherschutzverbände und Datenschutzbeauftragte könnten Orientierungshilfen bieten, indem sie „populäre Nutzerpräferenzen“ und „populäre Policies“ empfehlen. Die im Rahmen dieser Arbeit entwickelten Policies können nicht nur im Bereich der öffentlichen Stellen Verwendung finden, sondern sind auch auf andere Bereiche übertragbar. Sofern P3P eine breite Verwendung erfährt, wäre der Weg für Erweiterungen der Funktionalität des Standards geebnet.

Auf Anbieterseite erleichtert die erhöhte Transparenz einen Vergleich der Policies verschiedener Anbieter. So kann die Datenschutzfreundlichkeit von Diensten im Internet bewertet werden („Rating“). Durch gesetzeskonforme Präferenzen könnte erreicht werden, dass der Nutzer auf Angebote mit unzureichendem Datenschutz aufmerksam gemacht wird und diese meiden kann. Der Nutzer hat zudem die Möglichkeit, Dienste nicht aufzusuchen, die P3P nicht einsetzen. Verstöße gegen die in Deutschland geltenden rechtlichen Vorgaben wie z.B. Zweckbestimmung, Unterrichtungspflicht, Nutzerrechte etc. würden unmittelbar ersichtlich. Aus deutscher Sicht kann P3P als Ergänzung zum Datenschutzrecht gesehen werden. P3P kann Datenschutzverhalten nur beschreiben, während die rechtlichen Regelungen zur Durchsetzung von Vereinbarungen dienen (vgl. [Grimm00]).

Die hier genannten Vorteile führen dazu, dass P3P als „Privacy Enhancing Technology“ (PET) eingestuft wird. Als PET werden Technologien bezeichnet, durch die der Datenschutz verbessert wird. PET ist ein Überbegriff, der neben Policy-Tools wie P3P auch Kommunikationsfunktionen zwischen Nutzer und Server sowie Tools für Verschlüsselung, für Anonymität und Pseudonymität und Filter-Tools umfasst (vgl. [Grimm03]).

Für die öffentliche Verwaltung, die die Zielgruppe für den im Rahmen dieser Arbeit entwickelten P3P-Generator bildet, ist ein weit verbreiteter Einsatz von P3P besonders wünschenswert. Die Veröffentlichung von P3P-Policies kann als zusätzlicher Service für die Nutzer betrachtet werden, der eine Behörde an klar definierte, verbindliche Aussagen bindet und das Webangebot durch die erhöhte Transparenz attraktiver macht. So könnten staatliche Stellen eine Vorbildfunktion einnehmen und andere Webangebote zur Nachahmung anregen.

Für E-Government-Angebote wie z.B. die Abwicklung von Verwaltungsdienstleistungen per Internet spielt das Vertrauen der Nutzer in einen gesetzeskonformen Umgang mit ihren Daten eine große Rolle. Der Einsatz von P3P-Policies, die diesen Vorgaben entsprechen, kann Vertrauen fördern.

3 Anforderungen und Entwurf

Im folgenden Kapitel werden die Anforderungen beschrieben, die der Generator erfüllen soll (vgl. Abschnitt 3.1). Darauf aufbauend stellt Abschnitt 3.2 den Entwurf des Systems dar. Nach einer Übersicht über die wichtigsten Bestandteile des Systems werden die einzelnen Komponenten detaillierter vorgestellt und das Zusammenspiel der Bestandteile anhand von zentralen Abläufen beschrieben.

3.1 Anforderungen an den Generator

Der P3P-Policy-Generator soll die Erstellung von Policies und Policyreferenzdateien erleichtern. Der Generator wurde für Behörden entwickelt, die P3P auf ihren Webseiten einsetzen möchten. Zur Zielgruppe gehören Mitarbeiter aus dem IT-Bereich, aber auch aus anderen Bereichen wie z.B. der Öffentlichkeitsarbeit. Die Kenntnis des P3P-Standards kann hier nicht vorausgesetzt werden, daher sollen die einzelnen P3P-Elemente verständlich in deutscher Sprache beschrieben und abgefragt werden. Sofern Abwärtskompatibilität gewährleistet ist, soll P3P in der derzeit aktuellen Version 1.1 verwendet werden.

Um die Erstellung von Policies möglichst einfach zu gestalten, soll der Generator auf Templates aufbauen. Diese „Muster"-Dateien enthalten ein Gerüst, das der Nutzer mit Hilfe des Generators vervollständigt. In diesen Templates soll es zum einen Elemente geben, die bereits mit Werten belegt sind, zum anderen Elemente, deren Inhalt nicht vorgegeben ist. Die Inhalte der bereits „ausgefüllten" Elemente sollen als Empfehlung dienen, müssen aber vom Nutzer abgeändert werden können. So kann ein Template an die individuellen Bedürfnisse des jeweiligen Webangebots angepaßt werden. Die Empfehlungen, die durch die Templates gegeben werden, sollen sich am deutschen Datenschutzrecht orientieren. Bei Elementen, deren Inhalt nicht vorgegeben ist, soll der Generator die zugehörigen Angaben zwingend abfragen. Die aus der Rechtslage entstehenden Anforderungen und ihre Umsetzung in Templates werden in Kapitel 5 dargestellt.

Die Funktionalität des Generators bei der Erstellung von Policy und Policyreferenzdatei ist vergleichbar mit dem Ausfüllen eines Fragebogens: Der Nutzer soll Fragen zu den Datenschutzpraktiken seines Webangebots beantworten, wobei die einzelnen Punkte möglichst bewußt beantwortet werden sollen. Verhindert werden soll daher, dass der Nutzer den gesamten Fragebogen mit wenigen Mausklicks bestätigt, ohne über die Einzelheiten nachzudenken.

Die Erstellung von Compact Policies zu einer gegebenen P3P-Datei soll mit dem Generator möglich sein. Zudem soll der Generator eine natürlichsprachliche Übersicht über die Belegung der einzelnen P3P-Elemente ausgeben können. Dadurch kann der Nutzer nach Verwendung des Generators protokollieren, welche Angaben er zu den einzelnen Punkten gemacht hat.

Stehen einem Nutzer bereits fertige P3P-Policy-Dateien zur Verfügung, so soll es möglich sein, mit Hilfe des Generators eine Referenzdatei zu erstellen, die auf diese Dateien verweist.

Das Benutzerinterface soll durch ergonomische Gestaltung eine möglichst leichte und intuitive Bedienung des Generators ermöglichen. Soweit vom Aufwand her vertretbar soll auch Barrierefreiheit gewährleistet sein, um auch Nutzern mit Behinderung die Bedienung des Generators zu ermöglichen. Eine ausführlichere Darstellung der daraus resultierenden Anforderungen sowie ihrer Umsetzung findet sich in den Abschnitten 4.3 und 4.4.

Um Weiterentwicklungen des P3P-Standards Rechnung zu tragen, soll der Generator erweiterbar sein. Dies betrifft zum einen die Validierungsmechanismen, die bei Erscheinen einer neuen P3P-Version gegen neue Schemata validieren müssen. Zum anderen soll das Hinzufügen neuer Templates möglich sein, um eine Anpassung an Änderungen der Rechtslage zu ermöglichen.

3.2 Entwurf

Der folgende Abschnitt beschreibt den Entwurf des Systems. Der Generator wurde als eigenständige Java-Applikation entwickelt. Java bietet den Vorteil der Plattformunabhängigkeit, der Generator kann so auf verschiedenen Systemen genutzt werden. Für die graphische Benutzeroberfläche wurde Swing verwendet. Swing ist eine Bibliothek für graphische Oberflächen, die zum Java-Standard gehört. Sie ist Bestandteil der *Java Foundation Classes* (JFC), die u.a. auch ein *Application Programming Interface* (API) zur Unterstützung der Barrierefreiheit enthalten. Durch diese Schnittstelle können Swing-Komponenten barrierefrei zugänglich gemacht werden.

Für die Verarbeitung von P3P-Dateien verwendet der Generator weitere Bibliotheken, die den Umgang mit XML erleichtern. Zum Parsen dieser Dateien wird JDOM in Kombination mit dem Xerces-Parser eingesetzt. Eine vollständige Darstellung aller verwendeten Bibliotheken findet sich in Abschnitt 4.1.

Der Entwurf des Systems ist am „Model-View-Controller“-Paradigma (MVC) orientiert. Diese Bezeichnung umschreibt nach [Metsker02] die Grundidee, dass das interessierende Objekt (*model*) von den Elementen der graphischen Benutzerschnittstelle (GUI), die dieses Objekt anzeigen (*view*) sowie von der Komponente zur Manipulation des Modells (*controller*) getrennt werden kann. Das Ziel, das damit erreicht werden soll, ist die Aufteilung der Verantwortlichkeiten: Klassen und Pakete sollen so klein genug gehalten werden, um die Übersicht nicht zu verlieren und Wartbarkeit zu gewährleisten.

Das Model muss die Controller- und View-Klassen nicht kennen. Diese Trennung der Verantwortlichkeiten unterstützt Java durch die Bereitstellung von Klassen und Schnittstellen zur Implementierung des Beobachter-Entwurfsmusters. Dieses Muster beschreibt einen Mechanismus, der es z.B. einer View-Klasse ermöglicht, die Model-Klasse zu „beobachten“. Ändert sich die Model-Klasse, werden alle Beobachter benachrichtigt.

Der Generator besteht aus vier Paketen, die im Rahmen dieser Arbeit entwickelt wurden: `main`, `view`, `controller` und `model`. Abb. 3.1 zeigt ein UML-Paketdiagramm, das die vier Pakete und ihre wichtigsten Klassen darstellt. Diese Darstellung beschränkt sich auf die zentralen Bestandteile und Beziehungen, um einen übersichtlichen Gesamteindruck vermitteln zu können. Die Abhängigkeiten zwischen den Paketen sind durch gestrichelte Linien dargestellt. So wird angezeigt, dass Klassen eines Pakets auf die in einem anderen Paket enthaltenen Klassen zugreifen.

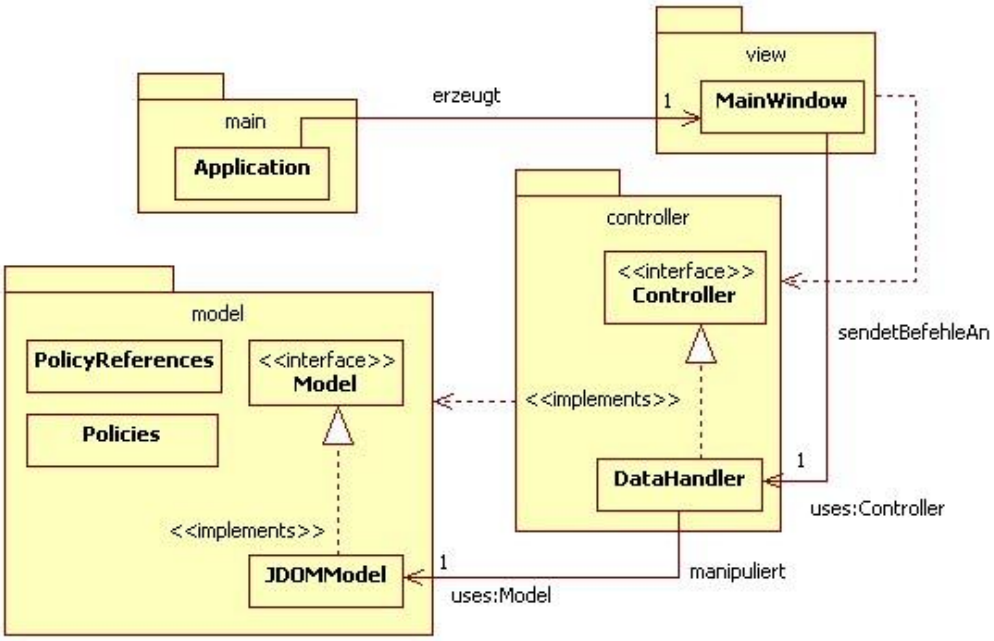

Abbildung 3.1: Paketdiagramm: Die wichtigsten Komponenten des Generators

Die Klasse `Application` des Pakets `main` enthält die `main`-Methode, diese erzeugt die GUI, indem sie eine Instanz der Klasse `MainWindow` erzeugt. Diese Klasse setzt das Singleton-Entwurfsmuster um, das verwendet wird, um sicherzustellen, dass nur eine Instanz einer Klasse erstellt wird und auf diese Instanz global zugegriffen werden kann (vgl. [Metsker02]). Innerhalb des so erzeugten Hauptfensters werden nacheinander verschiedene Views angezeigt, die dazu dienen, Angaben vom Nutzer abzufragen oder den aktuellen Zustand des Modells anzuzeigen. Diese Views sind im Paket `view` enthalten.

Hat der Nutzer innerhalb einer View ein Feld ausgefüllt, wird diese Eingabe mit einem entsprechenden Befehl an den `DataHandler` des `controller`-Pakets

übergeben. Dieser manipuliert das Datenmodell. Zugriffe auf das Datenmodell erfolgen nicht direkt von den Views aus, sondern über den `DataHandler`. Der Zugriff erfolgt über das `Controller`-Interface. Auch der `DataHandler` realisiert das Singleton-Muster, so wird sichergestellt, dass alle View-Klassen dieselbe Instanz der Klasse verwenden.

Das Datenmodell wird von den Klassen im Paket `model` vorgehalten. Dabei werden die Daten des Modells in zwei Schritten verarbeitet. Nach dem Einlesen eines Templates wird die mittels JDOM erzeugte Repräsentation in der Klasse `JDOMModel` vorgehalten. Auch von dieser Repräsentation wird genau eine Instanz benötigt, daher ist `JDOMModel` ebenfalls als Singleton implementiert. Zugriffe auf diese Klasse erfolgen über das Interface `Model`. Gleichzeitig werden die Daten auch in den Klassen `PolicyReferences` und `Policies` abgelegt, die eine zusätzliche Repräsentation des Datenmodells darstellen. Diese zweite Repräsentation wird aufgebaut, um das Auslesen einzelner Werte – z.B. zur Initialisierung von GUI-Elementen – einfacher und intuitiver zu gestalten.

Das Paket `main` enthält neben der Klasse `Application` eine weitere Klasse: Die Klasse `Util` stellt einige Methoden bereit, die von mehreren Paketen verwendet werden. Die Pakete `model`, `view` und `controller` werden in den folgenden Abschnitten detaillierter vorgestellt.

3.2.1 Datenmodell

Die Klassen des Pakets `model` bilden das Datenmodell des Generators. Sie sind in Abb. 3.2 dargestellt. Die Elemente, die in einem P3P-Template enthalten sind, wurden für den Generator durch entsprechende Klassen modelliert. Die Klassen `Policies` und `PolicyReferences` repräsentieren die beiden zentralen Bereiche eines Templates, die Policies und die Policy-Referenzen, die die Geltungsbereiche für die Policies definieren. Die beiden Bereiche können auch in unterschiedlichen Dateien abgelegt werden, die Templates für den Generator enthalten jedoch beide in einer Datei, um die Konfiguration der Policy-Referenzen möglichst einfach zu halten.

Die Klasse `Policies` dient zur Verwaltung der einzelnen Policies, die in Instanzen der Klasse `Policy` abgelegt werden. Sowohl `Policies` als auch `PolicyReferences` haben eine Gültigkeitsdauer, die durch die Klasse `Expiry` modelliert wird.

Die Klasse `PolicyReferences` verwaltet die Geltungsbereiche, die auf Instanzen der Klasse `PolicyRef` abgebildet werden. Mit Hilfe des P3P-Elements HINT können Anbieter auf die Policy-Referenzdateien anderer Anbieter verweisen. Solche Verweise werden in Instanzen der Klasse `Hint` abgelegt. Diese werden ebenfalls von `PolicyReferences` verwaltet. Eine Policy verweist auf die Policy-Referenz, die ihren Geltungsbereich definiert.

Die P3P-Elemente, die eine Policy bzw. eine Policy-Referenz enthält, sind zum Teil durch einfache Datentypen abbildbar, komplexere Elemente werden dagegen durch Klassen modelliert. Diese Klassen tragen in den meisten Fällen den Namen des P3P-Elements, das sie repräsentieren. Eine Ausnahme sind die

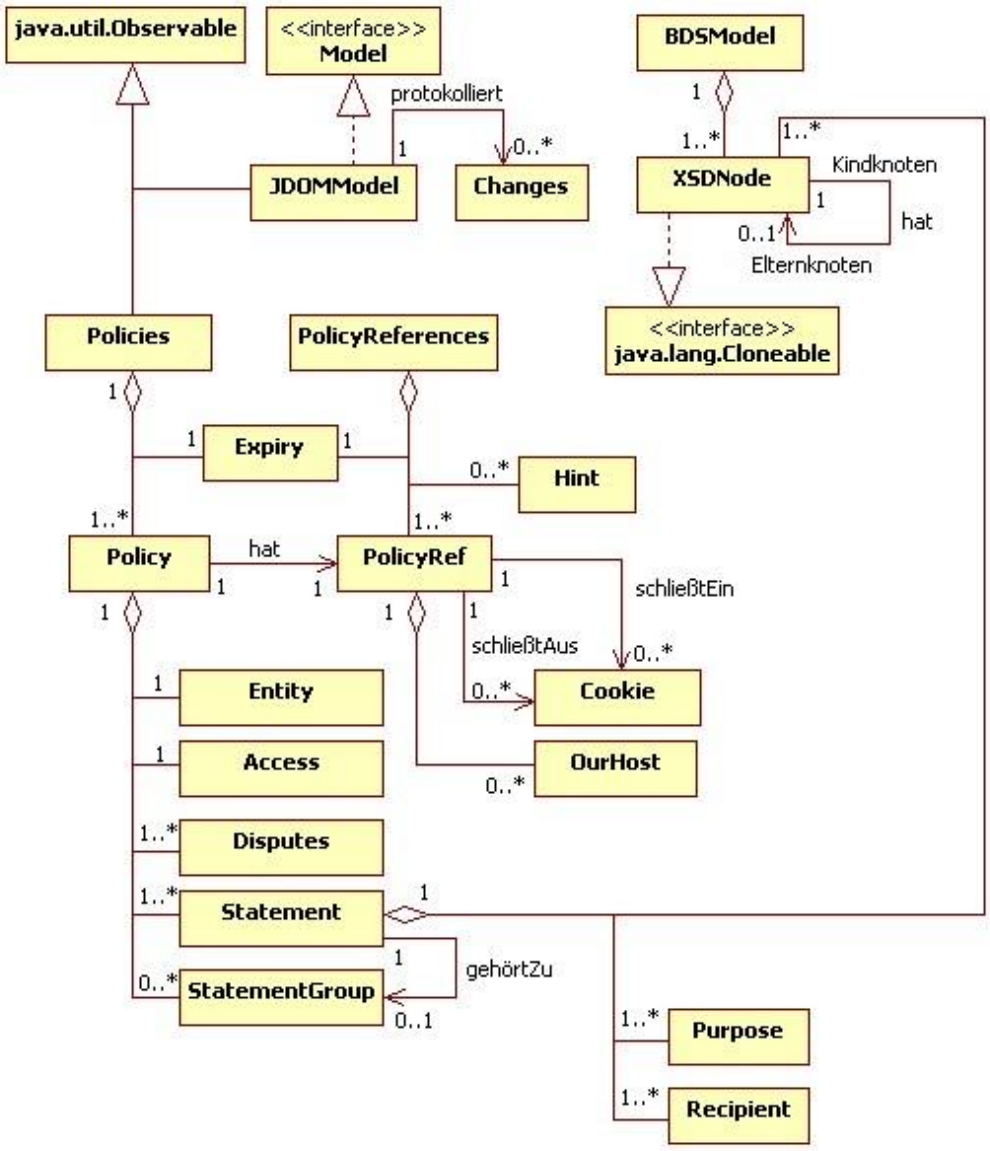

Abbildung 3.2: Klassendiagramm: Das `model`-Paket

P3P-Elemente COOKIE-INCLUDE und COOKIE-EXCLUDE, die zur Definition einer Policy-Referenz eingesetzt werden können. Für beide Elemente werden Instanzen der Klasse `Cookie` verwendet, die entweder in einer *Include*-Liste oder in einer *Exclude*-Liste abgelegt werden.

Das P3P-Element STATEMENT wird durch die Klasse `Statement` modelliert. Auch die Kindelemente dieses Elements werden zum Teil durch Klassen abgebildet. Eines dieser Kindelemente ist `DATA-GROUP`, das Element fasst die `datatype`-Elemente zusammen, mit denen die erhobenen Datenarten definiert werden. Der P3P-Standard definiert die für `datatype`-Elemente wählbaren Datenarten durch das *Base Data Schema*. Der Generator liest dieses Schema ein und generiert daraus ein Modell, das dessen Hierarchie abbildet. Daher liegt dieses Modell in Baumstruktur vor. Die einzelnen Knoten des Baumes sind Instanzen der Klasse `XSDNode`. Die Klasse `BDSModel` erzeugt und speichert den Baum.

Der Generator verfügt parallel über zwei Datenrepräsentationen: Zum einen werden die Daten in den bisher beschriebenen Klassen abgelegt, zum anderen wird beim Einlesen des Templates eine Instanz der Klasse `JDOMModel` erzeugt, die ein JDOM-Objektmodell enthält. Wird das Datenmodell manipuliert, so werden diese Änderungen an beiden Repräsentationen vorgenommen. Dabei wird die Selektion eines Elements im Falle des `JDOMModel` mit XPath bewerkstelligt.

Die Klasse Changes wird verwendet, um eine Änderung zu protokollieren. Die Protokolle werden von der GUI-Komponente ausgewertet, die den aktuellen Zustand des Datenmodells als Baum anzeigt. Diese Komponente ist in der Klasse TreeView des Pakets `view` angesiedelt (siehe Abschnitt 3.2.3).

3.2.2 Manipulation des Datenmodells

Die für den Generator entwickelten Klassen, die für das Laden, Manipulieren, Validieren und Löschen des Datenmodells zuständig sind, befinden sich im Paket `controller`. Sie sind in Abb. 3.3 dargestellt. Das Interface Controller bildet die Schnittstelle zum Datenmodell für Klassen aus dem Paket `view`. Die Implementierung der Schnittstelle erfolgt in der Klasse `DataHandler`. Der DataHandler referenziert weitere Handler-Klassen, die für verschiedene Aufgabenbereiche zuständig sind. Die Klasse `JDOMFileHandler` enhält Methoden zum Parsen der Template-Dateien und zum Schreiben der JDOM-Objektmodelle in Dateien. Die Klasse `PolicyReferenceHandler` ist mit Manipulationen der Policy-Referenzen betraut, während die Klasse `PolicyHandler` Policies und darin enthaltene Elemente manipuliert. Für Manipulationen an Statements und deren Kindelementen verwendet der PolicyHandler die Klasse `StatementHandler`.

Der `PolicyHandler` implementiert die `Observer`-Schnittstelle des Pakets `java.util`. Diese Schnittstelle dient dazu, das Beobachter-Entwurfsmuster umzusetzen, das es einem Objekt (dem Subjekt) ermöglicht, andere interessierte Objekte über Änderungen seines Zustands zu informieren, ohne diese „Beobachter“ kennen zu müssen. Ein Beobachter registriert sich selbst beim Subjekt, das zu diesem Zweck von der Klasse `java.util.Observable` abgeleitet sein muss (vgl. [Metsker02]). Der `PolicyHandler` beobachtet die Instanz der Klasse `Policies` des Datenmodells (siehe Abschnitt 3.2.1). So bleiben `PolicyHandler` und `StatementHandler` bei Änderungen des Datenmodells auf dem Laufenden.

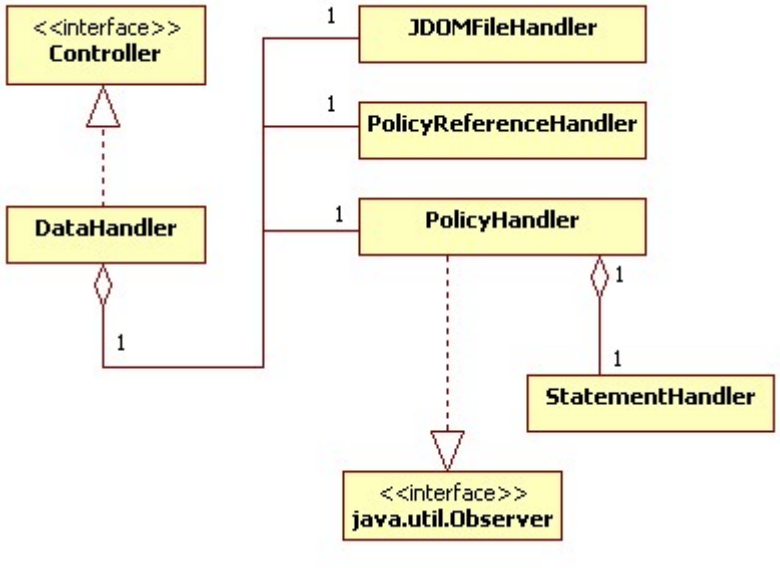

Abbildung 3.3: Klassendiagramm: Das `controller`-Paket

3.2.3 Benutzerinterface

Das Paket `view` enthält die Klassen, die für die graphische Benutzeroberfläche des Generators entwickelt wurden (siehe Abb. 3.4). Das Fenster, das alle Informations- und Eingabekomponenten enthält, wird durch die Klasse `MainWindow` realisiert. Während der Template-Bearbeitung wird eine JTabbedPane angezeigt, die zwei Reiter enthält. Ein Reiter zeigt eine Baumansicht des Datenmodells an, dies ist die Aufgabe der Klasse `TreeView`. Der zweite Reiter zeigt nacheinander verschiedene Eingabefenster an, die Angaben vom Benutzer abfragen. Diese Fenster werden durch Klassen modelliert, die von der Klasse `JPanel` abgeleitet sind. Ein Beispiel hierfür ist `EntityView`, diese Klasse erzeugt eine Eingabemaske, in der der Nutzer das Anbieterkennzeichen definieren kann.

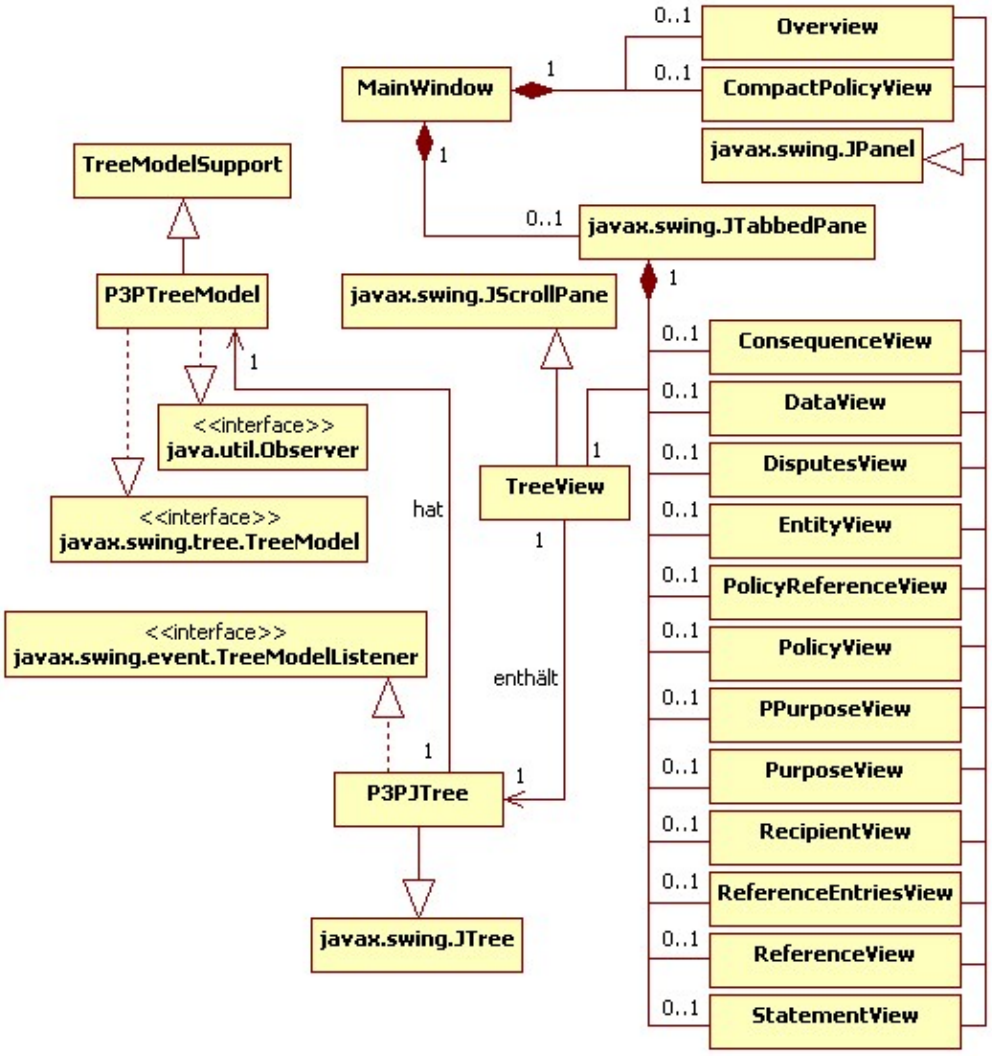

Abbildung 3.4: Klassendiagramm: Das `view`-Paket

Für Funktionalitäten wie die Erstellung einer Compact Policy oder einer natürlichsprachlichen Übersicht über die in der Policy enthaltenen Informationen sind die Klassen `Overview` bzw. `CompactPolicyView` zuständig. Sie sind ebenfalls von `JPanel` abgeleitet, werden aber nicht in einem Reiter, sondern direkt im `MainWindow` angezeigt.

Die Sicht auf das Datenmodell, die von der Klasse `TreeView` erstellt wird, arbeitet nach dem in Abschnitt 3.2.2 beschriebenen Beobachter-Entwurfsmuster. Der `JTree`, der das Datenmodell anzeigt, benötigt als Basis eine Implementation der Schnittstelle `TreeModel`. Dieses Interface wird von der Klasse `P3PTreeModel` realisiert, die sich über den `DataHandler` als Beobachter bei der Klasse `JDOMModel` des Pakets `model` registriert. Das Interface `TreeModel` verlangt die Implementierung von Methoden zur Registrierung von `TreeModelListener`-Instanzen und die Benachrichtigung der registrierten Instanzen bei Änderungen des Modells. Die Implementierung dieser Methoden findet sich in der Klasse `TreeModelSupport`, die von [Armstrong08] übernommen wurde.

3.2.4 Interaktion der Komponenten

Nach der Beschreibung der für den Generator entwickelten Klassen und ihrer Beziehungen zueinander aus Sicht der einzelnen Pakete soll nun die Zusammenarbeit dieser Klassen bei zentralen Abläufen des Generators charakterisiert werden. Diese Abläufe werden durch Kommunikationsdiagramme illustriert. Die Diagramme beschränken sich auf die wichtigsten Kommunikationsmechanismen, um diese übersichtlich darstellen zu können. Startet der Nutzer eine Funktionalität des Generators, so wird je nach Aufgabenstellung eine vollständige P3P-Datei, ein Template oder ein leeres Gerüst für eine P3P-Datei geladen. Dabei wird das Datenmodell erzeugt. Diesen Prozess der Initialisierung beschreibt Abb. 3.5 mit einem Kommunikationsdiagramm.

Nachdem der Nutzer den Generator gestartet hat, erscheint das `MainWindow`. Von dort aus können alle Funktionalitäten gestartet werden. Für das vorliegende Diagramm wurde die Policy-Bearbeitung mit Templates gewählt. Nachdem der Nutzer diese Funktionalität aus dem Menü ausgewählt hat, wird der `DataHandler` damit beauftragt, das Modell zu laden. Übergeben wird ihm der Dateiname des Templates, der in Abb. 3.5 *template* genannt wird, und der Boolesche Wert *true*, der aussagt, dass beim Parsen validiert werden soll. Der `DataHandler` parst daraufhin mit Hilfe des `JDOMFileHandlers` das Template und legt die so erzeugte JDOM-Repräsentation im `JDOMModel` ab. Danach werden `PolicyHandler` und `PolicyReferenceHandler` durch ihre Konstruktoren erzeugt. Diese Konstruktoren sorgen ihrerseits für die Initialisierung von `Policies` und `PolicyReferences`, die parallel zum `JDOMModel` das Datenmodell enthalten. Nachdem so beide Repräsentationen des Datenmodells initialisiert sind, werden innerhalb des `MainWindow` die `PolicyView` und die `TreeView` erzeugt und angezeigt. Diese zeigen Informationen aus dem Template an und greifen daher bei der Belegung ihrer GUI-Komponenten über den `DataHandler` auf das Datenmodell zu.

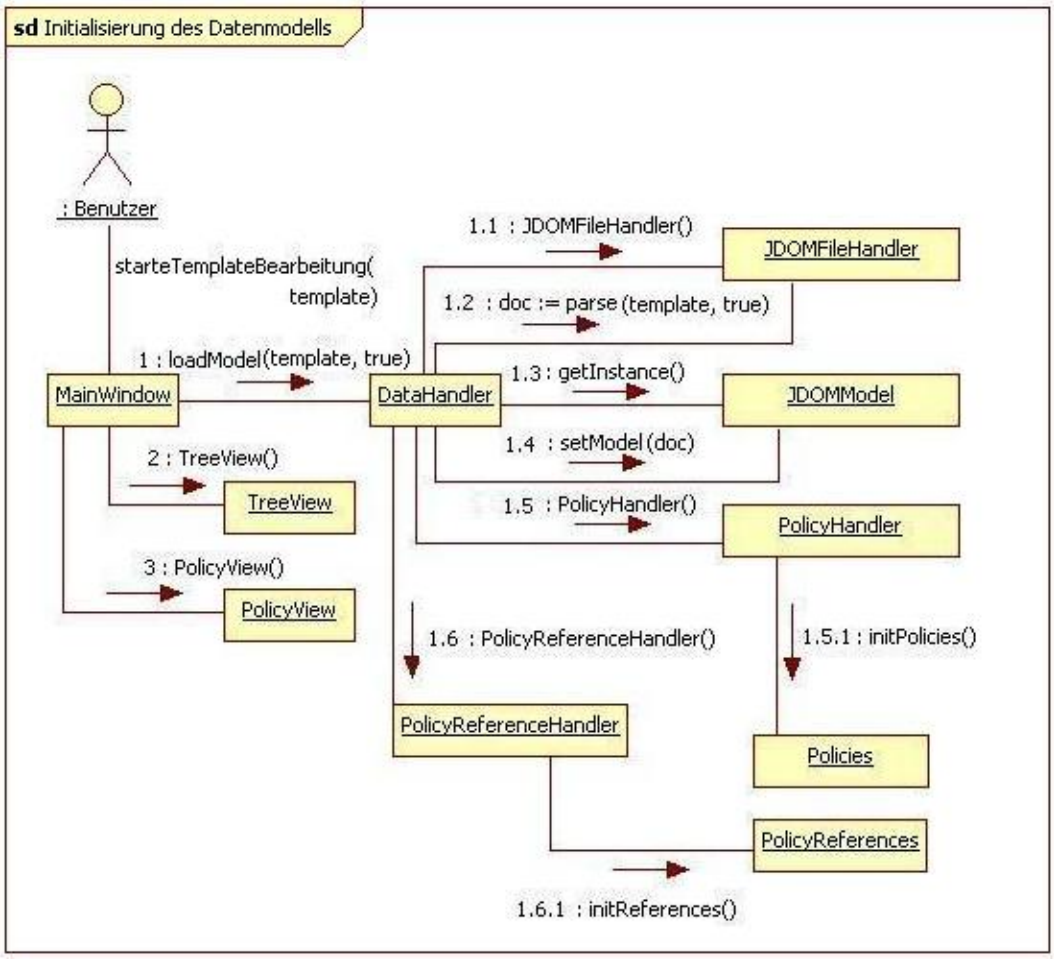

Abbildung 3.5: Kommunikationsdiagramm: Initialisierung des Datenmodells

Während eines Durchlaufs durch die einzelnen Arbeitsschritte der Template-Bearbeitung wählt der Nutzer Belegungen für P3P-Elemente und nimmt so Änderungen am Datenmodell vor. Die Verarbeitung einer Eingabe zeigt das Kommunikationsdiagramm in Abb. 3.6 anhand eines einfachen Beispiels. Das P3P-Element ACCESS, das eine Aussage zum Auskunftsrecht des Nutzers macht, kann eines von sechs vordefinierten Kindelementen enthalten. Diese werden durch einen String charakterisiert und haben weder Attribute noch Kindelemente. Die Auswahl für das ACCESS-Element trifft der Nutzer in der `DisputesView`.

Hat er ein Element selektiert, so benötigt die entsprechende Methode der `DisputesView` den Namen der Policy, die gerade bearbeitet wird (*policyName*), sowie den Bezeichner des gewählten Kindelements von ACCESS (*name*). Die erste Reaktion auf die Auswahl ist die Anzeige eines Hilfetextes in dem dafür vorgesehenen Bereich der GUI. Daraufhin wird das gewählte Element in die beiden Repräsentationen des Datenmodells geschrieben, dies ist Aufgabe des `PolicyHandler`.

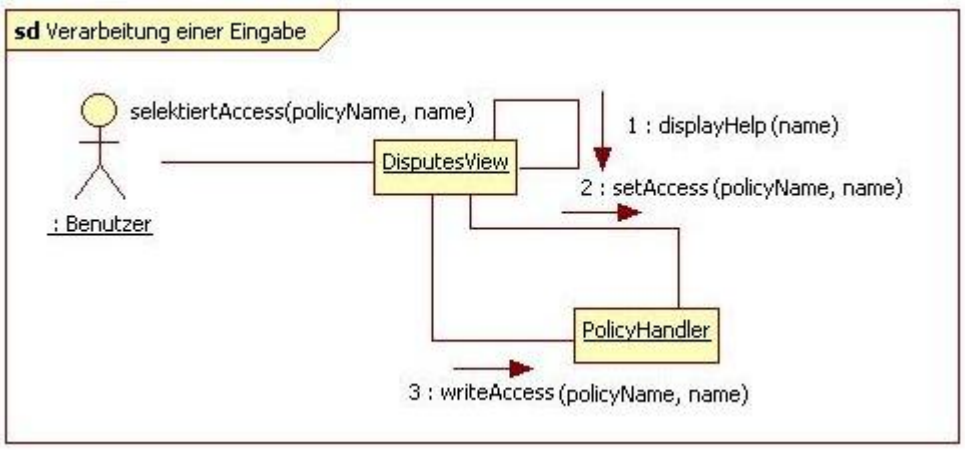

Abbildung 3.6: Kommunikationsdiagramm: Verarbeitung einer Eingabe

Ein wichtiger Ablauf ist auch das Zurücksetzen des Datenmodells. Diese Funktion wird z.B. benötigt, wenn der Nutzer die Bearbeitung abbricht. Damit beim Start der nächsten Funktionalität alle Komponenten neu erzeugt bzw. initialisiert werden, werden die im Datenmodell enthaltenen Daten gelöscht. Abb. 3.7 zeigt die entsprechenden Abläufe. Wie im vorigen Beispiel liegt dem Benutzer die `DisputesView` vor. Nachdem diese den Befehl zum Abbrechen empfangen hat, gibt sie dem `DataHandler` den Auftrag, das Datenmodell zu löschen. Der `DataHandler` ruft daraufhin *reset*-Methoden von `JDOMModel`, `PolicyHandler` und `PolicyReferenceHandler` auf. Anschließend wird das `MainWindow` mit einem leeren Panel belegt.

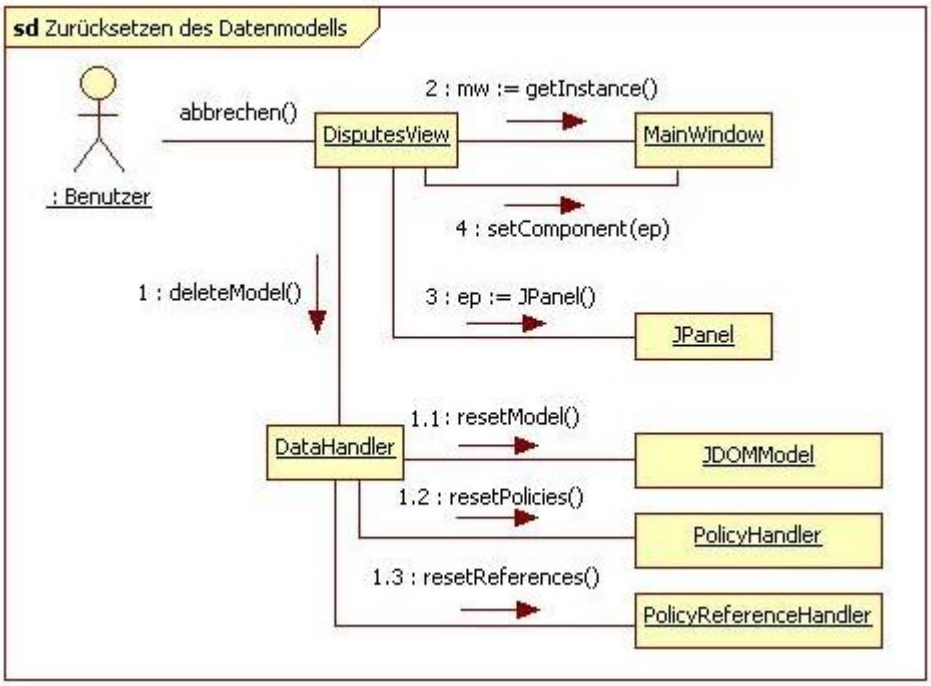

Abbildung 3.7: Kommunikationsdiagramm: Zurücksetzen des Datenmodells

4 Umsetzung

Das folgende Kapitel geht auf die Umsetzung der an den Generator gestellten Anforderungen ein. Dabei kommen verschiedene Software-Bibliotheken zum Einsatz, die in Abschnitt 4.1 beschrieben werden. Abschnitt 4.2 erklärt, für welche Funktionalitäten des Generators welche P3P-Version verwendet wurde. Abschnitt 4.3 erläutert, inwiefern Anforderungen an die softwareergonomische Gestaltung des Programms umgesetzt wurden. Die Funktionalität des Generators soll barrierefrei zugänglich sein, Abschnitt 4.4 diskutiert, inwieweit diese Anforderung realisiert wurde. Den Umgang des Generators mit Fehlern stellt Abschnitt 4.5 vor. Um sicherzustellen, dass die mit dem Generator erstellten Dateien der P3P-Grammatik entsprechen, werden Validierungen gegen die XML Schema-Definition des P3P-Standards vorgenommen. Abschnitt 4.6 stellt die Validierungsmechanismen vor. Abschließend erklärt Abschnitt 4.7, wie weitere Templates in den Generator integriert werden können.

4.1 Verwendete Software

Im Folgenden wird eine Übersicht über die Software gegeben, die für den Generator verwendet wurde (siehe Tabelle 4.1). Der Generator wurde unter dem Betriebssystem Windows XP entwickelt. Programmiert wurde in Java, für die graphische Oberfläche wurde Swing verwendet. Java wurde gewählt, um Plattformunabhängigkeit zu gewährleisten. Einmal übersetzte Java-Programme können auf verschiedenen Betriebssystem-Plattformen ausgeführt werden, ohne für die jeweilige Plattform erneut kompiliert werden zu müssen (vgl. [Krüger07]). Benötigt wird lediglich eine Java-Laufzeitumgebung auf der Zielplattform.

Für die Verwendung von Swing für die GUI-Komponenten spricht, dass diese Bibliothek durch die *Java Accessibility API* die Möglichkeit bietet, GUI-Komponenten barrierefrei zugänglich zu machen. Zudem ermöglicht Swing die Definition eines „Look and Feel“ für die GUI einer Anwendung. Damit sind Erscheinungsbild und Verhalten der einzelnen Komponenten gemeint, die je nach Betriebssystem Unterschiede aufweisen können. Für den Generator wurde das „Metal - Look and Feel“ gewählt, das standardmäßig von der Laufzeitumgebung verwendet wird. Dieses „Look and Feel“ hat den Vorteil, dass es auf allen Plattformen gleich aussieht, es wird auch als *CrossPlatformLookAndFeel* bezeichnet (vgl. [SUN07a]). Die GUI-Klassen des Generators wurden mit Hilfe des NetBeans GUI-Builder entwickelt.

Um P3P- und damit XML-Dateien einzulesen, zu bearbeiten und in Dateien zu schreiben, verwendet der Generator die Bibliothek JDOM. Diese ist – wie alle im Folgenden beschriebenen Bibliotheken – Open Source. Sie bietet ein auf Java basierendes Objektmodell, mit dem sich XML-Dokumente bearbeiten lassen. Das Objektmodell basiert nicht auf Interfaces, sondern auf

konkreten Klassen, die durch einen Konstruktor-Aufruf instanziiert werden können. Für die interne JDOM-Repräsentation eines XML-Dokuments wird auch die Collection-API von Java eingesetzt, insgesamt sind die Klassen auf die Bedürfnisse von Java-Entwicklern zugeschnitten. Durch die Verwendung Java-spezifischer Datenstrukturen ist ein effizienter Zugriff auf die interne Datenstruktur einer XML-Datei möglich (vgl. [Ullenboom07]). JDOM enthält selbst keinen XML-Parser, daher können hier verschiedene Parser zum Einsatz kommen. Für den Generator fiel die Wahl auf Xerces, einen von Apache entwickelten Parser, der sich durch eine gute Performance und eine Reihe von Features, wie z.B. der Unterstützung von XML-Schemata auszeichnet (vgl. [Niedermeier06]).

Mit dieser Unterstützung kann ein XML-Dokument beim Parsen gegen ein oder mehrere XML-Schemata validiert werden. Um die JDOM-Repräsentation des XML-Dokuments oder einzelne Elemente daraus losgelöst vom Parse-Vorgang validieren zu können, werden weitere Bibliotheken benötigt. Zu diesem Zweck bietet die Bibliothek JDOM-Contrib das Package `schema`. Hier ist u.a. die Klasse `Schema` enthalten, die eine solche In-Memory-Validierung ermöglicht. Diese Klasse baut auf JARV (Java API for RELAX Verifiers) auf, und benötigt eine Implemetierung dieser API. Für den Generator dient der Multi-Schema Validator von Sun als Implementierung.

JDOM enthält Unterstützung für die Anwendung von XPath-Ausdrücken auf XML-Dokumente, für diese Funktionalität wird die Bibliothek Jaxen verwendet, die von JDOM mitgeliefert wird.

Name	Version	Quelle
Java SE Development Kit (JDK)	6	http://java.sun.com/javase/downloads/index.jsp
JDOM	1.0	http://www.jdom.org
JDOM-Contrib	1.0	http://www.jdom.org
Xerces-J	2.9.0	http://xerces.apache.org/xerces2-j
Jaxen	1.0-FCS	enthalten in JDOM-Bibliothek, oder unter http://www.jaxen.org
JARV	2004/11/11-Release	http://iso-relax.sourceforge.net/JARV/
MSV	msv-20070407 (nightly build)	https://msv.dev.java.net/

Tabelle 4.1: Verwendete Software

4.2 Verwendete P3P-Versionen

Viele Nutzeragenten unterstützen P3P noch in der Version 1.0. Einige von ihnen verlassen sich bei der Entscheidung, ob ein Cookie angenommen werden soll oder nicht, ausschließlich auf Compact Policies (vgl. Kapitel 7). Aus diesem Grund erzeugt der im Rahmen dieser Arbeit entwickelte Generator Compact Policies, die die Syntax für „aggregierte“ Statements gemäß P3P 1.0 verwenden (siehe Abschnitt 2.5). P3P 1.1 erlaubt die Verwendung dieser Syntax weiterhin (vgl. [Wenning06]). Damit soll sichergestellt werden, dass alle Nutzeragenten diese Compact Policies verarbeiten können.

Die Erstellung von Compact-Policies ist die einzige Funktionalität des Generators, die P3P in der Version 1.0 verwendet. Für Policies und Policy-Referenzen wird die abwärtskompatible Version 1.1 des Standards verwendet.

Abwärtskompatibilität zu Version 1.0 wird dadurch erreicht, dass neu eingeführte Elemente innerhalb des EXTENSION-Elements plaziert werden. Die meisten P3P-Elemente, die in Version 1.0 definiert wurden, können dieses Element als Kindelement enthalten. In einem EXTENSION-Element darf beliebiger Inhalt eingefügt werden, so dass der Standard flexibel erweitert werden kann (vgl. [Wenning06]).

4.3 Softwareergonomische Aspekte

Eine Anforderung an den Generator ist die benutzerfreundliche Gestaltung, das Programm soll auch für Laien leicht bedienbar sein (siehe Abschnitt 3.1). Die Software-Ergonomie befasst sich mit der benutzer- und aufgabengerechten Gestaltung von Benutzerschnittstellen und entwickelt entsprechende Richtlinien (vgl. [Herczeg05]).

Als Grundlage der folgenden Betrachtungen werden die sogenannten Dialogprinzipien herangezogen, die in der Norm DIN EN ISO 9241-110 mit dem Titel „Grundsätze der Dialoggestaltung“ definiert sind. Die Norm beschreibt sieben Gestaltungsprinzipien für interaktive Systeme aller Art (vgl. [Geis07b]). Im folgenden werden die Prinzipien der Norm vorgestellt. Dabei wird untersucht, inwieweit diese Vorgaben im Generator umgesetzt werden konnten. Die hierfür verwendete Literatur beschreibt noch die bis April 2006 geltende Norm DIN EN ISO 9241-10, die sich nach einer Überarbeitung statt in Teil 10 nun in Teil 110 dieser Norm befindet. In die neue Version wurden die Dialogprinzipien übernommen, ihre Beschreibung wurde weiter präzisiert (vgl. [Geis07b]).

Aufgabenangemessenheit

Dieses Prinzip formuliert nach [Herczeg05] das Ziel, den Benutzer bei der effektiven und effizienten Erledigung seiner Aufgabe zu unterstützen. Das System soll daher keine unnötigen Bearbeitungsschritte erforderlich machen und die Ein- und Ausgabe an die Belange der jeweiligen Aufgabe und des Benutzers anpassen. Ein Dialog soll so gestaltet sein, dass die Kenntnisse und Fähigkeiten des Nutzers berücksichtigt werden.

Um diese Ziele zu erreichen, soll das System dem Benutzer zur Bearbeitung seiner Aufgaben (automatisierte) Hilfestellungen anbieten. Dies erfüllt der Generator, indem er die XML-Verarbeitung übernimmt und Templates anbietet, die Empfehlungen zur Belegung von Elementen geben. Diese empfohlenen Werte sind in der Benutzeroberfläche voreingestellt, können aber vom Nutzer geändert werden. Auch die Validierung der einzelnen Elemente bzw. des Gesamtdokuments nimmt der Generator vor. Den Bedürfnissen des Nutzers wird dadurch Rechnung getragen, dass Kenntnisse des P3P-Vokabulars nicht vorausgesetzt werden, sondern die einzelnen Dialoge selbstbeschreibend gestaltet wurden.

Die Funktionalität des Generators erstreckt sich über alle wesentlichen Bestandteile des P3P-Standards. Es können Policy-Referenzen und Policies sowie Compact Policies erstellt werden. Somit wird der Generator seiner Aufgabe gerecht.

Selbstbeschreibungsfähigkeit

Das Prinzip der Selbstbeschreibungsfähigkeit verlangt, dass jeder Dialogschritt entweder unmittelbar verständlich ist oder auf Anfrage des Benutzers erläutert wird. Zu diesem Zweck soll das System Rückmeldungen bei Bedienfehlern sowie Erläuterungen anbieten, die möglichst gut auf die Situation bezogen sind, in der der Nutzer sie benötigt. Dadurch soll vermieden werden, dass häufig Benutzerhandbücher oder weitere externe Informationsquellen konsultiert werden müssen (vgl. [Herczeg05]).

Die Terminologie des Generators ist daher – soweit möglich – an die Begrifflichkeiten des deutschen Datenschutzrechts angepasst. Zur Beschreibung der einzelnen P3P-Elemente bietet der Generator Hilfestellungen: Manchen Elementen ist ein in die jeweilige View integrierter Hilfe-Bereich zugeordnet, der Hilfestellungen zu den jeweils ausgewählten Elementen anzeigt. Dies ist aus Platzgründen nicht für alle Elemente praktikabel, in diesen Fällen setzt der Generator Hilfe-Buttons ein, die ein Popup-Fenster mit Erläuterungen öffnen. Auf die Konsequenzen der Wahl bestimmter Elemente macht der Generator den Nutzer durch Warnungen oder Hinweise aufmerksam.

Steuerbarkeit

Ein steuerbares System gibt dem Benutzer nach [Herczeg05] die Möglichkeit, den Start des Dialogablaufs zu bestimmen und bis zum Ende seine Richtung und Geschwindigkeit zu beeinflussen. Die Kontrolle über den Dialogablauf soll generell beim Benutzer liegen. Das System soll also die Geschwindigkeit nicht vorgeben, sondern dem Nutzer soviel Zeit lassen, wie er benötigt, um seine Aufgabe zu erledigen. Dies ist beim Generator sichergestellt, da der Nutzer sämtliche Funktionalitäten selbst über die GUI startet und der Übergang von einem Arbeitsschritt zum nächsten durch GUI-Ereignisse gesteuert wird, die der Nutzer auslöst.

Um den Arbeitsablauf kontrollieren und weiter planen zu können, sollen entsprechende Informationen ohne Verlust des momentanen Arbeitskontextes

einsehbar sein. Daher sollen mehrere unterbrechbare und wiederaufsetzbare Teildialoge möglich sein. Das Ausmaß der Autonomie bei der Steuerung des Dialogablaufs sollte nach [Herczeg05] den Erfordernissen der Arbeitsaufgabe angepaßt werden. Der Generator gibt die Reihenfolge der Dialoge innerhalb eines Durchlaufs durch eine Policy vor, hier ist der Nutzer also nur innerhalb der einzelnen Views autonom, nicht aber in der Wahl der Abfolge. Die vorgegebene Reihenfolge führt von allgemeinen zu spezielleren Angaben und ist ausgerichtet an der Reihenfolge der Elemente in einer P3P-Policy. Da der Nutzer nicht über Vorkenntnisse zu P3P verfügt, verhindern diese Vorgaben möglicherweise unnötige Verwirrung, die bei einer willkürlich wählbaren Abfolge entstehen könnte. Ohne Führung durch den Dialog könnte beispielsweise der Eindruck entstehen, dass einzelnen Statements Geltungsbereiche zugewiesen werden können, wenn zuerst ein Statement, dann ein Geltungsbereich bearbeitet wird. Ein Geltungsbereich gilt jedoch für eine Policy, was dadurch betont wird, dass dieses Element als erstes abgefragt wird. Zur Kontrolle des Arbeitsablaufs kann die „XML-View“ dienen, die eine baumartige Darstellung des gesamten Modells anzeigt.

Ein Wiederaufsetzen auf einem Zwischenstand ist im Generator im Sinne von Erweiterungen möglich: Ein Template kann bearbeitet und gespeichert werden. Sollen weitere Policies bzw. Statements hinzugefügt oder gewählte Elemente geändert werden, kann die P3P-Datei im Generator geöffnet und weiter bearbeitet werden.

Für das Kriterium der Steuerbarkeit wird auch die Rücknehmbarkeit von Aktionen gefordert, wenigstens der letzte Dialogschritt sollte rückgängig gemacht werden können, soweit seine Folgen reversibel sind und es für die Arbeitsaufgabe zweckmäßig ist (vgl. [Herczeg05]). Der Generator erlaubt es innerhalb einer View jederzeit, einen eingegebenen Wert durch einen anderen zu überschreiben oder die getroffene Auswahl abzuändern. Zudem verfügt jede View über einen „Zurück“-Button, mit dem in die vorhergehende View gewechselt werden kann, so dass es möglich ist, beliebig viele Arbeitsschritte zurückzuspringen und dabei Änderungen vorzunehmen.

Im Hinblick auf Eingabe-/Ausgabegeräte wird für die Steuerbarkeit gefordert, dass der Benutzer entscheiden kann, welche er nutzen will (vgl. [Herczeg05]). Der Generator setzt dies um, indem der Nutzer zwischen der Eingabe per Maus oder Tastatur wählen kann (vgl. Abschnitt 4.4). Empfohlen wird auch, dem Nutzer Einfluß auf die Art der Anzeige von Ein- und Ausgabedaten zu geben. Dies wurde aufgrund der begrenzten Zeit im Generator nicht realisiert.

Erwartungskonformität

Das Kriterium der Erwartungskonformität ist erfüllt, wenn die Dialoge eines Systems konsistent sind, d.h. wenn ihr Verhalten und die Art der Informationsdarstellung innerhalb des Systems einheitlich sind. Es wird empfohlen, die Dialoge für ähnliche Arbeitsaufgaben auch ähnlich zu gestalten, um den Erwartungen des Nutzers zu entsprechen und die Bildung eines mentalen Modells vom System zu fördern. Ein erwartungskonformer Dialog entspricht auch den Erfahrungen des Nutzers sowie den allgemein anerkannten Konventionen. Der

Stand der Bearbeitung sollte möglichst unmittelbar aktualisiert werden und stets an der gleichen Stelle angegeben werden (vgl. [Herczeg05]).

Die Darstellung von Informationen innerhalb des Generators ist an gängigen Konventionen orientiert. Es werden z.B. Checkboxen verwendet, wenn mehrere Alternativen wählbar sind, RadioButtons werden eingesetzt, wenn nur ein Element wählbar ist. Hilfebuttons sind in allen Fenstern gleich gestaltet, sie zeigen immer ein Fragezeichen an. Auch die Felder, die Erläuterungen enthalten, sind auf die gleiche Weise beschriftet und umrahmt. Die Navigationselemente, mit denen der Nutzer sich von einem Dialog zum nächsten bewegt, sind durchgängig in der rechten unteren Hälfte der Views angebracht.

Sofern Bestandteile des P3P-Standards von mehreren Funktionalitäten des Generators abgefragt werden, verwendet der Generator dafür gleiche oder sehr ähnlich gestaltete Eingabefenster. Dies ist beispielsweise bei der Definition von Geltungsbereichen der Fall.

Während der Bearbeitung einer Policy ist der aktuelle Zustand in der „XML-View“ sichtbar. Der Generator enthält zwei Reiter, so dass der Nutzer parallel zum jeweiligen Eingabefenster diese Ansicht des Datenmodells einsehen kann.

Fehlerrobustheit

Fehlerrobustheit ist nach [Herczeg05] dann gewährleistet, wenn der Nutzer das beabsichtigte Arbeitsergebnis trotz fehlerhafter Eingaben mit keinem oder möglichst wenig Korrekturaufwand erreichen kann. Das soll erreicht werden, indem das System den Nutzer auf Eingabefehler aufmerksam macht und ihm dabei hilft, diese zu vermeiden. Dadurch soll verhindert werden, dass eine fehlerhafte Eingabe zu undefinierten Systemzuständen oder Systemabbrüchen führen kann.

Der Generator überprüft in manchen Fällen die Eingaben des Nutzers, Beispiele hierfür sind die Felder, in die der Nutzer eine Postleitzahl oder eine Mail-Adresse eingeben kann. Hier kann überprüft werden, ob für die Postleitzahl fünf Zahlen eingegeben wurden, oder ob die Mail-Adresse ein „@“ enthält. Falls eine solche Eingabe fehlerhaft ist, wird der Fokus des Textfeldes nicht weitergegeben, die Eingabe wird rot eingefärbt und es wird eine Warnung im Terminalfenster ausgegeben. So wird verhindert, dass eine offensichtlich falsche Eingabe ins Modell geschrieben wird, gleichzeitig wird der Nutzer auf den Fehler aufmerksam, ohne bei jedem Tippfehler ein Popup-Fenster schließen zu müssen. Bei einigen Textfeldern des Generators findet dagegen keine Überprüfung des Inhalts statt, dies gilt z.B. für Felder, in die der Nutzer eine natürlichsprachliche Beschreibung eines Elements eingeben soll.

Ist ein Dialog des Generators unvollständig ausgefüllt, oder erfordert die Auswahl des Nutzers weitere Angaben, so wird der Nutzer durch Fehlermeldungen in Form von Popup-Dialogen auf die fehlenden Angaben aufmerksam gemacht. Diese Fehlermeldungen erscheinen, wenn der Nutzer durch Anklicken des „Weiter“-Buttons versucht, die Bearbeitung der View abzuschließen und zum nächsten Dialog zu navigieren.

Individualisierbarkeit

Das Kriterium der Individualisierbarkeit zielt auf Anpassungsmöglichkeiten des Systems an die Erfordernisse der Arbeitsaufgabe sowie Vorlieben und Fähigkeiten des Benutzers ab. Ein System, das diesem Kriterium genügt, ist anpassbar an Sprache und kulturelle Eigenheiten des Benutzers, an individuelle Vorkenntnisse auf dem Gebiet der Arbeitsaufgabe sowie an Wahrnehmungsvermögen, sensomotorische und geistige Fähigkeiten (vgl. [Herczeg05]).

Diese Anforderungen vollständig umzusetzen, war im zeitlich begrenzten Rahmen dieser Arbeit nicht möglich. Kulturelle Eigenheiten und Sprache sind durch die Zielgruppe der Mitarbeiter deutscher Behörden eingegrenzt, eine diesbezügliche Anpassung ist daher nicht erforderlich. Eine Individualisierung hinsichtlich des Kenntnisstandes des Benutzers wird von der Norm insofern empfohlen, als es z.B. möglich sein soll, den Umfang von Erläuterungen anzupassen oder den Nutzer sein eigenes Vokabular für das System definieren zu lassen. Dies wurde im Generator nicht realisiert. Das Angebot von Hilfen und Erläuterungen wurde im Generator allerdings so gestaltet, dass sie einen erfahrenen Nutzer nicht stören. Daher bringt eine Individualisierung im Hinblick auf den Umfang der Hilfen keinen Vorteil.

Die unterschiedlichen Voraussetzungen der Benutzer im Bezug auf das Wahrnehmungsvermögen und sensomotorische Fähigkeiten wurden insofern berücksichtigt, als Richtlinien zur Gestaltung zugänglicher Software beachtet wurden (siehe Abschnitt 4.4).

Lernförderlichkeit

Die Erlernbarkeit eines interaktiven Systems wird nach [Herczeg05] mit dem Kriterium der Lernförderlichkeit untersucht. Dieses Kriterium ist erfüllt, wenn ein Dialog den Nutzer beim Erlernen des Systems unterstützt und anleitet. Zu diesem Zweck sollen dem Nutzer Regeln und zugrundeliegende Konzepte zugänglich gemacht werden, um ihm die Orientierung zu erleichtern. Ein Beispiel hierfür ist eine konsistente Wahl von Kurzwahltastenkombinationen: Wenn möglich wird dafür der erste Buchstabe eines Menübezeichners verwendet, generell werden alle Kurzwahltasten auf die gleiche Art markiert, z.B. durch Unterstreichung. Dies ist ein Punkt, der vom Generator umgesetzt wird. Forderungen der Norm nach Anpassungsmöglichkeiten an die Häufigkeit der Nutzung von Kommandos oder nach interaktivem Online-Unterricht, der den Nutzer ermutigt, mit dem System zu experimentieren, wurden aufgrund der begrenzten Bearbeitungszeit nicht realisiert. Das Konzept des „Learning by doing“, das durch das Prinzip der Lernförderlichkeit unterstützt werden soll, wird im Generator durch die unter dem Punkt „Selbstbeschreibungsfähigkeit“ genannten Elemente umgesetzt. Eine weitere Lernhilfe ist die in dieser Arbeit enthaltene Bedienungsanleitung.

4.4 Barrierefreiheit

Eine Anforderung an den Generator besteht darin, die GUI-Komponenten für Menschen mit Behinderung zugänglich zu gestalten, sofern dies vom Aufwand her vertretbar ist (siehe Abschnitt 3.1). Um einen barrierefreien Zugang zum Generator zu ermöglichen, wurden die von Sun empfohlenen Richtlinien zur Sicherstellung grundlegender Zugänglichkeit zu einer Java-Applikation befolgt (vgl. [SUN08] und [SUN07b]). Hier können weitere Maßnahmen ergriffen werden, um die Zugänglichkeit weiter zu verbessern, diese Arbeit beschränkt sich aus zeitlichen Gründen auf die Umsetzung der genannten grundlegenden Richtlinien, die im folgenden beschrieben werden.

Die „Java Accessibility API“ ist Teil der Java Foundation Classes (JFC), die auch Swing beinhalten. Die zu dieser API gehörigen Klassen und Schnittstellen befinden sich im Paket *javax.accessibility*. Technische Hilfsmittel für Menschen mit Behinderung können Informationen von Komponenten beziehen, die die Accessibility API verwenden. Dies ist bei fast allen Standard-Swing-Komponenten der Fall.

In [SUN08] wird empfohlen, für alle Komponenten einer graphischen Oberfläche einen *AccessibleName* zu vergeben. Dieser Name wird bei beschrifteten Komponenten wie Labels, Buttons etc. automatisch gesetzt und mit der Beschriftung der Komponente belegt. Allen anderen Komponenten sollte ein solcher Name zugewiesen werden, damit technische Hilfsmittel dem Nutzer den Namen der Komponente, die gerade den Focus hat, mitteilen können.

Möglicherweise ist ein Name nicht aussagekräftig genug. Sofern der Kontext und die genaue Bedeutung einer Komponente durch den *AccessibleName* nicht deutlich genug beschrieben werden, sollte die Komponente mit einer *AccessibleDescription* versehen werden, die präzisere Erläuterungen enthält.

Für Labels, die eine Komponente beschreiben, sollte die Zusammengehörigkeit der beiden Komponenten definiert werden, indem die *labelFor*-Eigenschaft des Labels auf die dazugehörige Komponente verweist. Dies ist z.B. bei Textfeldern und deren Beschriftungs-Label der Fall. So können technische Hilfsmittel das zugehörige Textfeld identifizieren.

Da viele Nutzer mit Behinderung nicht mit der Maus, sondern ausschließlich mit der Tastatur arbeiten können, sollte eine Bedienung des Programms mit der Tastatur ermöglicht werden. Hilfreich ist daher eine logische Navigationsabfolge per Tab. Diese sollte sich an der gewohnten Leserichtung des Nutzers orientieren, für den Generator bewegt sich die Abfolge also von links nach rechts und von oben nach unten. Um die Verwendung der Tastatur zu ermöglichen, sind zudem allen wichtigen Funktionen Mnemonics zuzuordnen. Darunter versteht man die Markierung eines Buchstabens in der Beschriftung einer Komponente. Wird die „Alt“-Taste in Kombination mit dem markierten Buchstaben betätigt, wandert der Focus auf diese Komponente, was dem Nutzer die Möglichkeit gibt, ohne Umwege zu den gewünschten Komponenten zu navigieren. Ein Menüpunkt kann mit einem *Accelerator*, d.h. einer Tastenkombination zur direkten Aktivierung versehen werden (vgl. [SUN07b]). Dieser

Mechanismus funktioniert auch, wenn das entsprechende Menü oder Untermenü nicht ausgeklappt und damit nicht sichtbar ist.

Für Instanzen der Klasse `ImageIcon` sollten textuelle Beschreibungen bereitgestellt werden (vgl. [SUN07b]). Im Falle des Generators wird `ImageIcon` für einige Buttons eingesetzt. Diese wurden mit einer Beschreibung versehen.

Logisch zusammengehörende Komponenten sollten innerhalb eines Containers platziert werden. Dafür bieten sich z.B. Panels an. Diese sollten – wie alle anderen Komponenten – mit einem Namen und bei Bedarf mit einer Beschreibung versehen werden.

Die Eigenschaften von Schriften sowie Farben sollten nach [SUN08] nicht „hart“ codiert werden. Dies wurde im Generator berücksichtigt, für Farben und Schriften wurden die Standardeinstellungen beibehalten. Der Generator enthält jedoch keine Funktion zur Einstellung von Farben und Schriften.

Da die GUI-Komponenten des Generators mit Hilfe der NetBeans-Entwicklungsumgebung entwickelt wurden, wurde zur Unterstützung und Überprüfung dieser Zugänglichkeitseigenschaften das *a11y Checker Module* von NetBeans verwendet. Dieses Plugin vereinfacht das Editieren der Komponenten im Hinblick auf die hier aufgeführten Empfehlungen und unterstützt den Nutzer u.a. bei der Definition einer geeigneten Tab-Navigationsabfolge. Es gibt einige weitere Tools, um die Zugänglichkeit eines Programms zu testen, diese könnten zur Weiterentwicklung des Generators verwendet werden (vgl. [NBA08] und [SUN08]).

4.5 Fehlerbehandlung

Die Programmierung eines Systems völlig fehlerfrei zu gestalten, ist nicht möglich. Daher muss eine Applikation über Mechanismen verfügen, die eine Reaktion auf Fehler ermöglichen und die Fehlersuche erleichtern. Da der Generator Template-Dateien einliest und bearbeitet, können Fehler in der Eingabedatei neben Bugs im System eine weitere Problemquelle sein. Zum Umgang des Generators mit Fehlern gehört die Erstellung einer Logdatei sowie die Reaktion auf Exceptions.

Log-Datei

Der Generator verwendet zur Erstellung einer Logdatei die Klasse `Logger` des Pakets `java.util.logging`. Eine Instanz dieser Klasse wird beim Start des Programms in der Klasse `Application` des Generators erstellt. Auf diese Instanz können alle Generator-Klassen zugreifen. Die Meldungen des Loggers werden in die Datei *log.txt* im Hauptverzeichnis des Generators geschrieben und enthalten jeweils folgende Informationen:

- den Zeitpunkt, an dem die Meldung protokolliert wurde, in Form von Datum und Uhrzeit
- die für die Meldung verantwortliche Methode sowie die Klasse, in der die Methode sich befindet

- den Log-Level, der etwas über den Charakter der Meldung aussagt (z.B. „Info“ oder „Warning“)
- die Meldung

Der Logger wird verwendet, um die Ausführung wichtiger Funktionen zu dokumentieren, indem Erfolgsmeldungen oder Fehlerbeschreibungen in die Logdatei geschrieben werden.

Exceptions

In Java können bei Laufzeitfehlern oder vom Entwickler definierten Bedingungen Ausnahmen – sog. Exceptions – ausgelöst werden. Das Auftreten einer Exception kann im Programm behandelt werden, indem eine Reaktion auf die Exception definiert wird. Wird eine Exception nicht behandelt, führt dies zum Abbruch des Programms (vgl. [Krüger07]).

Der Generator behandelt Exceptions durch Protokollierung des Fehlers in der Logdatei sowie eine Ausgabe der zur Exception gehörenden Details auf der Konsole mittels `exception.printStackTrace()`. Wo es möglich ist, wird durch das Auftreten einer Exception auch der Rückgabewert der Methode, in der diese auftrat, verändert. So kann die aufrufende Stelle den Rückgabewert interpretieren und den Nutzer über Fehler informieren. Tritt beispielsweise beim Laden eines Dokuments wegen Fehlern in der Eingabedatei eine Exception auf, so wird dem Nutzer eine Fehlermeldung angezeigt, die auf die detaillierteren Angaben in der Logdatei verweist.

4.6 Validierung

Der Generator validiert Templates oder vom Nutzer erstellte P3P-Dateien, die durch die „Öffnen“-Funktion geladen werden, beim Einlesen. Damit soll sichergestellt werden, dass die Bearbeitung auf einer fehlerfreien Basis aufsetzt. Die Grammatik von P3P-Dokumenten ist mit XML Schema definiert. Zur Validierung benötigt der Generator folgende Dateien:

- P3Pv1.xsd: Definition der P3P-Version 1.0
- P3Pv11.xsd: Definition der P3P-Version 1.1
- P3Pv11BDS.xsd: : Definition des P3P *Base Data Schema* für P3P Version 1.1
- xml.xsd: Schema, das den Namespace `http://www.w3.org/XML/1998/namespace` mit dem Präfix „xml“ beschreibt
- XMLSchema.dtd: DTD für XML Schemas (Part 1: Structures)
- datatypes.dtd: DTD für XML Schemas (Part 2: Datatypes)

Die aktuelle Version 1.1 des P3P-Standards befindet sich in P3Pv11.xsd, in dieser Datei werden Version 1.0, das *Base Data Schema* sowie der „xml“-Namensraum importiert. Das *Base Data Schema* bindet die DTD für XML Schema-Dokumente ein, diese enthält eine Referenz auf die DTD für Datentyp-Definitionen.

Die Import-Anweisungen wurden für den Generator leicht abgeändert, um zu ermöglichen, dass eine Validierung ohne Internet-Zugang stattfinden kann. Zu diesem Zweck wurde dem Attribut `schemaLocation` des Elements `import` die lokale Kopie des Schema-Dokuments zugewiesen. So wurde z.B. die Import-Anweisung

```
<import namespace="http://www.w3.org/2002/01/P3Pv1"
   schemaLocation="http://www.w3.org/2002/01/P3Pv1"/>
```

abgeändert in folgende Anweisung:

```
<import namespace="http://www.w3.org/2002/01/P3Pv1"
   schemaLocation="P3Pv1.xsd"/>
```

Ebenfalls zugunsten einer Offline-Validierung wurde in P3Pv11BDS.xsd die Anweisung zur Einbindung einer DTD von der ursprünglichen Form:

```
<!DOCTYPE schema PUBLIC "-//W3C//DTD XMLSchema 200102//EN"
   "http://www.w3.org/2001/XMLSchema.dtd">
```

in die folgende Form umgewandelt:

```
<!DOCTYPE schema SYSTEM "XMLSchema.dtd">
```

Die geänderten Kopien der benötigten Schemata befinden sich im Hauptverzeichnis des Generators. Die Validierung beim Einlesen eines Dokuments findet in der Methode `parse(String fileName, boolean validate)` der Klasse `JDOMFileHandler` statt. Ein bereits eingelesenes Dokument oder einzelne Elemente dieses Dokuments können ebenfalls validiert werden. Dies geschieht, wenn der Nutzer ein Element durch seine Eingaben geändert hat. Die hierfür benötigten Methoden sind in der Klasse `DataHandler` enthalten. Beide Klassen befinden sich im Paket `controller`. Neue P3P-Versionen müssten an diesen Stellen eingebunden werden.

4.7 Integration weiterer Templates

Der Generator kann an veränderte Gegebenheiten z.B. im Bereich der Gesetzgebung angepaßt werden, indem neue Templates hinzugefügt oder vorhandene Templates geändert werden. Damit der Generator ein neues Template zur Bearbeitung anbietet, muss in der Konfigurationsdatei *templates.properties*, die im Hauptverzeichnis des Generators liegt, ein Bezeichner für das Template und der relative Pfad zur Template-Datei vom Hauptverzeichnis des Generators aus angegeben werden. Standardmäßig sind die Template-Dateien unter `data/templates` abgelegt. Für ein neues Template, das diesem Verzeichnis hinzugefügt wurde, müßte *templates.properties* z.B. um folgenden Eintrag ergänzt werden:

```
meinTemplate = data\\templates\\meinTemplate.xml
```

Die Pfad-Separatoren „\\“ sowie „/“ werden vom Generator an die Gegebenheiten des Betriebssystems angepasst.

5 Template-Policies für öffentliche Stellen in Rheinland-Pfalz

In diesem Kapitel wird zunächst P3P aus rechtlicher Sicht beschrieben (siehe Abschnitt 5.1). Auf dieser Grundlage basieren die Templates, die mit Hilfe des P3P-Generators vervollständigt werden können. Die Templates stellt Abschnitt 5.2 vor.

5.1 P3P aus datenschutzrechtlicher Sicht

Im folgenden wird P3P aus datenschutzrechtlicher Perspektive beleuchtet. Abschnitt 5.1.1 gibt eine kurze Übersicht über die Gesetzgebung im Bereich Datenschutz in Deutschland. Abschnitt 5.1.2 geht auf die Frage ein, inwieweit P3P konform zum Telemediengesetz ist.

5.1.1 Datenschutzrecht bei Telemedien

Das Datenschutzrecht befaßt sich nach [Wohlgemuth05] mit den Voraussetzungen und Folgen einer personenbezogenen Erhebung und Verwendung von Daten. Personenbezogene Daten sind nach § 3 Abs. 1 Bundesdatenschutzgesetz (BDSG) „Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person“, d.h. Angaben, die direkt oder mit geeignetem Aufwand einer natürlichen Person zugeordnet werden können (vgl. [BDS06]).

Der Schutz von im Internet anfallenden personenbezogenen Daten in Deutschland ist auf verschiedenen Ebenen durch unterschiedliche Gesetze geregelt (vgl. [Patzak06]). Auf Bedeutungs- oder Inhaltsebene einer Kommunikation gelten das BDSG bzw. auf Länderebene das entsprechende Landesdatenschutzgesetz (LDSG). Neben diesen allgemeinen Regelungen gibt es vorrangige, bereichsspezifische Regelungen für die Nutzung von Telemedien, diese sind im Telemediengesetz (TMG) formuliert. Auf der Transportebene gilt das Telekommunikationsgesetz (TKG).

Das TMG findet Anwendung bei allen elektronischen Informations- und Kommunikationsdiensten, soweit sie nicht ausschließlich Telekommunikationsdienste oder Rundfunk sind (§ 1 Abs. 1 TMG, vgl. [TMG07]). Der Begriff der Telemedien umfasst ein breites Gebiet, Beispiele für Telemediendienste sind neben reinen Informationsangeboten Online-Angebote von Waren oder Dienstleistungen mit unmittelbarer Bestellmöglichkeit, sowie Online-Dienste, die Instrumente zur Datensuche, zum Zugang zu Daten oder zur Datenabfrage bereitstellen. (vgl. [Geis07a]).

Das TMG unterteilt in den §§ 14, 15 personenbezogene Daten in drei Kategorien:

- *Bestandsdaten* sind personenbezogene Daten, die für die Begründung, inhaltliche Ausgestaltung oder Änderung eines Vertragsverhältnisses zur Nutzung von Telemediendiensten erforderlich sind (vgl. § 14 Abs. 1 TMG).
- *Nutzungsdaten* sind personenbezogene Daten, die erforderlich sind, um die Nutzung eines Telemediendienstes zu ermöglichen und abzurechnen, wie z.B. Merkmale zur Identifikation des Nutzers oder Angaben über Beginn und Ende der Nutzung bzw. über die Telemedien, die der Nutzer in Anspruch genommen hat (vgl. § 15 Abs. 1 TMG).
- *Abrechnungsdaten* sind personenbezogene Daten, die für Zwecke der Abrechnung mit dem Nutzer erforderlich sind (vgl. § 15 Abs. 4 TMG).

Die Regelungen zum Datenschutz in TMG und BDSG basieren auf folgenden Prinzipien (vgl. [Batista00]):

- Erlaubnisvorbehalt
- Zweckbindung
- Transparenz
- Unabhängige Kontrolle
- Selbst- und Systemdatenschutz

Erlaubnisvorbehalt Die Erhebung und Verwendung von personenbezogenen Daten ist nur zulässig, wenn das TMG oder eine andere auf Telemedien bezogene Rechtsvorschrift es erlaubt oder aber der Nutzer in die Verwendung seiner Daten eingewilligt hat (vgl. § 12 Abs. 1 TMG). Diese Einwilligung muss freiwillig erfolgen, es muss also eine Wahl zwischen Einverständnis und Verweigerung geben, der Nutzer darf keinerlei Zwängen ausgesetzt sein (vgl. [Roßnagel03]). Zugunsten dieser Freiwilligkeit enthält das TMG in § 12 Abs. 3 ein Kopplungsverbot, das besagt, dass die Bereitstellung von Telemedien nicht gekoppelt werden darf an die Einwilligung des Nutzers in eine Verwendung seiner Daten für andere Zwecke, sofern der Nutzer nicht oder nicht in zumutbarer Weise einen anderen Zugang zu diesen Telemedien hat.

Die Einwilligung kann nach § 13 Abs. 2 TMG elektronisch erklärt werden, sofern der Nutzer diese bewusst und eindeutig erteilt, die Einwilligung protokolliert wird und ihr Inhalt für den Nutzer jederzeit abrufbar ist. Die Einwilligung muss vom Nutzer jederzeit widerrufen werden können.

Zweckbindung Für die rechtmäßig erhobenen Daten besteht eine strikte Zweckbindung, die Daten dürfen ausschließlich für die Zwecke verwendet werden, die gesetzlich erlaubt sind oder in die der Nutzer eingewilligt hat (vgl. § 12 Abs. 2 TMG).

Über Bestands und Abrechnungsdaten darf der Diensteanbieter auf Anordnung der zuständigen Stellen Auskunft geben, als mögliche Zwecke nennt das TMG Strafverfolgung, Gefahrenabwehr durch Polizeibehörden der Länder, Erfüllung der gesetzlichen Aufgaben der Verfassungsschutzbehörden des Bundes und der Länder, des Bundesnachrichtendienstes oder des Militärischen Abschirmdienstes oder zur Durchsetzung der Rechte am geistigen Eigentum (vgl. § 14 Abs. 4 und § 15 Abs. 5 TMG).

Nutzungsdaten dürfen dann über den Nutzungsvorgang hinaus verwendet werden, wenn sie für Abrechnungszwecke erforderlich sind (vgl. § 15 Abs. 4 TMG). Nutzungsprofile dürfen nach § 15 Abs. 3 TMG unter Verwendung von Pseudonymen für Zwecke der Werbung, der Marktforschung oder zur bedarfsgerechten Gestaltung der Telemedien erstellt werden, sofern der Nutzer dieser Verwendung seiner Daten nicht widerspricht. Anonymisierte Nutzungsdaten dürfen für Zwecke der Marktforschung an andere Diensteanbieter weitergegeben werden (vgl. § 15 Abs. 5 TMG).

Transparenz Transparenz wird geschaffen, indem der Nutzer vor der Datenerhebung über diese aufgeklärt wird und ihm während der Datenerhebung Anspruch auf Auskunft eingeräumt wird (vgl. [Roßnagel03]). Vor jeder Erhebung muss der Nutzer unterrichtet werden über Art, Umfang und Zwecke der Erhebung und Verwendung personenbezogener Daten sowie über die Möglichkeit, eine Einwilligung jederzeit widerrufen zu können (siehe § 13 Art. 1,3 TMG). Auch über eine Weitervermittlung zu einem anderen Diensteanbieter muss der Nutzer nach § 13 Abs. 5 TMG informiert werden.

Sofern ein Dienst anonym oder unter Pseudonym nutzbar ist, muss der Nutzer auf diese Möglichkeit rechtzeitig – also vor der Erhebung personenbezogener Daten – hingewiesen werden (siehe § 13 Art. 6 TMG). Diese Unterrichtung muss nicht nur über das „Ob“ Auskunft geben, sondern auch erläutern, wie eine anonyme oder pseudonyme Nutzung möglich ist (vgl. [Roßnagel03]).

Der Diensteanbieter muss dem Nutzer nach § 13 Abs. 7 TMG auf Verlangen Auskunft geben über die zu seiner Person oder seinem Pseudonym gespeicherten Daten. Diese Auskunft kann auch elektronisch erfolgen und erstreckt sich nach § 34 Abs. 1 BDSG über die zur Person gespeicherten Daten, die Empfänger oder Kategorien von Empfängern der Daten sowie über den Zweck der Speicherung. Ferner hat der Nutzer nach § 35 BDSG das Recht, die Korrektur oder die Löschung bzw. Sperrung seiner Daten zu verlangen.

Hier wird der Bezug zu P3P deutlich, der P3P-Standard kann als technisches Hilfsmittel zur Förderung der gesetzlich gebotenen Transparenz eingesetzt werden. Inwiefern P3P die hier aufgeführten Forderungen ausdrücken kann, beschreibt Abschnitt 5.1.2.

Unabhängige Kontrolle Nach [Wohlgemuth05] wird die Einhaltung der Datenschutzvorschriften von Kontrollinstanzen überprüft. Zu diesen Instanzen gehört nach § 38 BDSG die Aufsichtsbehörde, sie ist für die Privatwirtschaft zuständig. Öffentliche Stellen der Länder und Gemeinden fallen unter die Zuständigkeit des jeweiligen Landesbeauftragten für den Datenschutz (LfD), ent-

sprechend ist für die öffentlichen Stellen des Bundes der Bundesbeauftragte für den Datenschutz (BfD) zuständig. Eine Ausnahme ist die Kontrolle der Telekommunikationsunternehmen, sie werden nach § 115 Abs. 4 TKG der Kontrolltätigkeit des BfD zugeordnet (vgl. [TKG07]).

Selbst- und Systemdatenschutz Das Konzept des Selbstdatenschutz gründet auf der Idee, dass der Nutzer die Möglichkeit hat, seine personenbezogenen Daten selbst zu schützen. Dies muss zum einen durch Technik, zum anderen durch eine entsprechende rechtliche Absicherung möglich sein (vgl. [Roßnagel03]).

Hinter dem Begriff des Systemdatenschutzes steht die Zielsetzung, Datenschutz und Datensicherheit zum integralen Bestandteil von Datenverarbeitungssystemen zu machen, um so nicht ausschließlich darauf zu bauen, dass Menschen sich bei der Datenverarbeitung an rechtliche Regelungen halten oder aber ihre Rechte kennen und einfordern (vgl. [Roßnagel03]).

Das BDSG enthält in § 3a den Grundsatz der Datenvermeidung und Datensparsamkeit: Datenverarbeitungssysteme sollen so gestaltet und ausgewählt werden, dass sie keine oder so wenig wie möglich personenbezogene Daten erheben, verarbeiten oder nutzen. Präzisiert wird diese Vorgabe durch die Pflicht, Anonymisierung und Pseudonymisierung einzusetzen, wenn dies möglich ist und der Aufwand angemessen im Verhältnis zum angestrebten Schutzzweck ist. Diese Pflicht ist bedeutsam für Selbst- und Systemdatenschutz, da sie sich auf die Gestaltung der technischen Systeme ebenso wie die Möglichkeiten des Nutzers auswirkt.

5.1.2 Anwendung des TMG auf P3P

Bei Entwicklung von P3P als weltweitem Standard bestand die Schwierigkeit, die in verschiedenen Ländern vorhandenen unterschiedlichen rechtlichen Regelungen über den Datenschutz zu vereinheitlichen. Daraus entstanden die folgenden allgemeinen P3P-Datenschutzrichtlinien (vgl. [Wenning06]):

- Notice
- Choice
- Integrity
- Security

Notice

Der Anbieter eines Webangebots unterrichtet den Nutzer durch P3P-Policies über seine Datenschutzpraktiken. Diese Unterrichtung sollte nach Empfehlung des W3C in einer sog. „Safe Zone“ stattfinden, innerhalb der keine personenbezogenen Daten erhoben werden. In der Save Zone sollte der Nutzeragent alle

nicht unbedingt erforderlichen Informationen unterdrücken, d.h. keinen HTTP-Referer-Header versenden, keine Cookies annehmen und keine Informationen über das verwendete System versenden.

Unterrichtung durch Policies: Der Nutzer muss nach § 13 Abs. 1 und 3 TMG unterrichtet werden über Art, Umfang und Zwecke der Erhebung und Verwendung personenbezogener Daten sowie über das Recht, seine Einwilligung zu widerrufen. Folgende Punkte können durch das P3P-Vokabular ausgedrückt werden, die Abbildung dieser Punkte auf P3P-Elemente erfolgt in Anlehnung an [Batista00]:

- *Identität des Anbieters:* Der Anbieter eines Telemediendienstes muss ein Anbieterkennzeichen veröffentlichen. Laut § 5 TMG muss er u.a. Namen, Anschrift und eine elektronische Kontaktmöglichkeit angeben. Diese Angaben können durch das ENTITY-Element des P3P-Standards beschrieben werden. Durch die Kindelemente des ENTITY-Elements lassen sich die Identität des Anbieters sowie Kontaktinformationen wie Adresse, E-Mailadresse und URL des Webangebots angeben.

- *Geltungsbereich:* Der Ort der Datenerhebung wird in der Policy-Referenzdatei definiert. Innerhalb des Elements POLICY-REF können URIs eingeschlossen bzw. ausgeschlossen werden. Dasselbe gilt für Cookies. Weiterhin kann der Geltungsbereich auf bestimmte HTTP-Anfragemethoden begrenzt werden.

- *Umfang der Datenerhebung:* Der Umfang der Datenerhebung wird durch datatype-Elemente innerhalb des DATAGROUP-Elements definiert. Ein datatype-Element referenziert ein Datenelement aus dem „P3P Base Data Schema“, das hierarchisch organisiert ist und den einzelnen Datenarten jeweils Kategorien zuordnet (vgl. 2.4.2). Eine Unterscheidung von Bestands-, Nutzungs- und Abrechnungsdaten wie im TMG gibt es für P3P nicht, in [Batista00] werden folgende Zuordnungsmöglichkeiten festgestellt: Datenkategorien wie „Physical Contact Information“, „Online Contact Information“ oder „Unique Identifiers“ können als Bestandsdaten angesehen werden. Nutzungsdaten wären Datenelemente aus „Computer Information“, „Navigation and Clickstream Data“ und „Interactive Data“. Für Abrechnungsdaten gibt es keine P3P-Entsprechung, da diese aus Vertrags- und Nutzungsdaten abgeleitet werden.

- *Zweck der Datenerhebung:* Der Zweck einer Datenerhebung wird durch das PURPOSE-Element definiert. Diesem Element können ein oder mehrere Kindelemente zugeordnet werden, die vordefinierte Zwecke beschreiben. Das TMG definiert Bestands-, Nutzungs- und Abrechnungsdaten, die implizit ihren Erhebungszweck beinhalten.

  Bestandsdaten werden zur Begründung, Gestaltung oder Änderung eines Vertragsverhältnisses erhoben. Zum Ausdruck dieses Zwecks bietet sich der Wert `current` an, der aussagt, dass die Daten zur Abwicklung

der Aktivität verwendet werden dürfen, für die sie angegeben wurden. Problematisch ist die Verwendung dieses Werts beispielsweise für Online-Bestellungen, bei denen ein Anbieter nach Auslieferung der Ware die Bestandsdaten nicht löscht, sondern weiterhin behält. Für solche Fälle könnte der Wert `other-purpose` eingesetzt werden, der eine natürlichsprachliche Erklärung des Zwecks erfordert. Diese ist als Text im Element enthalten, ideal wäre ein Verweis auf eine Webseite, die detaillierte Informationen zu diesem Zweck bzw. der Vertragsgestaltung enthält.

Für Zwecke der Werbung oder Marktforschung können die Werte `develop` (Research and Development) oder `contact` (Contacting Visitors for Marketing of Services or Products) angegeben werden.

Nutzungsdaten fallen bei der Inanspruchnahme von Telemedien an und werden gewöhnlich in den Log-Dateien eines Webservers protokolliert. Es dürfen hier nur Daten erhoben werden, die für diese Inanspruchnahme erforderlich sind. Wenn Informationen über den vom Nutzer verwendeten Browser erhoben werden, um die Webseiten entsprechend anzupassen, kann dieser Zweck durch den Wert `tailoring` (One-time Tailoring) modelliert werden. Eine Speicherung von Informationen zur Nutzer-Interaktion darf im Hinblick auf die Erforderlichkeit nur vom darunterliegenden Server-System verwendet werden, hierfür kann der Wert `admin` (Web Site and System Administration) eingesetzt werden.

Für Abrechnungsdaten kann innerhalb des `current`-Wertes durch das Kindelement PPURPOSE, das den primaren Zweck der Datenerhebung beschreibt, auf Abrechungszwecke hingewiesen werden. Dafür können innerhalb von PPURPOSE die Werte `account` (Account and/or Subscription Management) und `payment` (Payment and Transaction Facilitation) verwendet werden.

- *Weitergabe von Daten:* Anbieter dürfen laut § 15 Abs. 5 TMG Abrechnungsdaten an andere Anbieter oder Dritte übermitteln, wenn dies zu Abrechnungszwecken erforderlich ist. Für Bestands-, Nutzungs- und Abrechnungsdaten definiert das Gesetz eine Zweckbindung (siehe 5.1.1). Daher bietet sich für das Element RECIPIENT, das den oder die Emfänger der erhobenen Daten angibt, der Wert `ours` an. Dieser Wert umfasst den Anbieter selbst und Stellen, die Daten im Auftrag des Anbieters für den angegebenen Zweck verarbeiten.

  Eine spezielle Regelung zu Nutzungsdaten enthält der oben genannte Absatz: In anonymisierter Form dürfen Nutzungsdaten zum Zweck der Marktforschung an andere Diensteanbieter übermittelt werden. Um anzuzeigen, dass die Daten anonymisiert sind, muss im entsprechenden STATEMENT-Element das Kindelement NON-IDENTIFIABLE verwendet werden (siehe Abschnitt 2.4.2). Für die Angabe des Empfängers stehen folgende Werte zur Verfügung: `same` (Legal entities following our practices), `other-recipient` (Legal entities following different practices) oder `unrelated` (Unrelated third parties). Letzterer Wert sagt aus,

dass der Anbieter nichts über die Datenschutzpraktiken des Empfängers weiß.

- *Online-Unterrichtung:* Das TMG verpflichtet den Anbieter zur Unterrichtung des Nutzers über eine Datenerhebung. Eine natürlichsprachliche Unterrichtung kann im P3P-Element CONSEQUENCE angegeben werden. Dieses Element ist in einem P3P-Statement enthalten und beschreibt dessen Datenschutzpraktiken. Die P3P-Spezifikation definiert das CONSEQUENCE-Element als optional, es sollte allerdings verwendet werden, um dem TMG zu entsprechen.

  Verpflichtend laut P3P-Spezifikation ist dagegen die Angabe einer URI im *discuri*-Attribut des POLICY-Elements. Diese URI verweist auf die Webseite, die die natürlichsprachliche Datenschutzerklärung des Anbieters enthält. Dies erfüllt die Aussage des TMG, dass die Unterrichtung für den Nutzer jederzeit abrufbar sein muss (vgl. § 13 Abs. 1 TMG).

  Nach TMG muss auch die Weitervermittlung des Nutzers zu einem anderen Anbieter angezeigt werden (vgl. § 13 Abs. 5 TMG). Sofern die Weitervermittlung durch das Anklicken eines Links im Webangebot des Anbieters ausgelöst wird, erfolgt diese Anzeige durch die Ausgabe des neuen URI in der Statuszeile des Web-Clients. Zudem könnte ein P3P-Nutzeragent den Nutzer informieren, wenn der Geltungsbereich einer Policy verlassen wird.

  Werden dagegen Anfragen eines URI automatisch an Dritte weitergeleitet und dabei personenbezogene Daten des Nutzers innerhalb des Weiterleitungs-URI eingebettet, so muss der Anbieter die übermittelten Datenarten und die dritte Partei als Empfänger in seiner Policy beschreiben (vgl. [Wenning06]). Die Unterrichtung über diese Weiterleitung erfolgt daher in den P3P-Elementen DATA-GROUP, RECIPIENT und CONSEQUENCE.

- *Auskunft und Einsicht:* Das P3P-Element ACCESS gibt an, welche Art von Auskunft dem Nutzer über die zu seiner Person gespeicherten Daten gewährt wird. Die P3P-Spezifikation verlangt lediglich, dass für dieses Element ein Wert angegeben wird. Der Wert `none` besagt, dass dem Nutzer keinerlei Auskünfte gegeben werden. Dieser Wert wäre nicht konform zum TMG, welches in § 13 Abs. 7 fordert, dass der Nutzer auf Verlangen nach Maßgabe von § 34 BDSG Auskunft über die zu seiner Person oder zu seinem Pseudonym gespeicherten Daten erhalten muss. Daher empfiehlt sich für eine TMG-konforme Policy der ACCESS-Wert `all` (All Identified Data), sofern personenbezogene Daten erhoben werden.

  Erhebt ein Diensteanbieter dagegen keine benutzeridentifizierenden Daten, kann auch der Wert `nonident` eingesetzt werden. In diesem Fall muss in den Statements, die in der Policy enthalten sind, das NON-IDENTIFIABLE-Element vorhanden sein.

  Das TMG sagt aus, dass dem Nutzer auch elektronisch Auskunft erteilt werden kann (vgl. § 13 Abs. 7 TMG). Dieser Punkt wird von der

P3P-Spezifikation nicht behandelt. Das ACCESS-Element macht lediglich Aussagen darüber, ob dem Nutzer Auskunft gegeben wird, und ob diese Auskunft vollständig oder eingeschränkt auf bestimmte Datenarten ermöglicht wird.

- *Widerruf von Einwilligungen:* Der Diensteanbieter hat nach § 13 Abs. 3 TMG die Pflicht, den Nutzer vor Erklärung der Einwilligung darüber zu informieren, dass diese Einwilligung jederzeit mit Wirkung für die Zukunft widerrufen werden kann. Der P3P-Standard bietet kein Element, das dieses Recht zum Ausdruck bringt. Der Hinweis kann aber natürlichsprachlich erfolgen, beispielsweise in der natürlichsprachlichen Datenschutzerklärung, auf die das *discuri*-Attribut des POLICY-Elements verweist.

  Für zwei der in einem Statement enthaltenen Elemente kann das Attribut *required* gesetzt werden, diese Elemente sind PURPOSE und RECIPIENT. Diesem Attribut können u.a. die Werte `opt-in` und `opt-out` zugewiesen werden, was dem Nutzer die Möglichkeit zur expliziten Einwilligung bzw. zum expliziten Widerspruch einräumt (siehe Abschnitt 2.4.2). Diese Möglichkeit ist allerdings nur für diese beiden Elemente vorgesehen, die Einwilligung zur Policy als Ganzes kann dagegen nicht in P3P ausgedrückt werden.

- *Löschfristen:* Bestandsdaten müssen nach Ende des Vertragsverhältnisses gelöscht werden (vgl. § 14 Abs. 1 TMG, § 35 Abs. 2 Nr. 3 BDSG). Nutzungsdaten dürfen nur über das Ende des Nutzungsvorgangs hinaus verwendet werden, wenn sie zu Abrechnungszwecken benötigt werden (vgl. § 15 Abs. 4 TMG). Auch die Dauer der Speicherung von Abrechnungsdaten ist demnach an die Erforderlichkeit für den Zweck der Abrechnung geknüpft.

  Die Aufbewahrungsdauer der erhobenen Daten wird in P3P durch das RETENTION-Element ausgedrückt. Hierfür stehen vordefinierte Werte zur Verfügung. Aufgrund der für alle Datenarten vorliegenden Zweckbindung bietet sich der Wert `stated-purpose` an: Die erhobenen Daten werden für den angegebenen Zweck verwendet und so früh wie möglich gelöscht. Bei Angabe dieses Werts müssen in der natürlichsprachlichen Datenschutzerklärung konkretere Angaben zur Aufbewahrungsdauer gemacht werden.

  Für Nutzungsdaten kommt der restriktivere Wert `no-retention` in Betracht, der die Speicherung jeglicher Daten auf die Dauer einer einzelnen Online-Interaktion beschränkt. Gemäß der P3P-Spezifikation kann dieser Wert beispielsweise für Webangebote eingesetzt werden, die keine Server-Logdateien führen, die Cookies nur für eine Sitzung setzen oder Suchanfragen ermöglichen, diese aber nicht in Logdateien protokollieren.

- *Kontrollinstanz:* Die Einhaltung von Datenschutzvorschriften wird von Kontrollinstanzen überprüft (vgl. Abschnitt 5.1.1). Auf diese Stellen können Anbieter mit Hilfe des P3P-Element DISPUTES verweisen. Hier können sämtliche Stellen angegeben werden, an die sich ein Nutzer bei Streitigkeiten mit dem Anbieter wenden kann.

Choice

P3P ermöglicht dem Nutzer nur eine implizite Einwilligung in die Datenschutzpraktiken des Anbieters. Der Nutzer trifft eine Entscheidung zwischen der Nutzung eines Angebots und damit der Akzeptanz der Datenschutzpraktiken des Anbieters oder dem Verzicht auf Nutzung des Angebots. Das TMG fordert eine explizite Einwilligung nur dann, wenn die Erhebung und Verwendung personenbezogener Daten über die Erlaubnisnormen des TMG oder anderer Rechtsvorschriften hinausgeht, die sich ausdrücklich auf Telemedien beziehen. [Batista00] beschreibt eine entsprechende Umsetzung auf den P3P-Standard: Ein Nutzeragent müßte so konfiguriert werden, dass er die (minimalen) datenschutzrechtlichen Forderungen in Form von Nutzerpräferenzen beinhaltet. Solange dieser Nutzeragent Policies empfängt, deren Regelungen mit den gesetzlichen Vorschriften in Einklang stehen, erfolgt der Abruf der entsprechenden Webseiten und damit die implizite Einwilligung. Wird dagegen eine Policy empfangen, die den Präferenzen widerspricht, kann der Nutzer entweder entscheiden, das Angebot trotzdem zu nutzen und damit explizit einzuwilligen oder aber das Angebot abzulehnen. Letzteres wäre insofern für deutsche Anbieter problematisch, als das TMG fordert, dass die Bereitstellung von Telemedien nicht an die Einwilligung des Nutzers in eine Verwendung seiner Daten für andere Zwecke gekoppelt werden darf. Eine Ausnahme hiervon besteht nur, wenn dem Nutzer ein anderer Zugang zu diesen Telemedien auf zumutbare Weise möglich ist (vgl. § 12 Abs. 3 TMG).

Das Recht auf Widerruf einer Einwilligung mit Wirkung für die Zukunft ist nicht durch ein spezielles P3P-Element abbildbar (siehe Abschnitt 5.1.2).

Das TMG erlaubt auch elektronische Einwilligungen, sofern der Anbieter dafür sorgt, dass der Nutzer seine Einwilligung bewusst und eindeutig erteilt, die Einwilligung protokolliert wird, der Inhalt der Einwilligung jederzeit abrufbar ist, und der Nutzer darauf hingewiesen wird, dass er seine Einwilligung jederzeit widerrufen kann (vgl. § 13 Abs. 2,3 TMG). Zur Abwicklung einer elektronischen Einwilligung empfiehlt sich die Verwendung digitaler Signaturen, um die Einwilligung nichtabstreitbar zu machen. Mechanismen zur Herstellung von Nichtabstreitbarkeit sind in P3P noch nicht verwirklicht, sind aber für zukünftige Versionen des Standards geplant (vgl. [Wenning06]).

Integrity

Die P3P-Spezifikation enthält Richtlinien, die auf einen verantwortungsvollen und fairen Gebrauch von P3P abzielen. Um die Daten ihrer Nutzer zu schützen und Vertrauen zu fördern, sollten Diensteanbieter demnach ihre Datenschutzpraktiken klar und eindeutig erklären. Sie sollten Daten nur für die

bei der Erhebung angegebenen Zwecke verwenden und sie nicht länger als nötig speichern. Weiterhin sollten Anbieter Verantwortlichkeiten klar definieren und sicherstellen, dass die Daten eines Nutzers gemäß der Policy verwendet werden, unter der sie erhoben wurden (vgl. [Wenning06]).

Während manche dieser Empfehlungen – wie z.B. Zweckbindung und Erforderlichkeit – im deutschen Datenschutzrecht ihre Entsprechung finden, haben andere keine Gesetzeskraft. In [Batista00] wird daher die Standardisierung von Policies und Präferenzen empfohlen. Standardisierte Policies für verschiedene Internet-Anwendungen wie beispielsweise Online-Datenbanken oder E-Commerce könnten für Anbieter bereitgestellt werden. Wenn eine solche Policy verwendet wird, müßte sich der Anbieter nach den darin enthaltenen Angaben richten. Empfohlene Präferenzdateien helfen den Nutzern, ihre Nutzeragenten entsprechend zu konfigurieren.

Security

Der P3P-Standard selbst enthält keine Sicherheitsmechanismen, sollte aber in Verbindung mit Sicherheitswerkzeugen genutzt werden. Die P3P-Spezifikation legt Anbietern nahe, zusätzliche Maßnahmen zum Schutz der erhobenen personenbezogenen Daten zu ergreifen und für eine sichere Übertragung der Daten zu sorgen (vgl. [Wenning06]). Auch das TMG fordert, dass der Nutzer Telemedien gegen Kenntnisnahme Dritter geschützt in Anspruch nehmen kann (siehe § 13 Abs. 3). Dies kann z.B. durch eine SSL/TLS-Verschlüsselung verwirklicht werden.

Die Verarbeitung der Daten beim Anbieter unterliegt der Datenschutzkontrolle, das Schutzniveau kann durch Maßnahmen aus dem Bereich des Systemdatenschutzes erhöht werden (vgl. Abschnitt 5.1.1).

5.2 Templates

In den folgenden Abschnitten werden die Templates vorgestellt, die im Policy-Generator zur Verfügung stehen.

5.2.1 Minimal-Template

Eine Policy, die konform zum TMG ist, setzt die Prinzipien des vorigen Abschnitts in P3P um und erhebt möglichst wenige personenbezogene Daten. Die Bestandteile der hier vorgestellten Templates sind durch den Generator änderbar, um eine Anpassung an die individuellen Bedürfnisse eines Anbieters zu ermöglichen. Zunächst wird nun ein Minimal-Template vorgestellt, das z.B. von einfachen Informationsangeboten verwendet werden könnte, die lediglich Informationen bereitstellen und ihren Nutzern ermöglichen, per Mail Kontakt zu ihnen aufzunehmen. Dieses Minimal-Template ist in Listing 5.1 aufgeführt.

Listing 5.1: Minimal-Template

```
<?xml version="1.0" encoding="UTF-8"?>
<META xmlns="http://www.w3.org/2002/01/P3Pv1"
      xmlns:p3p11="http://www.w3.org/2006/01/P3Pv11"
      xmlns:p3p11bds="http://www.w3.org/2006/01/
          P3Pv11BDS">
<POLICY-REFERENCES>
  <EXPIRY max-age="86400" />
  <POLICY-REF about="#NameIhrerPolicy">
    <INCLUDE>/*</INCLUDE>
  </POLICY-REF>
</POLICY-REFERENCES>
<POLICIES>
  <EXPIRY max-age="86400" />
  <POLICY name="NameIhrerPolicy"
    discuri="www.ihreBehoerde.de/datenschutzerklaerung.
        html"
    xml:lang="de">
    <ENTITY>
      ...
    </ENTITY>
    <ACCESS><all /></ACCESS>
    <DISPUTES-GROUP>
      <DISPUTES resolution-type="service"
        service="www.ihrKundenservice.de">
        <LONG-DESCRIPTION>
          Bei Streitigkeiten wenden Sie sich bitte an
              unseren Kundenservice.
        </LONG-DESCRIPTION>
        <REMEDIES>
          <correct />
        </REMEDIES>
        </DISPUTES>
      <DISPUTES resolution-type="independent"
        service="http://www.info-mainz.de
                          /datenschutz-RP/index.html">
        <LONG-DESCRIPTION>
          Der Landesbeauftragte für den Datenschutz
              überwacht die Einhaltung des LDSG und
              anderer Vorschriften über den Datenschutz.
        </LONG-DESCRIPTION>
        <REMEDIES>
          <law />
        </REMEDIES>
        </DISPUTES>
        <DISPUTES resolution-type="law"
```

```
41          service="http://www.gesetze-im-internet.de/tmg/"
              >
42          <LONG-DESCRIPTION>
43            Die erhobenen personenbezogenen Daten fallen
                unter den Schutz des TMG.
44          </LONG-DESCRIPTION>
45          <REMEDIES>
46            <law/>
47          </REMEDIES>
48          </DISPUTES>
49      </DISPUTES-GROUP>
50      <STATEMENT>
51          <CONSEQUENCE>
52          Ihr Name und Ihre Kontaktdaten werden erhoben,
              falls Sie mit uns Kontakt aufnehmen. Ihre
              Daten werden verwendet, um auf Ihre Anfragen
              zu antworten. Mit der Angabe willigen Sie in
              diese Nutzung ein. Diese Einwilligung können
              Sie jederzeit mit Wirkung für die Zukunft
              widerrufen.
53          </CONSEQUENCE>
54        <PURPOSE>
55          <EXTENSION optional="yes">
56            <PPURPOSE xmlns="http://www.w3.org/2006/01/
                P3Pv11">
57              <feedback/>
58            </PPURPOSE>
59          </EXTENSION>
60          <current/>
61        </PURPOSE>
62        <RECIPIENT>
63          <EXTENSION>
64            <JURISDICTION
65                  service="http://www.gesetze-im-internet.
                      de/tmg/"
66                  short-description="Telemediengesetz (TMG
                      )">
67                  Telemediengesetz vom 26. Februar 2007 (
                      BGBl. I S. 179), Ausfertigungsdatum:
                      26.02.2007.
68              </JURISDICTION>
69          </EXTENSION>
70          <ours/>
71        </RECIPIENT>
72        <RETENTION>
73          <stated-purpose/>
```

```
74          </RETENTION>
75          <DATA-GROUP>
76            <EXTENSION>
77              <p3p11:DATA-GROUP>
78                      ...
79              </p3p11:DATA-GROUP>
80            </EXTENSION>
81            <DATA ref="#user.name" optional="yes"/>
82            <DATA ref="#user.home-info.postal" optional="yes
                "/>
83            <DATA ref="#user.home-info.telecom" optional="
                yes"/>
84            <DATA ref="#user.home-info.online.email"
                optional="yes"/>
85          </DATA-GROUP>
86        </STATEMENT>
87        <STATEMENT>
88          <CONSEQUENCE>
89            Wir protokollieren Zugriffe auf unser Webangebot
                 in Protokolldateien. Diese Informationen
                werden 3 Monate lang gespeichert und danach
                automatisch gelöscht. Dabei werden
                Informationen, über die sich ein
                Personenbezug herstellen ließe, verkürzt und
                gefiltert gespeichert (siehe unsere
                Datenschutzerklärung).
90          </CONSEQUENCE>
91          <NON-IDENTIFIABLE/>
92          <PURPOSE>
93            <EXTENSION optional="yes">
94              <PPURPOSE xmlns="http://www.w3.org/2006/01/
                    P3Pv11">
95                <browsing/>
96              </PPURPOSE>
97            </EXTENSION>
98            <admin/>
99            <tailoring/>
100         </PURPOSE>
101         <RECIPIENT>
102           <EXTENSION>
103             <JURISDICTION
104                     service="http://www.gesetze-im-internet.
                        de/tmg/"
105                     short-description="Telemediengesetz (TMG
                        )">
106                     Telemediengesetz vom 26. Februar 2007 (
```

```
                    BGBl. I S. 179), Ausfertigungsdatum:
                    26.02.2007.
107             </JURISDICTION>
108         </EXTENSION>
109         <ours/>
110       </RECIPIENT>
111       <RETENTION>
112         <stated-purpose/>
113       </RETENTION>
114       <DATA-GROUP>
115         <EXTENSION>
116           <p3p11:DATA-GROUP>
117             ...
118           </p3p11:DATA-GROUP>
119         </EXTENSION>
120         <DATA ref="#dynamic.clickstream.uri"/>
121         <DATA ref="#dynamic.clickstream.timestamp"/>
122         <DATA ref="#dynamic.clickstream.other.bytes"/>
123         <DATA ref="#dynamic.clickstream.clientip.
              partialip"/>
124         <DATA ref="#dynamic.http"/>
125       </DATA-GROUP>
126     </STATEMENT>
127   </POLICY>
128 </POLICIES>
129 </META>
```

Im Element POLICY-REFERENCES gibt es nur eine Referenz, sie verweist auf die Policy, die im Template enthalten ist (ab Zeile 5). Durch den Wert `/*` des INCLUDE-Elements wird der Geltungsbereich dieser Policy definiert: Die referenzierte Policy gilt für das gesamte Webangebot.

Sowohl für die Policy als auch für die Policy-Referenzen wurde für das EXPIRY-Element, das die Gültigkeitsdauer definiert, der Standardwert von einem Tag (86400 Sekunden) beibehalten. Diese Dauer ist relativ zu dem Zeitpunkt zu verstehen, an dem der Nutzer die P3P-Datei abruft. Die Wahl dieses Werts wird von [Cranor02a] empfohlen, um Policies möglichst einfach ändern zu können. Anbieter sollten allerdings beachten, dass einmal erhobene Daten generell so behandelt werden müssen, wie es in der Policy definiert war, unter der die Daten erhoben wurden, unabhängig davon, ob in der Zwischenzeit eine geänderte Policy veröffentlicht wurde. Da der Wert des EXPIRY-Elements der erlaubten Cache-Dauer auf Nutzerseite entspricht, müssen evtl. Übergangsfristen beachtet werden, in denen noch nach der alten Policy verfahren wird.

Der Hinweis auf die URI der natürlichsprachlichen Datenschutzerklärung erfolgt in Zeile 14 im Attribut *discuri* des POLICY-Elements. Dieser Wert muss vom Nutzer editiert werden. Im ENTITY-Element ist das Anbieterkennzeichen enthalten, hier fragt der Generator die Kontaktdaten des Anbieters ab. Dazu

zählen u.a. der Name der Organisation, die Adresse, die Telefonnummer, eine E-Mailadresse usw. Das ACCESS-Element beschreibt, über welche Daten dem Nutzer Auskunft erteilt wird (siehe Zeile 19). Hier ist der Wert `all` gewählt, dem Nutzer wird also Auskunft über alle zu ihm gespeicherten personenbezogenen Daten gewährt, womit die entsprechende Forderung des TMG erfüllt wird.

Kontrollinstanzen und sonstige Anlaufstellen bei Streitigkeiten sind im Element DISPUTES-GROUP aufgelistet (ab Zeile 20). Das Template enthält zwei Statements, eines für Kontaktdaten, die bei Nutzeranfragen anfallen und verwendet werden, um diese Fragen zu beantworten. Das zweite Statement gibt die Erhebung von anonymisierten Nutzungsdaten an, die in Logdateien eines Webservers gespeichert werden. Eine natürlichsprachliche Unterrichtung über die mit dem Statement verbundene Datenerhebung erfolgt im CONSEQUENCE-Element (siehe Zeile 51 und 89).

Die gesetzlich vorgeschriebene Bindung der erhobenen Daten an einen bestimmten Zweck wird im PURPOSE-Element definiert (vgl. Zeile 54 bzw. 92). Im Falle der Kontaktdaten ist der Wert `feedback` als Hauptzweck ausgewählt. Der Wert für das PURPOSE-Element ist `current`, die Daten werden demnach für die Aktivität verwendet, für die sie angegeben wurden. Im Statement für die Nutzungsdaten ist der Hauptzweck `browsing`, für PURPOSE sind die Zwecke `admin` und `tailoring` eingesetzt. Die Daten werden also zur Administration des Webangebots und zur Anpassung der Webseiten an die Gegebenheiten auf Nutzerseite genutzt.

In beiden Statements enthält das RECIPIENT-Element den Wert `ours`, was besagt, dass die Daten nicht an Dritte weitergegeben werden (vgl. Zeile 62 bzw. 101). Zudem ist in diesem Element das JURISDICTION-Element eingebunden, das Auskunft über die für den Empfänger geltende Gesetzgebung gibt (siehe Zeile 64 und 103). Dieses Element wird nicht vom Generator abgefragt, da diese Angabe für alle deutschen Behörden zutrifft.

Die Aufbewahrungsdauer der erhobenen Daten wird durch das RETENTION-Element angegeben. Hier kommt in beiden Fällen der Wert `stated-purpose` zum Einsatz, demnach werden die Daten für den angegebenen Zweck verwendet und so früh wie möglich gelöscht.

Für welche Daten ein Statement gilt, wird im DATA-GROUP-Element definiert (vgl. Zeile 75 und 114). Aus Gründen der Übersichtlichkeit zeigt Listing 5.1 diese nur in der Form an, die in P3P-Version 1.0 definiert wurde, das Template enthält aber sowohl diese Darstellung als auch die Entsprechung in Version 1.1, um Abwärtskompatibilität zu gewährleisten.

Das Statement für die Kontaktinformationen gibt an, dass Name, postalische Adresse, Telefonnummer und Mailadresse des Nutzers erhoben werden. Das Nutzungsdaten-Statement definiert Datenarten aus den Bereichen `dynamic.clickstream` und `dynamic. http`. Die „clickstream"-Angaben umfassen u.a. die URI der angefragten Ressource, einen Zeitstempel, die Anzahl der übertragenen Bytes und die anonymisierte IP-Adresse des Nutzers.

Dass die IP-Adresse anonymisiert ist, wird zum einen dadurch ausgedrückt, dass der Wert `partialip` verwendet wird, zum anderen sagt das Element NON-IDENTIFIABLE in diesem Statement, dass keine personenbezogenen Daten erhoben werden (vgl. Zeile 91). Unter die Bezeichnung „dynamic.http“ fallen Angaben zum Browser des Nutzers sowie der HTTP-Referer-Header. Letzterer muss ebenfalls anonymisiert sein, damit dem TMG und dem NON-IDENTIFIABLE-Element Rechnung getragen wird. Dies kann durch eine Kürzung auf den vorderen Teil des URI der aufrufenden Stelle erreicht werden (vgl. [LfD02]).

Erhebt ein Anbieter weniger oder andere Daten, kann dies im Generator abgeändert werden. Hier besteht auch die Möglichkeit, konkretere Angaben zu den erhobenen Daten zu machen, indem Unterelemente der hier angegebenen Datenarten selektiert werden. Statt `dynamic.http` könnte beispielsweise `dynamic.http.useragent` angegeben, wenn nur Angaben zum Nutzeragenten, nicht aber der Referer-Header gespeichert wird. Die hierarchische Anordnung aller wählbaren Datenarten ist im Generator durch eine baumartige Darstellung im Bearbeitungsfenster „Datentypen“ umgesetzt.

5.2.2 Template für Suchmaschinen

Das oben beschriebene Minimal-Template wird im folgenden erweitert um ein Statement für Anbieter, die ihren Nutzern eine Suchmaschine zur Verfügung stellen. Die Gestaltung dieser Erweiterung orientiert sich an [Batista00].

Listing 5.2: Template für Suchmaschinen

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <META ...>
3  <POLICY-REFERENCES>
4      ...
5  </POLICY-REFERENCES>
6  <POLICIES>
7    <EXPIRY max-age="86400" />
8    <POLICY ...>
9      ...
10     <STATEMENT>
11       <CONSEQUENCE>
12         Unser Webangebot enthält eine Suchmaschine. Die
               eingegebenen Suchanfragen werden nur zur
               Ausführung der Suche verwendet und nicht mit
               anderen Daten zusammengeführt oder
               ausgewertet. Nach der Ausführung der Suche
               werden die Anfragen automatisch gelöscht.
13         </CONSEQUENCE>
14         <NON-IDENTIFIABLE/>
15         <PURPOSE>
```

```
16            <EXTENSION optional="yes">
17              <PPURPOSE xmlns="http://www.w3.org/2006/01/
                    P3Pv11">
18                <search/>
19              </PPURPOSE>
20              </EXTENSION>
21            <current/>
22          </PURPOSE>
23          <RECIPIENT>
24            <EXTENSION>
25              <JURISDICTION
26                    service="http://www.gesetze-im-internet.
                        de/tmg/"
27                    short-description="Telemediengesetz (TMG
                        )">
28                    Telemediengesetz vom 26. Februar 2007 (
                        BGBl. I S. 179), Ausfertigungsdatum:
                        26.02.2007.
29                </JURISDICTION>
30            </EXTENSION>
31            <ours/>
32          </RECIPIENT>
33          <RETENTION>
34            <no-retention/>
35          </RETENTION>
36          <DATA-GROUP>
37            <EXTENSION>
38              <p3p11:DATA-GROUP>
39                    ...
40              </p3p11:DATA-GROUP>
41            </EXTENSION>
42            <DATA ref="#dynamic.searchtext"/>
43          </DATA-GROUP>
44        </STATEMENT>
45      </POLICY>
46  </POLICIES>
47  </META>
```

Das Statement, das die Datenerhebung durch eine Suchmaschine modelliert, ist in Listing 5.2 abgebildet. Neben den beiden Statements des Minimaltemplates, die Kontaktdaten bei Anfragen sowie Nutzungsdaten behandeln, enthält dieses Template ein Statement, das Aussagen zum Umgang des Anbieters mit Suchanfragen macht. Dieses Template berücksichtigt ebenfalls das TMG.

Die natürlichsprachliche Beschreibung erfolgt im CONSEQUENCE-Element. Das Statement enthält das Element NON-IDENTIFIABLE, es werden also keine personenbezogenen Daten beim Anbieter gespeichert (siehe Zeile 14).

Das einzige Datenelement, das hier erhoben wird, ist `dynamic.searchtext`. Der Wert `no-retention` im RETENTION-Element gibt an, dass die Daten nur für die Dauer einer Nutzerinteraktion aufbewahrt werden, um die Suche ausführen zu können. Dieses Verfahren steht im Einklang mit § 15 Abs. 4 TMG, wonach Nutzungsdaten nur für den jeweiligen Nutzungsvorgang gespeichert werden dürfen, sofern sie nicht für Abrechnungszwecke benötigt werden.

Empfänger der Daten ist der Anbieter selbst, dies wird durch den Wert `ours` im Element RECIPIENT ausgedrückt (ab Zeile 23). Als Hauptzweck ist der Wert `search` gewählt, das PURPOSE-Element enthält den Wert `current`, wonach die Daten nur zur Erledigung der Aufgabe verwendet werden, für die sie angegeben wurden (ab Zeile 15).

5.2.3 Template für ein Dienstleistungsportal

Das Minimaltemplate aus Abschnitt 5.2.1 wird für die Erbringung von Online-Dienstleistungen um zwei weitere Statements ergänzt (vgl. Listing 5.3).

Listing 5.3: Template für Online-Dienstleistungen

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <META ...>
3  <POLICY-REFERENCES>
4      ...
5  </POLICY-REFERENCES>
6  <POLICIES>
7    <EXPIRY max-age="86400"/>
8    <POLICY ...>
9      ...
10     <STATEMENT>
11       <CONSEQUENCE>
12         Wenn Sie sich bei uns registrieren, wird ein
               Nutzerkonto für Sie angelegt. Zur
               Verwaltung Ihres Nutzerkontos werden von
               Ihnen vorgenommene Änderungen an Daten oder
                Einstellungen ihres Nutzerkontos
               protokolliert.
13       </CONSEQUENCE>
14       <PURPOSE>
15         <EXTENSION optional="yes">
16           <PPURPOSE xmlns="http://www.w3.org/2006/01/
                 P3Pv11">
17             <account/>
18           </PPURPOSE>
19         </EXTENSION>
20         <current/>
21       </PURPOSE>
```

```
22        <RECIPIENT>
23          <EXTENSION>
24            <JURISDICTION
25                  service="http://www.gesetze-im-internet.
                        de/tmg/"
26                  short-description="Telemediengesetz (TMG
                        )">
27                  Telemediengesetz vom 26. Februar 2007 (
                        BGBl. I S. 179), Ausfertigungsdatum:
                        26.02.2007.
28              </JURISDICTION>
29          </EXTENSION>
30          <ours/>
31        </RECIPIENT>
32        <RETENTION>
33          <stated-purpose/>
34        </RETENTION>
35        <DATA-GROUP>
36          <EXTENSION>
37            <p3p11:DATA-GROUP>
38                  ...
39            </p3p11:DATA-GROUP>
40          </EXTENSION>
41          <DATA ref="#dynamic.miscdata">
42            <CATEGORIES>
43              <interactive/>
44              <other-category>
45                    Widerruf der Einwilligung
46              </other-category>
47            </CATEGORIES>
48          </DATA>
49        </DATA-GROUP>
50      </STATEMENT>
51      <STATEMENT>
52        <CONSEQUENCE>
53        Zur Nutzung unserer Dienstleistungen benötigen
            Sie eine Registrierung. Falls Sie eine
            kostenpflichtige Dienstleistung nutzen,
            werden Informationen zur Zahlungsart
            gespeichert. Die durch uns erhobenen Daten
            fallen unter den Schutz des
            Telemediengesetzes. Die Erhebung und
            Verarbeitung Ihrer Daten ist daher nur
            zulässig, wenn Sie in diese eingewilligt
            haben. Diese Einwilligung kann mit Wirkung
            für die Zukunft widerrufen werden, indem Sie
```

```
            Ihr Benutzerkonto löschen. Nach Löschung des
            Benutzerkontos werden die Daten für 6 Monate
            gespeichert, danach werden sie endgültig
            gelöscht.
54        </CONSEQUENCE>
55      <PURPOSE>
56        <EXTENSION optional="yes">
57          <PPURPOSE xmlns="http://www.w3.org/2006/01/
                P3Pv11">
58            <account/>
59            <payment/>
60          </PPURPOSE>
61        </EXTENSION>
62        <current/>
63      </PURPOSE>
64      <RECIPIENT>
65        <EXTENSION>
66          <JURISDICTION
67                service="http://www.gesetze-im-internet.
                    de/tmg/"
68                short-description="Telemediengesetz (TMG
                    )">
69                Telemediengesetz vom 26. Februar 2007 (
                    BGBl. I S. 179), Ausfertigungsdatum:
                    26.02.2007.
70            </JURISDICTION>
71        </EXTENSION>
72        <ours/>
73      </RECIPIENT>
74      <RETENTION>
75        <stated-purpose/>
76      </RETENTION>
77      <DATA-GROUP>
78        <EXTENSION>
79          <p3p11:DATA-GROUP>
80                ...
81          </p3p11:DATA-GROUP>
82        </EXTENSION>
83        <DATA ref="#user.name"/>
84        <DATA ref="#user.bdate"/>
85        <DATA ref="#user.home-info.postal"/>
86        <DATA ref="#user.home-info.online.email"/>
87        <DATA ref="#user.login"/>
88        <DATA ref="#business.contact-info.telecom.fax"
              optional="yes"/>
89        <DATA ref="#business.contact-info.postal"
```

```
                optional="yes"/>
90            <DATA ref="#dynamic.miscdata">
91              <CATEGORIES>
92                <purchase/>
93                <other-category>
94                  Daten zum Nutzerkonto: Datum der
                        Einwilligung, in Anspruch genommene
                        Dienstleistungen
95                </other-category>
96              </CATEGORIES>
97            </DATA>
98          </DATA-GROUP>
99        </STATEMENT>
100     </POLICY>
101 </POLICIES>
102 </META>
```

Das erste der beiden ergänzenden Statements beschreibt die Protokollierung von Aktivitäten, die der Nutzer an seinem Nutzerkonto ausführt. Dazu zählen Login-Versuche, die Bearbeitung der Nutzerdaten, die Löschung eines Benutzerkontos usw. Diese Informationen werden im Datenelement `dynamic.miscdata` mit Hilfe der Kategorie `interactive` ausgedrückt. Der Widerruf der Einwilligung wird ebenfalls protokolliert, dies wird durch den Wert `other-category` und eine entsprechende Beschreibung festgehalten (ab Zeile 44).

Die Aufbewahrungsdauer ist durch den Wert `stated-purpose` des Elements RETENTION an den Zweck der Erhebung gebunden (vgl. Zeile 33), als Hauptzweck der Erhebung gibt dieses Statement den Wert `account` an, was die Verwaltung von Benutzerkonten und Registrierungsvorgängen repräsentiert (vgl. Zeile 17). Empfänger der Daten ist der Anbieter selbst, was der Wert `ours` im RECIPIENT-Element zum Ausdruck bringt.

Das zweite ergänzende Statement behandelt die Daten, die erhoben werden, wenn ein Nutzer online eine Dienstleistung in Anspruch nimmt. Es ähnelt in einigen Punkten dem vorherigen Statement, ein Unterschied zeigt sich in der Belegung des Hauptzwecks der Datenerhebung. Hier ist neben `account` auch der Wert `payment` angegeben, da die Zahlungsart protokolliert wird, wenn der Nutzer eine kostenpflichtige Dienstleistung in Anspruch nimmt. Dies spiegelt sich auch durch ein Datenelement wieder: Der Wert `dynamic.miscdata` wird durch Kategorien definiert, in diesem Fall durch die Kategorie `purchase`, die Informationen zum „Kauf“ und zur Zahlungsart repräsentiert sowie durch `other-category`. Letzeres Element wird durch eine natürlichsprachliche Beschreibung erläutert (siehe Zeile 93).

Neben diesem Datenelement werden Name, Kontaktdaten und Geburtsdatum des Nutzers gespeichert. Falls der Nutzer ein Unternehmen oder eine Behörde ist, können die postalische Adresse und die Faxnummer ebenfalls angegeben werden.

5.2.4 Template für ein Dienstleistungsportal mit Sitzungscookies

Das Template für das Dienstleistungsportal wird im folgenden ergänzt um die Verwendung von Sitzungscookies zur Anpassung der Webseiten an die Nutzerbedürfnisse. Dazu wird das Statement, das sich mit Nutzungsdaten befasst und das aus dem Minimal-Template übernommen wurde, ergänzt (siehe Listing 5.4).

Dieses Template verwendet ausschließlich Sitzungscookies, d.h. Cookies, die nur für die Dauer einer Sitzung vorgehalten und beim Schließen des Browsers gelöscht werden. Diese Art des Einsatzes von Cookies widerspricht nicht dem TMG, da die Cookies mit dem Ende der Sitzung gelöscht und alle damit erhobenen Daten anonymisiert werden. Der Nutzer kann dadurch nicht bei seinem nächsten Besuch anhand des Cookies wiedererkannt werden.

Listing 5.4: Template für die Verwendung von Sitzungscookies

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <META ...>
3  <POLICY-REFERENCES>
4    <EXPIRY max-age="86400" />
5    <POLICY-REF about="#NameIhrerPolicy">
6      <INCLUDE>/*</INCLUDE>
7      <COOKIE-INCLUDE name="*" value="*" domain="*" path="
         *" />
8    </POLICY-REF>
9  </POLICY-REFERENCES>
10 <POLICIES>
11   <EXPIRY max-age="86400" />
12   <POLICY ...>
13     ...
14     <STATEMENT>
15       <CONSEQUENCE>
16         Wir protokollieren Zugriffe auf unser Webangebot
             in Protokolldateien. Diese Informationen
           werden 3 Monate lang gespeichert und danach
           automatisch gelöscht. Dabei werden
           Informationen, über die sich ein
           Personenbezug herstellen ließe, verkürzt und
           gefiltert gespeichert (siehe unsere
           Datenschutzerklärung). Wir verwenden
           Sitzungscookies, um unsere Webseiten an Ihre
           Bedürfnisse anpassen zu können.
17       </CONSEQUENCE>
18       <NON-IDENTIFIABLE/>
19       <PURPOSE>
```

```
20          <EXTENSION optional="yes">
21            <PPURPOSE xmlns="http://www.w3.org/2006/01/
                  P3Pv11">
22              <browsing/>
23            </PPURPOSE>
24          </EXTENSION>
25          <admin/>
26          <tailoring/>
27        </PURPOSE>
28        <RECIPIENT>
29          <EXTENSION>
30            <JURISDICTION
31                    service="http://www.gesetze-im-internet.
                        de/tmg/"
32                    short-description="Telemediengesetz (TMG
                        )">
33                    Telemediengesetz vom 26. Februar 2007 (
                        BGBl. I S. 179), Ausfertigungsdatum:
                        26.02.2007.
34              </JURISDICTION>
35          </EXTENSION>
36          <ours/>
37        </RECIPIENT>
38        <RETENTION>
39          <stated-purpose/>
40        </RETENTION>
41        <DATA-GROUP>
42          <EXTENSION>
43            <p3p11:DATA-GROUP>
44              ...
45            </p3p11:DATA-GROUP>
46          </EXTENSION>
47          <DATA ref="#dynamic.clickstream.uri"/>
48          <DATA ref="#dynamic.clickstream.timestamp"/>
49          <DATA ref="#dynamic.clickstream.other.bytes"/>
50          <DATA ref="#dynamic.clickstream.clientip.
                partialip"/>
51          <DATA ref="#dynamic.http"/>
52          <DATA ref="#dynamic.cookies">
53            <CATEGORIES>
54              <state/>
55            </CATEGORIES>
56          </DATA>
57        </DATA-GROUP>
58      </STATEMENT>
59    </POLICY>
```

```
60 </POLICIES>
61 </META>
```

Zunächst spiegelt sich der Einsatz von Cookies innerhalb des Elements POLICY-REF wieder. Das COOKIE-INCLUDE-Element definiert unter Verwendung von Wildcards für die enthaltenen Attribute, dass die zugehörige Policy für alle Cookies dieses Webangebots gilt (siehe Zeile 7).

Der Zweck, für den die Cookies eingesetzt werden, ist mit dem Wert `tailoring` (One-time Tailoring) des PURPOSE-Elements abgedeckt. Im Element DATA-GROUP kommt nun der Wert `dynamic.cookies` hinzu, der durch eine oder mehrere Kategorien beschrieben werden muss. Für die hier verwendeten Sitzungscookies wird die Kategorie `state` (State Management Mechanisms) verwendet (siehe Zeile 52).

5.2.5 Template für die Erstellung von Nutzerprofilen

Das TMG erlaubt Diensteanbietern, Nutzungsprofile für Zwecke der Werbung, der Marktforschung oder zur bedarfsgerechten Gestaltung der Telemedien zu erstellen. Diese Erlaubnis gilt unter der Bedingung, dass bei der Profilerstellung Pseudonyme verwendet werden. Die Profile dürfen nicht mit Daten über den Träger des Pseudonyms zusammengeführt werden. Die Erlaubnis besteht nur, solange der Nutzer dem nicht widerspricht. Auf das Widerspruchsrecht muss der Nutzer hingewiesen werden (vgl. § 15 Abs. 3 TMG). In § 15 TMG wird der Umgang mit Nutzungsdaten geregelt, so dass die Regelung zu Nutzungsprofilen in diesen Kontext einzuordnen ist.

Die Erstellung von Nutzerprofilen ist zwar unter den aufgeführten Bedingungen erlaubt, dürfte aber von datenschutzbewussten Nutzern abgelehnt werden und eher Misstrauen gegenüber dem Anbieter bewirken, da der Umgang mit ihren Daten hier nicht datensparsam gestaltet wird.

Listing 5.5: Template für die Erstellung von Nutzungsprofilen

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <META ...>
3  <POLICY-REFERENCES>
4      ...
5  </POLICY-REFERENCES>
6  <POLICIES>
7    <EXPIRY max-age="86400" />
8    <POLICY name="NameIhrerPolicy"
9      discuri="www.ihreBehoerde.de/datenschutzerklaerung.
           html"
10     opturi="www.ihreBehoerde.de/widerspruch.html"
11     xml:lang="de">
12     ...
13     <STATEMENT>
14       <CONSEQUENCE>
```

```
15          Wir protokollieren Zugriffe auf unser Webangebot
                in Protokolldateien. Diese Informationen
            werden 3 Monate lang gespeichert und danach
            automatisch gelöscht. Dabei werden
            Informationen, über die sich ein
            Personenbezug herstellen ließe, verkürzt und
            gefiltert gespeichert. Wir verwenden die
            protokollierten Daten zur Erstellung von
            Nutzungsprofilen. Hierfür werden Pseudonyme
            verwendet, die nicht mit personenbezogenen
            Daten zusammengeführt werden. Der Erstellung
            von Nutzungsprofilen können Sie widersprechen
             (siehe unsere Datenschutzerklärung).
16        </CONSEQUENCE>
17        <NON-IDENTIFIABLE/>
18        <PURPOSE>
19          <EXTENSION optional="yes">
20            <PPURPOSE xmlns="http://www.w3.org/2006/01/
                  P3Pv11">
21              <browsing/>
22              <marketing/>
23            </PPURPOSE>
24          </EXTENSION>
25          <admin/>
26          <tailoring/>
27          <develop required="opt-out"/>
28          <pseudo-analysis required="opt-out"/>
29        </PURPOSE>
30        <RECIPIENT>
31          <EXTENSION>
32            <JURISDICTION
33                    service="http://www.gesetze-im-internet.
                          de/tmg/"
34                    short-description="Telemediengesetz (TMG
                          )">
35                    Telemediengesetz vom 26. Februar 2007 (
                          BGBl. I S. 179), Ausfertigungsdatum:
                          26.02.2007.
36              </JURISDICTION>
37          </EXTENSION>
38          <ours/>
39        </RECIPIENT>
40        <RETENTION>
41          <stated-purpose/>
42        </RETENTION>
43        <DATA-GROUP>
```

```
44              <EXTENSION>
45                <p3p11:DATA-GROUP>
46                  ...
47                </p3p11:DATA-GROUP>
48              </EXTENSION>
49              <DATA ref="#dynamic.clickstream.uri"/>
50              <DATA ref="#dynamic.clickstream.timestamp"/>
51              <DATA ref="#dynamic.clickstream.other.bytes"/>
52              <DATA ref="#dynamic.clickstream.clientip.
                    partialip"/>
53              <DATA ref="#dynamic.http"/>
54            </DATA-GROUP>
55          </STATEMENT>
56        </POLICY>
57  </POLICIES>
58  </META>
```

Um das Minimal-Template im Hinblick auf die Erstellung von Nutzungsprofilen zu erweitern, wird das zweite Statement, welches sich mit den Nutzungsdaten befaßt, um die entsprechenden Zwecke ergänzt (vgl. Listing 5.5). Werbezwecke werden durch den Wert `marketing` (Advertising, Marketing, and/or Promotion) im PPURPOSE-Element abgebildet. Die bedarfsgerechte Gestaltung der Telemedien wird durch den Wert `pseudo-analysis` (Pseudonymous Analysis) des PURPOSE-Elements repräsentiert, Marktforschungszwecke sind durch den Wert `develop` (Research and Development) angegeben (ab Zeile 18).

Für die PURPOSE-Elemente, die für die Erstellung von Nutzungsprofilen hinzugefügt wurden, ist das Attribut *required* auf den Wert `opt-out` gesetzt, um darauf hinzuweisen, dass der Nutzer dieser Verwendung widersprechen kann. Daher muss das POLICY-Element im Attribut *opturi* einen URI angeben, der anzeigt, wo Instruktionen zu den Möglichkeiten des Widerspruchs zu finden sind (siehe Zeile 10).

Dass die erhobenen Daten nicht mit personenbezogenen Daten zusammengeführt werden, ist am Element NON-IDENTIFIABLE erkennbar (siehe Zeile 17). Im Element CONSEQUENCE wird der Nutzer über die Verwendung seiner Daten für diese Zwecke und seine Möglichkeit zum Widerspruch unterrichtet (siehe Zeile 14).

6 Bedienung des P3P-Policy-Generators

Dieses Kapitel beschreibt die Bedienung des Generators. Abschnitt 6.1 gibt einen Überblick über die Verzeichnisstruktur des Generators und erklärt, wie das Programm gestartet werden kann. Die Erstellung einer Policy mit Template und ohne Template beschreiben die Abschnitte 6.2 und 6.3. Abschnitt 6.4 enthält eine Anleitung zur Erstellung von Compact Policies, Abschnitt 6.5 erläutert, wie eine textuelle Übersicht über die Inhalte einer P3P-Datei erzeugt werden kann. Die Erstellung einer Policy-Referenzdatei mit dem Generator beschreibt Abschnitt 6.6.

6.1 Start des Programms

Der P3P-Policy-Generator ist eine Java-Applikation, die als JAR-Archiv vorliegt (*p3pgenerator.jar*). Um den Generator ausführen zu können, wird eine Java-Laufzeitumgebung ab Version 6 benötigt. Der Generator in seiner Gesamtheit wird als ZIP-Archiv ausgeliefert (*p3pPolicyGenerator.zip*). Dieses Archiv kann in ein beliebiges Verzeichnis entpackt werden. Es ist wie folgt strukturiert:

- data-Verzeichnis:
  - templates-Verzeichnis: Hier sind die Templates abgelegt (vgl. Abschnitt 5.2).
  - workspace-Verzeichnis: Hier können die vom Nutzer erzeugten Dateien abgelegt werden.
- lib-Verzeichnis: Hier sind die Software-Bibliotheken enthalten, die der Generator verwendet (vgl. Abschnitt 4.1).
- p3pgenerator.jar: Die kompilierten Quelldateien des Generators
- templates.properties: Properties-Datei mit Pfaden zu den Template-Dateien
- generator.properties: Properties-Datei mit Pfaden zu weiteren vom Generator benötigten Dateien
- P3Pv1.xsd: Definition der P3P-Version 1.0
- P3Pv11.xsd: Definition der P3P-Version 1.1
- P3Pv11BDS.xsd: Definition des P3P *Base Data Schema*

- xml.xsd: Definition des Namensraums `http://www.w3.org/XML/1998/namespace`
- XMLSchema.dtd: DTD für XML Schemas (Part 1: Structures)
- datatypes.dtd: DTD für XML Schemas (Part 2: Datatypes)
- readme.txt: Informationen zum Generator und den verwendeten Bibliotheken

Zum Starten des Programms dient die Ausführung des folgenden Befehls auf der Kommandozeile:

```
java -jar p3pgenerator.jar
```

Unter Windows kann das Programm auch mit einem Doppelklick auf die Datei *p3pgenerator.jar* gestartet werden.

6.2 Erstellen einer Policy mit Template

Abbildung 6.1: Das Menü „Datei“

Nach dem Start des Generators sind die verschiedenen Funktionalitäten des Generators über das „Datei“-Menü erreichbar (siehe Abb. 6.1). Um auf Basis eines Templates eine Policy zu erstellen, wählt man den Menüpunkt „Neu“ und in dessen Untermenü „Musterpolicy verwenden“.

Daraufhin erscheint ein Dialog, in dem das Template ausgewählt werden kann (siehe Abb. 6.2). Wird die Auswahlliste heruntergeklappt, sind alle verfügbaren Templates sichtbar.
Das Auswählen eines Templates führt zum ersten Bearbeitungsfenster, das zur Definition und Bearbeitung von Policies dient.

6.2.1 Bearbeitungsfenster „Policies“

Die Templates des Generators enthalten sowohl Policies als auch Policy-Referenzen. Policies enthalten Aussagen über Datenschutzpraktiken eines Anbieters. Policy-Referenzen definieren, für welche Bereiche eines Webangebots eine Policy gelten soll. Sie werden im Generator als „Geltungsbereiche“ bezeichnet.

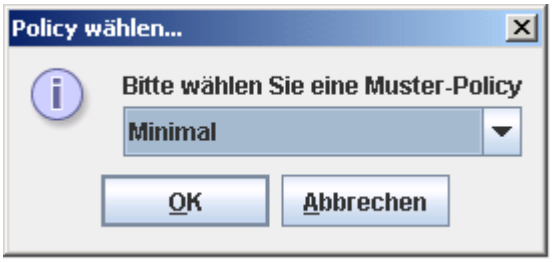

Abbildung 6.2: Dialog zur Auswahl von Templates

Das Bearbeitungsfenster „Policies“ zeigt eine tabellarische Übersicht über die enthaltenen Policies. Jede Policy ist durch einen eindeutigen Namen gekennzeichnet. Neben diesem Namen führt die Tabelle weitere Informationen zur Policy auf. Dazu zählen die URL der natürlichsprachlichen Datenschutzerklärung, der Geltungsbereich der Policy sowie die Anzeige des Bearbeitungszustands.

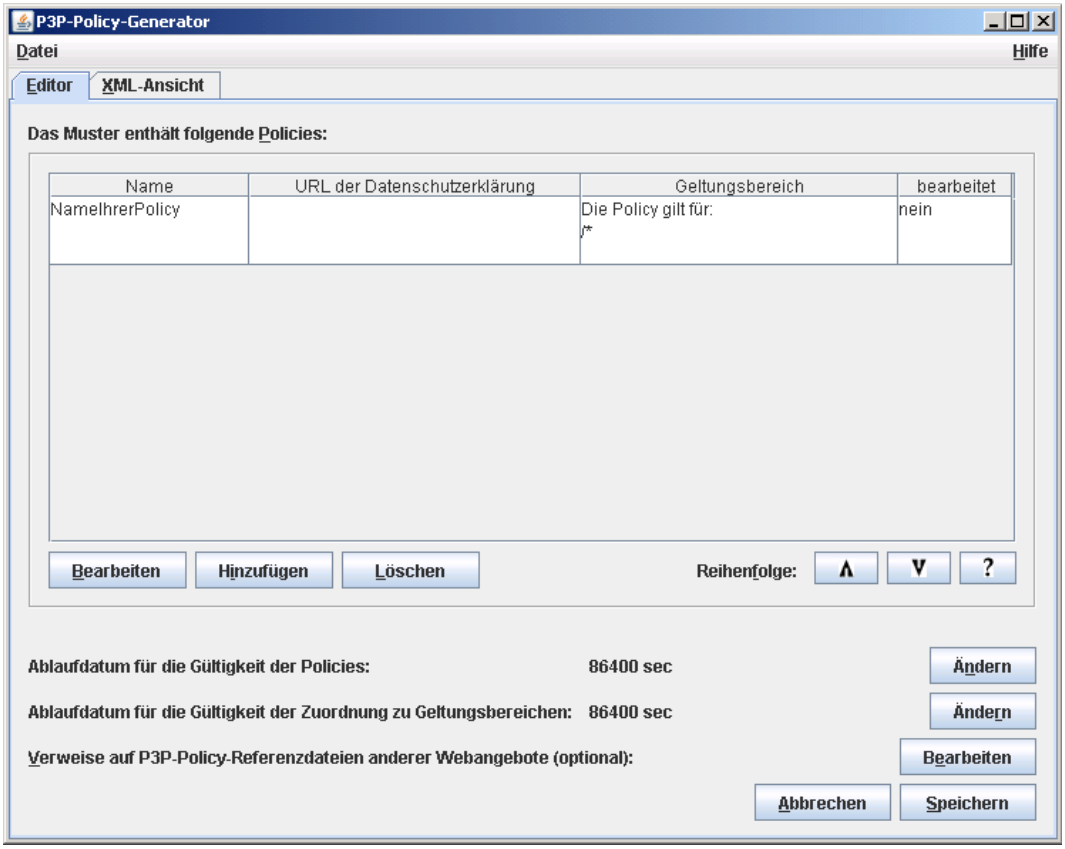

Abbildung 6.3: Fenster zur Bearbeitung von Policies

Abb. 6.3 zeigt den Startzustand nach Wahl des Templates „Minimal“. Dieses Muster enthält eine Policy, die noch nicht bearbeitet ist. Als Geltungsbereich wird hier die Standardeinstellung verwendet, das Kürzel „/*“ sagt aus, dass die Policy für alle Webseiten des Webangebots gilt. Dies kann in einem späteren

Bearbeitungsschritt vom Nutzer geändert werden. Ähnliches gilt für die URL der Datenschutzerklärung, diese muss vom Nutzer angegeben werden.

Die einzelnen Policies in der Tabelle können mit den darunter angeordneten Buttons bearbeitet werden. Der Button „Bearbeiten“ führt zum nächsten Bearbeitungsfenster, das zur Definition des Geltungsbereichs dient. Nach einem Durchlauf durch alle Bearbeitungsfenster der Policy erscheint erneut das vorliegene Übersichtsfenster aus Abb. 6.3. Wenn alle Policies bearbeitet sind, kann das Ergebnis als XML-Datei gespeichert werden. Dazu dient der Button „Speichern“.

Die Reihenfolge der Referenzen auf Policies kann mit den beiden Buttons geändert werden, die sich rechts unter der Tabelle befinden. Der linke Button verschiebt die selektierte Referenz um eins nach oben, der rechte Button verschiebt sie nach unten. Bei der Wahl der Reihenfolge sollte beachtet werden, dass die jeweiligen Geltungsbereiche von oben nach unten verarbeitet werden.

Zu einem gegebenen URI wird die erste Policy ausgewählt, deren Geltungsbereich mit diesem URI übereinstimmt, unabhängig davon, ob weiter unten in der Tabelle weitere Geltungsbereiche übereinstimmen würden.

Wenn z.B. Policy A für /info/* gilt, d.h. für alle Webseiten, die im Verzeichnis „info“ liegen, und Policy B für /info/aktuell/* gilt, dann sollte die Referenz auf Policy B über der Referenz auf Policy A stehen, damit sie zuerst gefunden wird. Andernfalls würde der URI *ihrHost*/info/aktuell/neuigkeiten.html der Policy A zugeordnet.

Im unteren Teil des Fensters kann eine Geltungsdauer sowohl für die Policies als auch für die Policy-Referenzen festgelegt werden. Für beide Angaben ist in den Templates der Standardwert von einem Tag (86400 Sekunden) eingestellt. Diese Dauer ist relativ zu dem Zeitpunkt zu verstehen, an dem die Datei vom Nutzer abgerufen wird. Für diese Dauer darf die Policy auf Nutzerseite im Cache gespeichert werden. Es wird empfohlen, den Standardwert von einem Tag beizubehalten, um möglichst einfach Änderungen an der Policy vornehmen zu können.

Bei der Wahl der Geltungsdauer sollte beachtet werden, dass mit einmal erhobenen Daten immer – ohne zeitliche Beschränkung – entsprechend der Policy verfahren werden muss, die zum Zeitpunkt der Erhebung gültig war. Wird eine neue Policy veröffentlicht, so gelten die neuen Bestimmungen nur für Daten, die unter dieser neuen Policy erhoben wurden. Falls z.B. eine relative Dauer von einer Woche gewählt wurde, muss nach Veröffentlichung einer neuen Policy ein Übergangszeitraum von über einer Woche gewahrt werden, während dem noch nach der alten Policy verfahren wird. Der Grund dafür ist, dass die Policy eine Woche lang von den Nutzern im Cache gehalten werden darf (vgl. [Cranor02a]).

Die Geltungsdauer für Policies bzw. Policy-Referenzen kann über den in Abb. 6.4 angezeigten Dialog geändert werden. Dieser Dialog ist über die „Ändern“-Buttons neben der Anzeige der Ablaufdaten erreichbar. Es kann entweder eine relative Dauer oder ein absolutes Ablaufdatum angegeben werden.

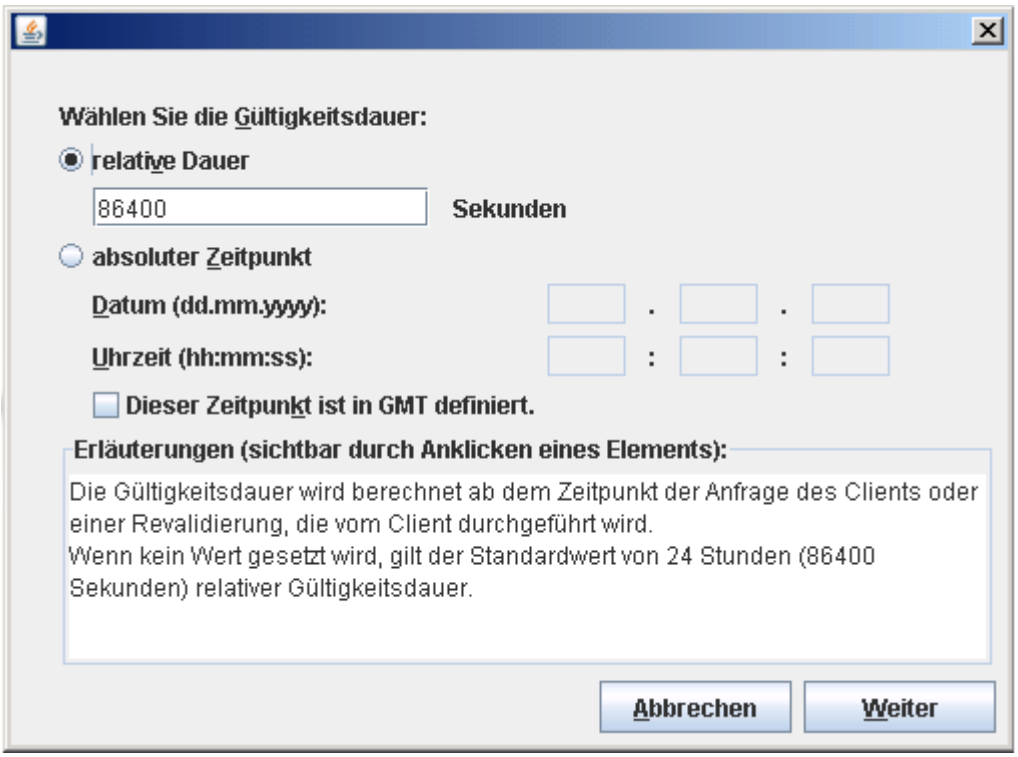

Abbildung 6.4: Dialog zur Definition der Geltungsdauer

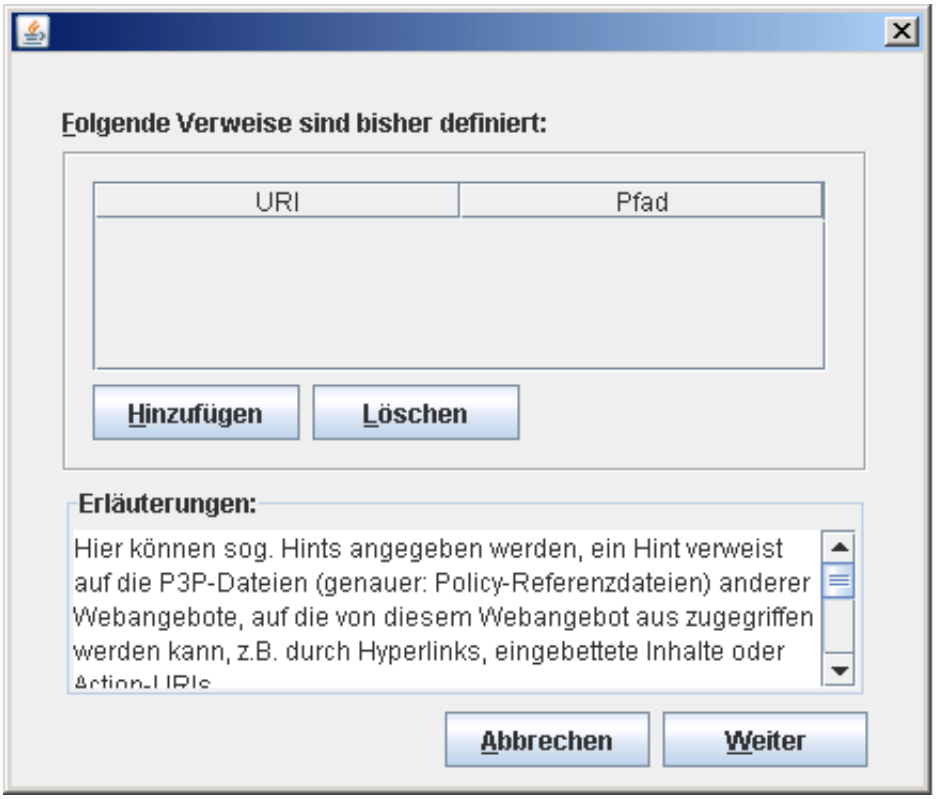

Abbildung 6.5: Dialog zur Definition von Hints

Ebenfalls im unteren Teil des Fensters können sog. Hints definiert werden. Ein Hint verweist auf die Policy-Referenzdatei eines anderen Webangebots, auf das durch Hyperlinks, eingebettete Inhalte oder Action-URIs zugegriffen werden kann. Der Dialog zur Definition von Hints ist in Abb. 6.5 zu sehen.

Im Feld URI wird der URI beschrieben, auf den der Hint verweist (z.B.: http://www.example.com). Dieser URI darf weder Pfade, noch Queries oder Fragmente enthalten. Die Wildcard * ist nur am Anfang des Host-Teils erlaubt (z.B.: http://*.example.com).

Im Feld Pfad wird angegeben, wo die Policy-Referenzdatei dieses Webangebots zu finden ist (z.B.: /w3c/prf.xml). Diese Angabe wird relativ zu dem im Feld URI angegebenen URI interpretiert, hier darf also kein absoluter URI angegeben werden.

Neben dem jeweils aktuellen Bearbeitungsfenster zeigt der Generator auch eine baumartige Ansicht des internen Datenmodells an. Diese Ansicht – die sog. „XML-View“ – enthält die P3P-Elemente mit den ihnen zugeordneten Werten (siehe Abb. 6.6).

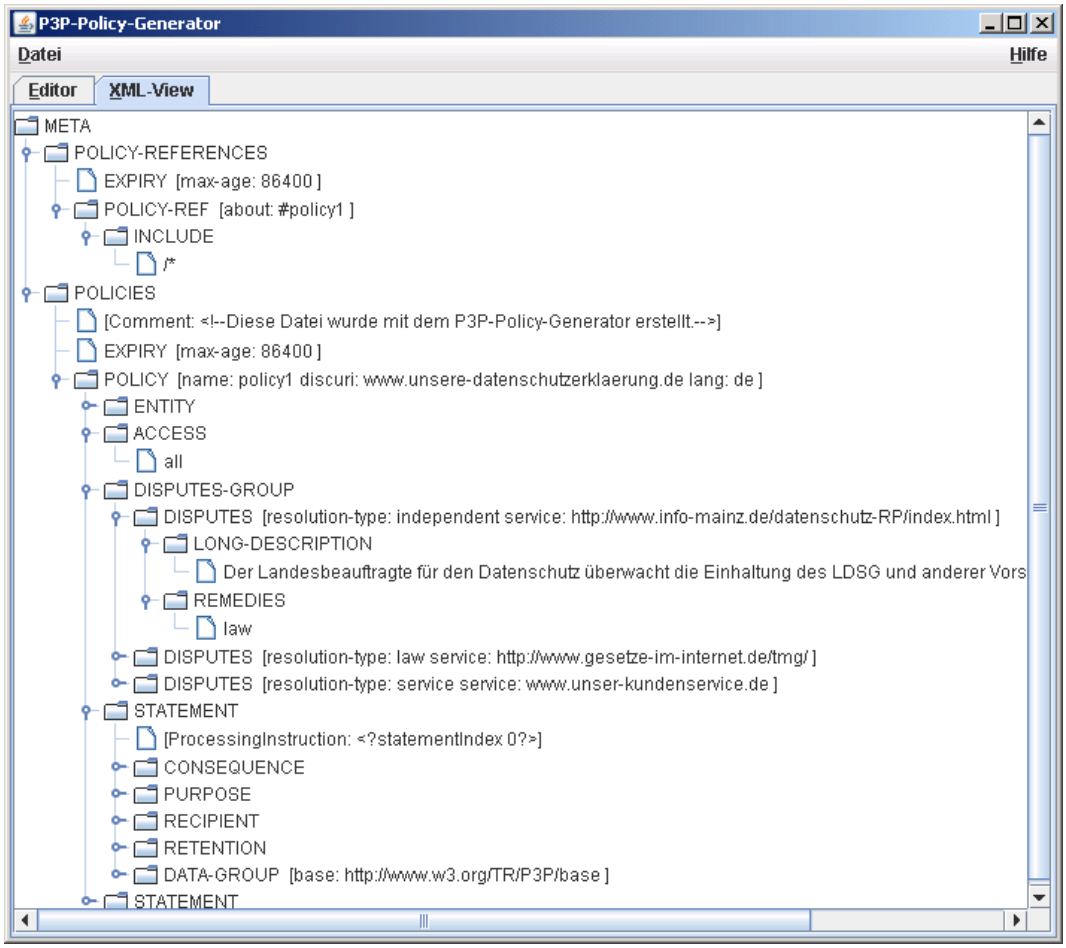

Abbildung 6.6: Die XML-View

6.2.2 Bearbeitungsfenster „Geltungsbereich“

Das Bearbeitungsfenster zur Definition des Geltungsbereichs einer Policy zeigt Abb. 6.7. Im oberen Bereich des Fensters kann der Policy ein eindeutiger Name gegeben werden. Zwingend wird die URL der natürlichsprachlichen Datenschutzerklärung abgefragt. Den einzelnen Elementen sind Hilfe-Buttons zugeordnet, die ein Fenster mit Erläuterungen zu diesem Element öffnen. Diese Buttons sind mit einem Fragezeichen gekennzeichnet.

Im rechten Teil des Fensters ist die aktuelle Konfiguration des Geltungsbereichs zu sehen. Diese Übersicht spiegelt die Änderungen wider, die in der linken Hälfte des Fensters gemacht werden. Hier können URIs in den Geltungsbereich einbezogen oder von diesem ausgeschlossen werden. Die URIs müssen dabei relativ zu dem Host angegeben werden, auf dem die P3P-Datei abgelegt wird. Analog zu URIs können auch Cookies ein- bzw. ausgeschlossen werden. Für beide kann die Wildcard * verwendet werden, um nicht einzelne Elemente, sondern Mengen zu definieren. Mit der relativen URL /* ist beispielsweise das gesamte Webangebot gemeint, mit /*.pdf alle PDF-Dateien des Webangebots (vgl. [Cranor02a]).

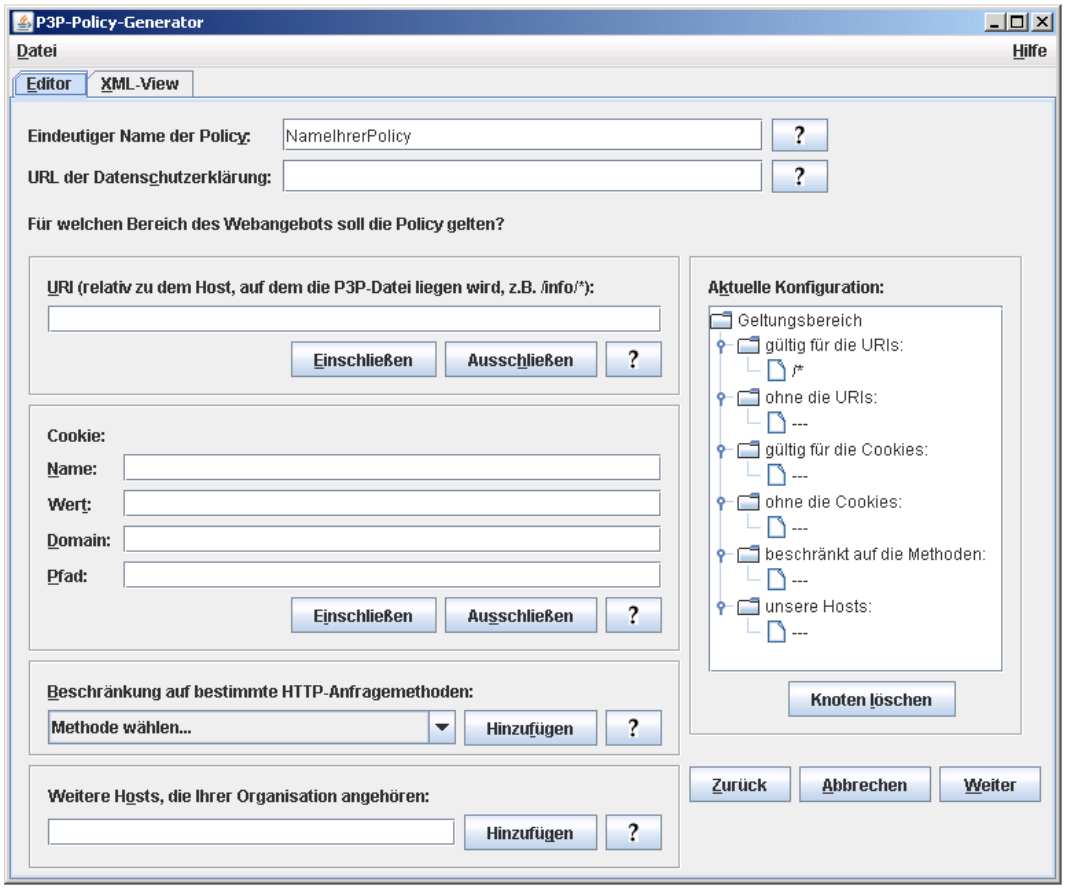

Abbildung 6.7: Fenster zur Bearbeitung des Geltungsbereichs

Bei der Definition von Policies für Cookies sollte folgendes beachtet werden (vgl. [Wenning06]):

- Für Cookies sollte eine Compact Policy erstellt werden. Diese muss in dem HTTP-Antwort-Header, der das Cookie setzt, plaziert werden.
- Falls mit dem Cookie eine Compact Policy gesetzt wird, so muss die Lebensdauer der vollständigen Policy ebenso lang sein, wie die des Cookies.
- Falls ein Cookie zu mehreren Hosts innerhalb einer Domain zurückgespielt wird, müssen alle diese Hosts die dem Cookie zugeordnete Policy befolgen.

Die Gültigkeit einer Policy kann auch auf HTTP-Anfragen beschränkt werden, die eine bestimmte Methode benutzen. Standardmäßig wird eine Referenz unabhängig von der Zugriffsmethode auf eine Ressource angewendet. Dies ist in den meisten Fällen ausreichend. Ein Beispiel für den Einsatz dieser Beschränkung wäre, dass ein Anbieter mehr Daten erheben möchte, wenn eine DELETE-Anfrage ausgeführt wird, als bei einer GET-Anfrage (vgl. [Wenning06]).

Ein Anbieter kann weitere Hosts angeben, die zu seiner Organisation gehören oder als Dienstleister für diese Organisation agieren. Diese Hosts können im linken unteren Teil des Bearbeitungsfensters eingegeben werden. Eine solche Angabe kann von Nutzertools verwendet werden, um zwischen solchen Hosts und Hosts Dritter zu unterscheiden.

Über den Button „Weiter“ gelangt man zum nächsten Bearbeitungsfenster, das das Anbieterkennzeichen abfragt. Der „Zurück“-Button führt zum „Policies“-Fenster.

6.2.3 Bearbeitungsfenster „Anbieterkennzeichen“

Das Bearbeitungsfenster „Anbieterkennzeichen“ fragt alle Angaben ab, die für eine Kontaktaufnahme des Nutzers mit dem Anbieter benötigt werden (siehe Abb. 6.8). Die Angabe einer Faxnummer und einer URL sind optional.

6.2.4 Bearbeitungsfenster „Auskunft und Konfliktlösung“

Zum Auskunftsrecht des Nutzers bietet das nächste Bearbeitungsfenster mehrere vordefinierte Elemente an. Wird eines dieser Elemente selektiert, werden im darunter liegenden Bereich Erläuterungen dazu angezeigt (siehe Abb. 6.9).

Der untere Teil des Fensters dient zur Definition von Verfahren zur Konfliktlösung. Die Tabelle listet die im Template vorhandenen Verfahren auf. Die hier vorhandenen Einträge sollten selektiert und vervollständigt werden. Dazu dient der Button „Ändern“, der den Dialog zur Bearbeitung eines Eintrags öffnet (siehe Abb. 6.10). In diesem Dialog können Anlaufstellen zur Lösung von Streitigkeiten genannt und beschrieben werden. Der rechte Teil des Dialogs zeigt Erläuterungen zu den wählbaren Elementen.

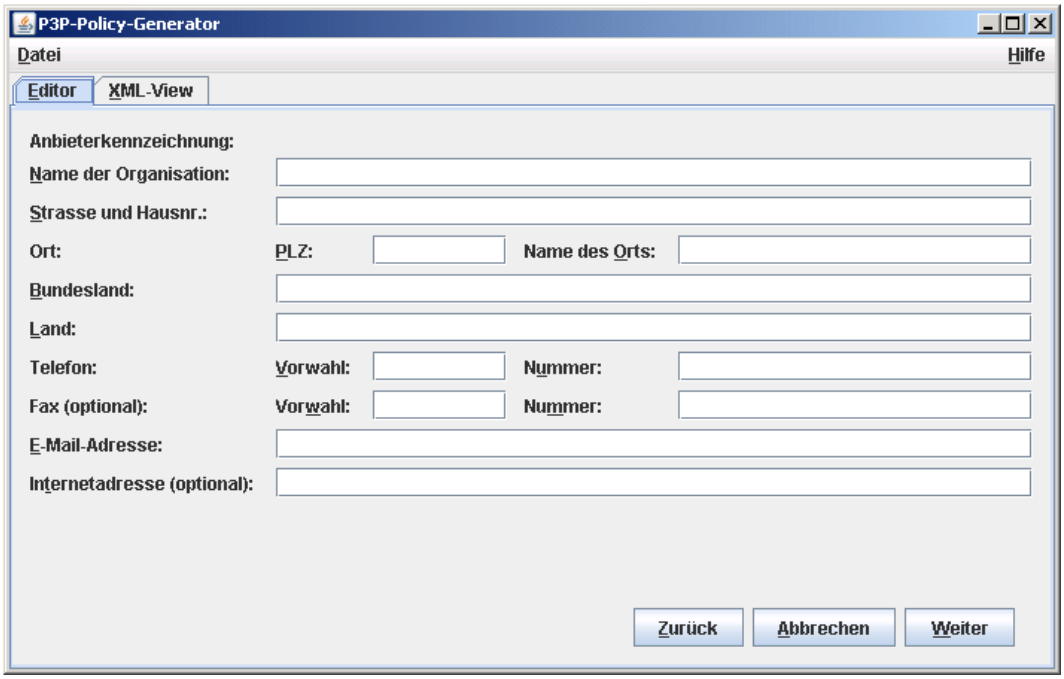

Abbildung 6.8: Fenster zur Angabe des Anbieterkennzeichens

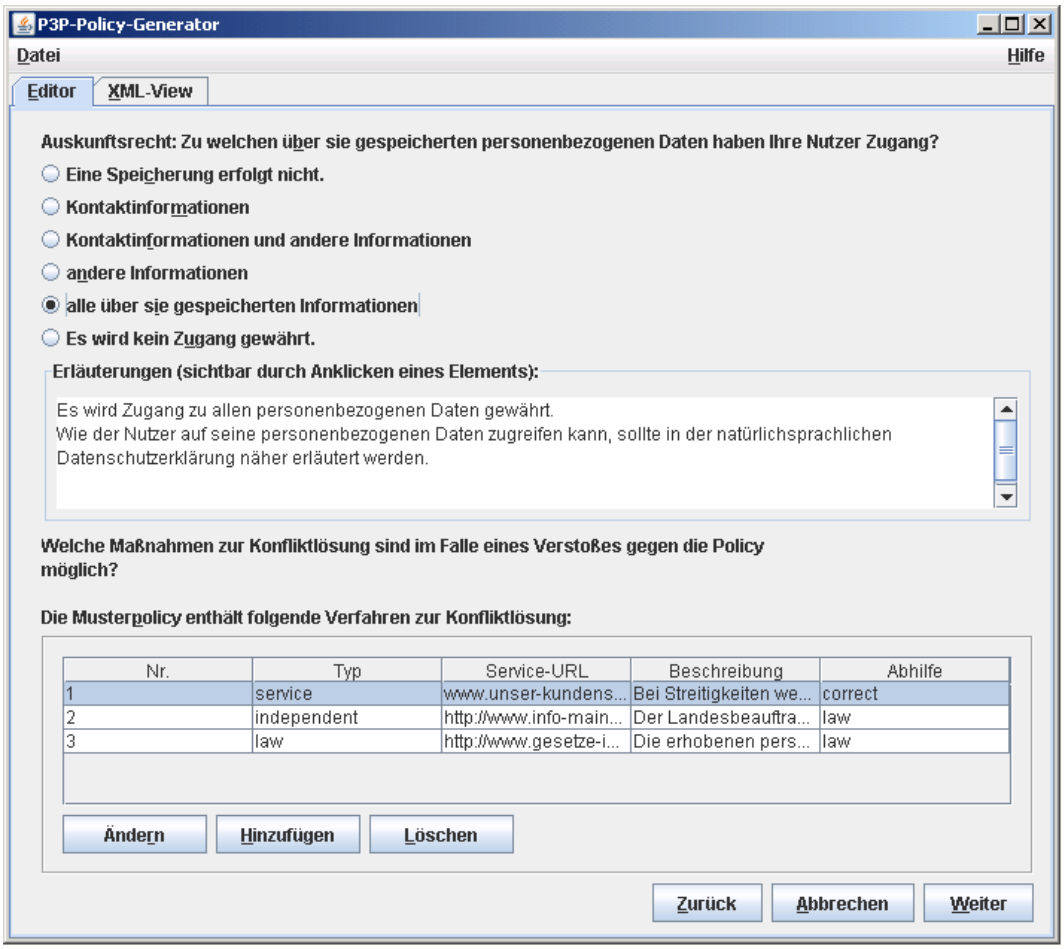

Abbildung 6.9: Fenster zur Bearbeitung von Auskunft und Konfliktlösung

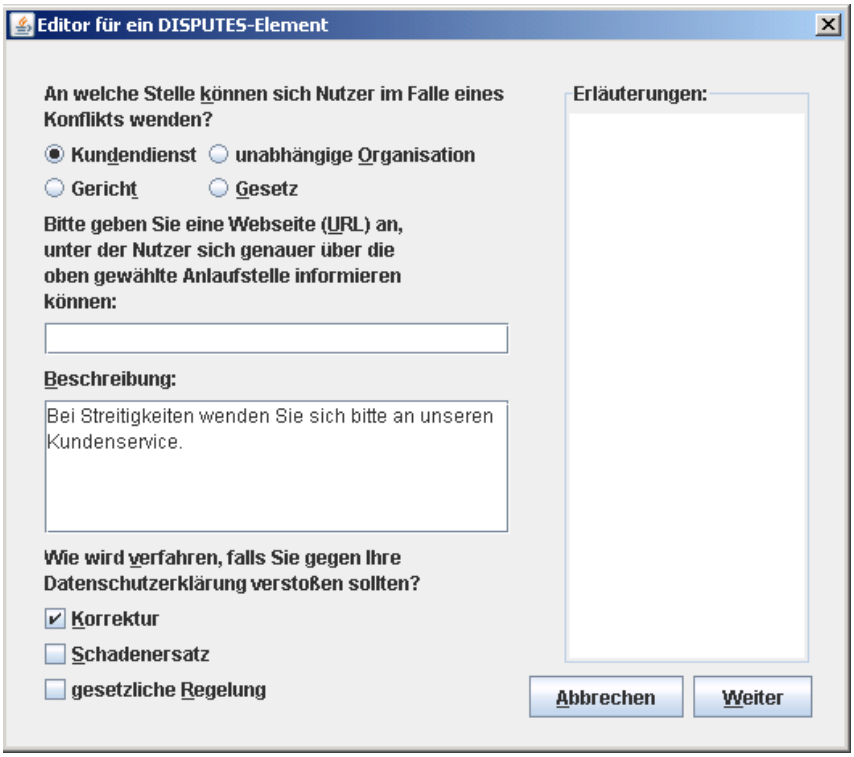

Abbildung 6.10: Dialog zur Angabe von Anlaufstellen zur Konfliktlösung

6.2.5 Bearbeitungsfenster „Statements“

Ein Statement repräsentiert das Datenschutzverhalten des Anbieters im Bezug auf eine bestimmte Gruppe von Daten. Beispielhaft für solche Gruppen sind Daten für administrative Zwecke (wie die Logdatei eines Web-Servers) oder Daten, die für die Zustellung einer Bestellung benötigt werden.

Innerhalb eines Statements wird definiert, welche Daten der Anbieter speichert, für welche Zwecke diese Daten verwendet werden, an wen sie weitergegeben werden und wie lange die Daten gespeichert werden.

Das Fenster zur Bearbeitung von Statements zeigt im oberen Teil eine Übersicht über die im Template vorhandenen Statements (siehe Abb. 6.11). Die darunter liegenden Buttons dienen zur Manipulation der Tabelle. Der Button „zu Gruppe zuordnen“ dient zur Verknüpfung eines Statements mit einer Statement-Gruppe. Ein Nutzertool kann Statements, die zur selben Gruppe gehören, unter einer gemeinsamen Überschrift anzeigen.

Die Verwendung von Statement-Gruppen ist optional, eine Auflistung der definierten Gruppen ist im linken unteren Teil des Fensters zu sehen. Hier kann über den Button „Hinzufügen“ eine neue Gruppe angelegt werden. Dazu öffnet sich der Dialog aus Abb. 6.12. Neben einem Namen und einer Beschreibung der Gruppe kann hier angegeben werden, ob der Nutzer die Möglichkeit hat, alle Statements dieser Gruppe geschlossen zu akzeptieren bzw. abzulehnen. Diese Aussage bezieht sich auf Elemente, für die definiert werden kann, ob sie für die Nutzung des Angebots erforderlich sind, oder ob der Nutzer dieser speziellen Verwendung seiner Daten explizit zustimmen oder diese explizit ablehnen kann

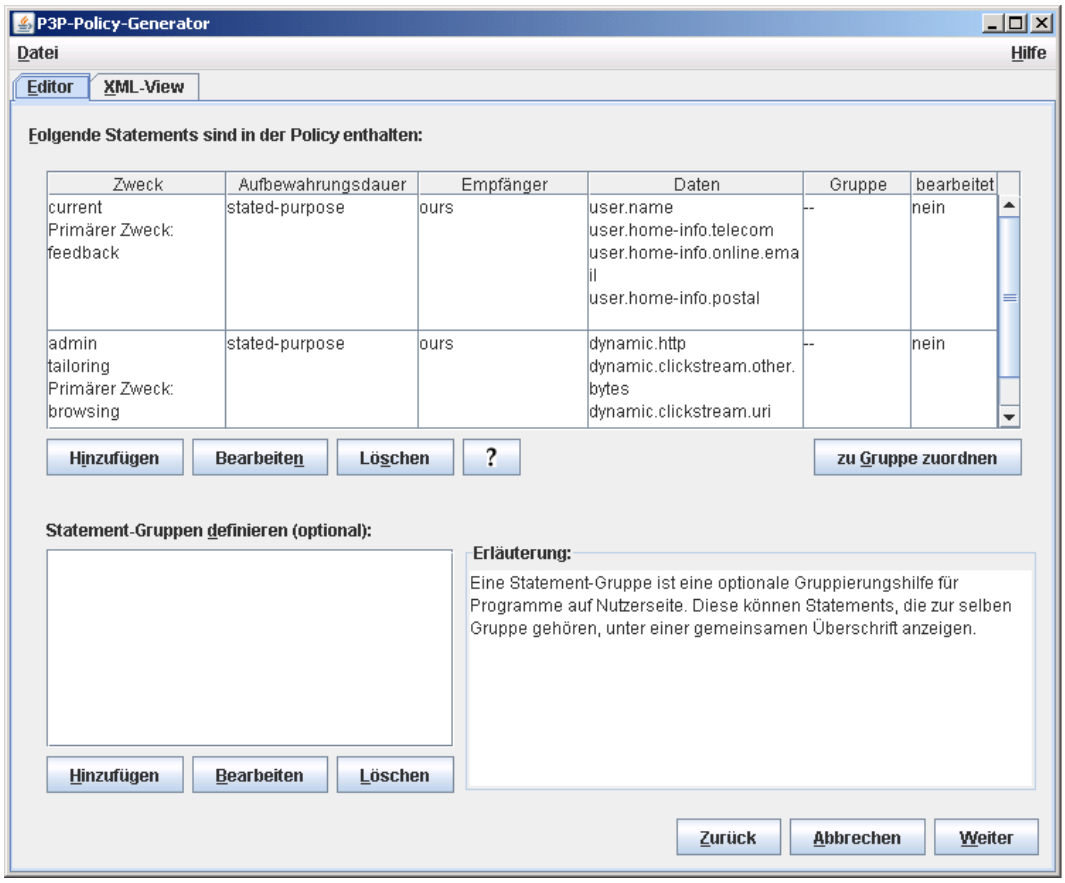

Abbildung 6.11: Fenster zur Bearbeitung von Statements

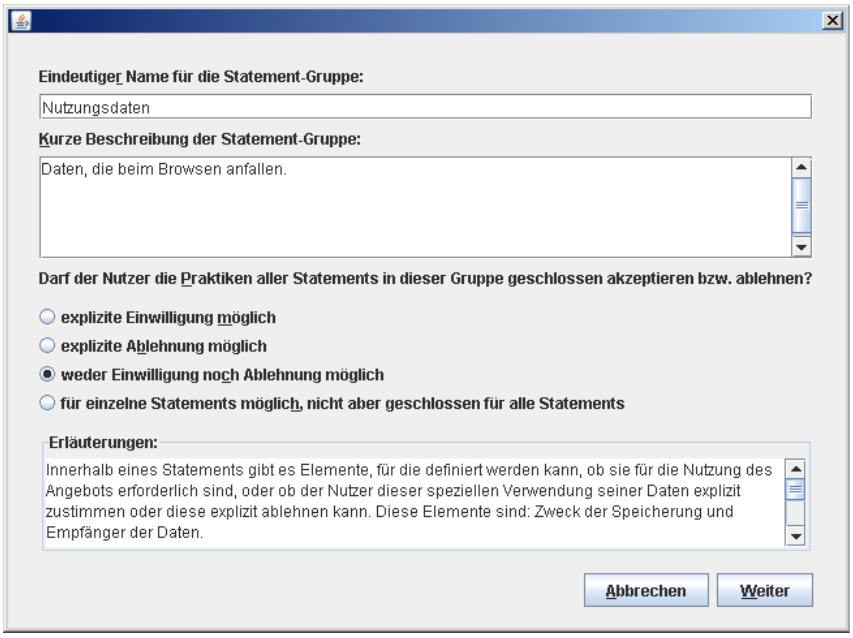

Abbildung 6.12: Dialog zur Definition von Statement-Gruppen

(„opt-in“ bzw. „opt-out“). Diese Elemente sind: Zweck der Speicherung und Empfänger der Daten. Falls der Nutzer diese Entscheidung gleichzeitig für alle Statements in dieser Gruppe treffen kann, wird dies hier definiert.

Die Bearbeitung eines Statements wird über den Button „Bearbeiten“ unter der Statement-Tabelle gestartet. Dieser Button führt zum nächsten Bearbeitungsfenster, in dem eine Zusammenfassung des Statements angegeben werden kann. Nachdem alle Fenster zur Bearbeitung des Statements durchlaufen wurden, erscheint wieder das vorliegende Bearbeitungsfenster (siehe Abb. 6.11). Wenn alle Statements bearbeitet sind, führt der Button „Weiter“ zurück zum ersten Bearbeitungsfenster, das eine Übersicht über die Policies anzeigt. Die gerade durchlaufene Policy wird dort dann als bearbeitet markiert.

6.2.6 Bearbeitungsfenster „Zusammenfassung eines Statements“

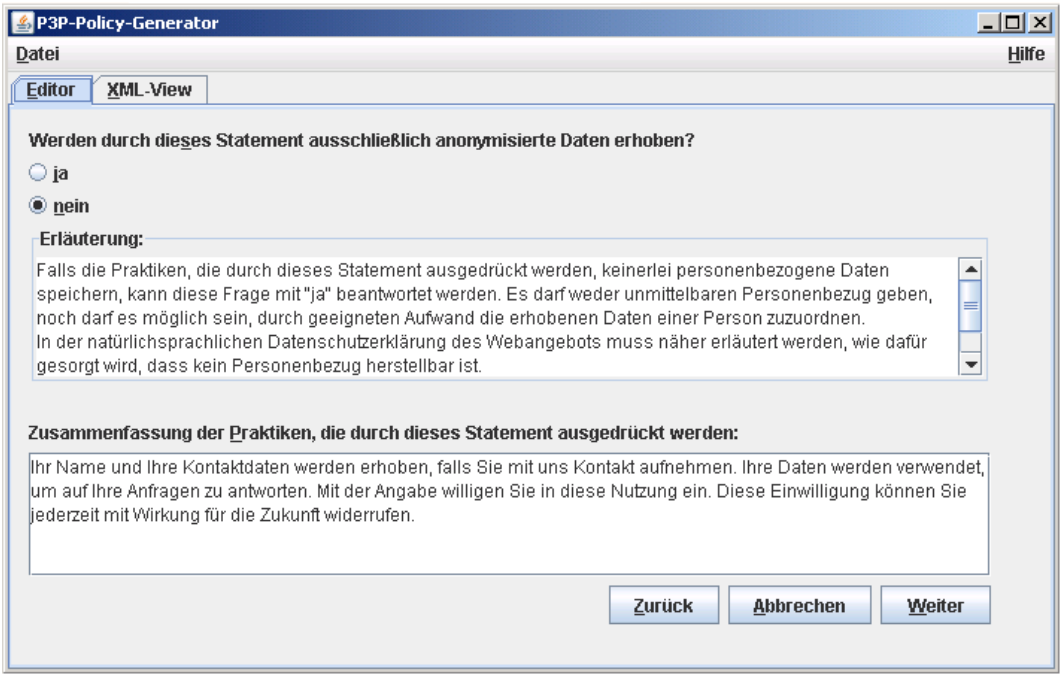

Abbildung 6.13: Fenster zur Zusammenfassung eines Statements

Für das Statement, das im vorangehenden Fenster zur Bearbeitung ausgewählt wurde, kann nun eine natürlichsprachliche Beschreibung angegeben werden (siehe Abb. 6.13). Im unteren Teil des Fensters ist die Beschreibung eingeblendet, die das Template für dieses Statement vorschlägt. Bei Bedarf sollte der Text angepaßt oder um zusätzliche Informationen ergänzt werden.

Im oberen Teil des Fensters muss der Nutzer angeben, ob das Statement ausschließlich anonymisierte Daten erhebt, oder ob ein Personenbezug hergestellt werden kann. Werden nur anonymisierte Daten erhoben, so muss die natürlichsprachliche Datenschutzerklärung darlegen, wie die Anonymisierung sichergestellt wird.

6.2.7 Bearbeitungsfenster „Zweck der Datenerhebung“

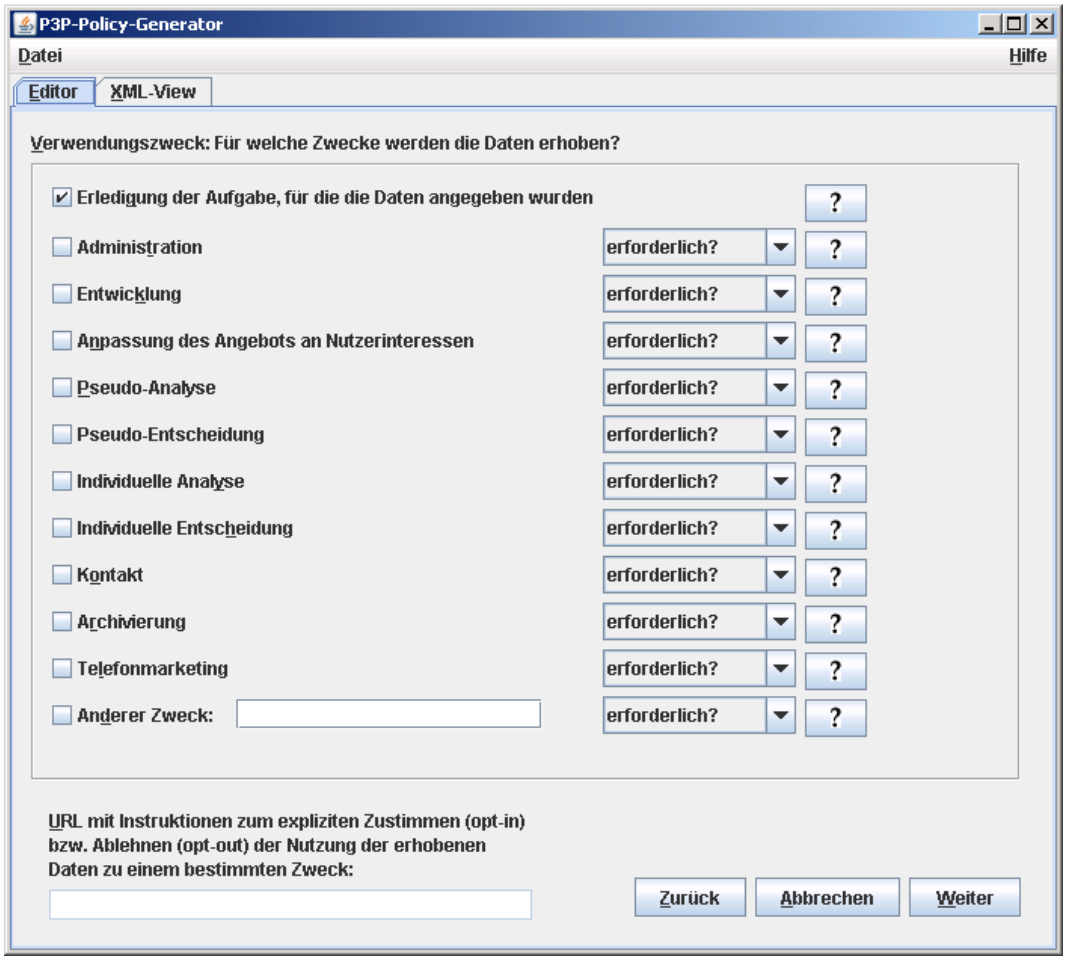

Abbildung 6.14: Fenster zur Angabe der Verwendungszwecke

Das Bearbeitungsfenster zur Angabe der Verwendungszwecke bietet mehrere vordefinierte Elemente (siehe 6.14). Die durch das Template vorgeschlagenen Elemente sind selektiert. Wie die einzelnen Elemente definiert sind, ist durch die Hilfe-Buttons abrufbar. Mit Ausnahme des ersten Elements („Erledigung der Aufgabe, für die die Daten angegeben wurden“) kann für alle Elemente eine Angabe zur Erforderlichkeit gemacht werden. Dafür stehen die Werte „immer“, „opt-in“ und „opt-out“ zur Auswahl. Wird kein Wert angegeben, so ist „immer“ der Standardwert. Der Wert „opt-in“ wird verwendet, um die Möglichkeit einer expliziten Einwilligung in diesen Verwendungszweck auszudrücken. Kann der Nutzer explizit einem Verwendungszweck widersprechen, so wird „opt-out“ verwendet. Für „opt-in“- oder „opt-out“-Verfahren muss eine URL angegeben werden, unter der der Nutzer eine entsprechende Anleitung finden kann.

Falls keiner der vorgegebenen Zwecke passen sollte, kann das Element „Anderer Zweck“ verwendet werden, das im nebenstehenden Textfeld beschrieben werden muss.

6.2.8 Bearbeitungsfenster „Primärer Zweck der Datenerhebung“

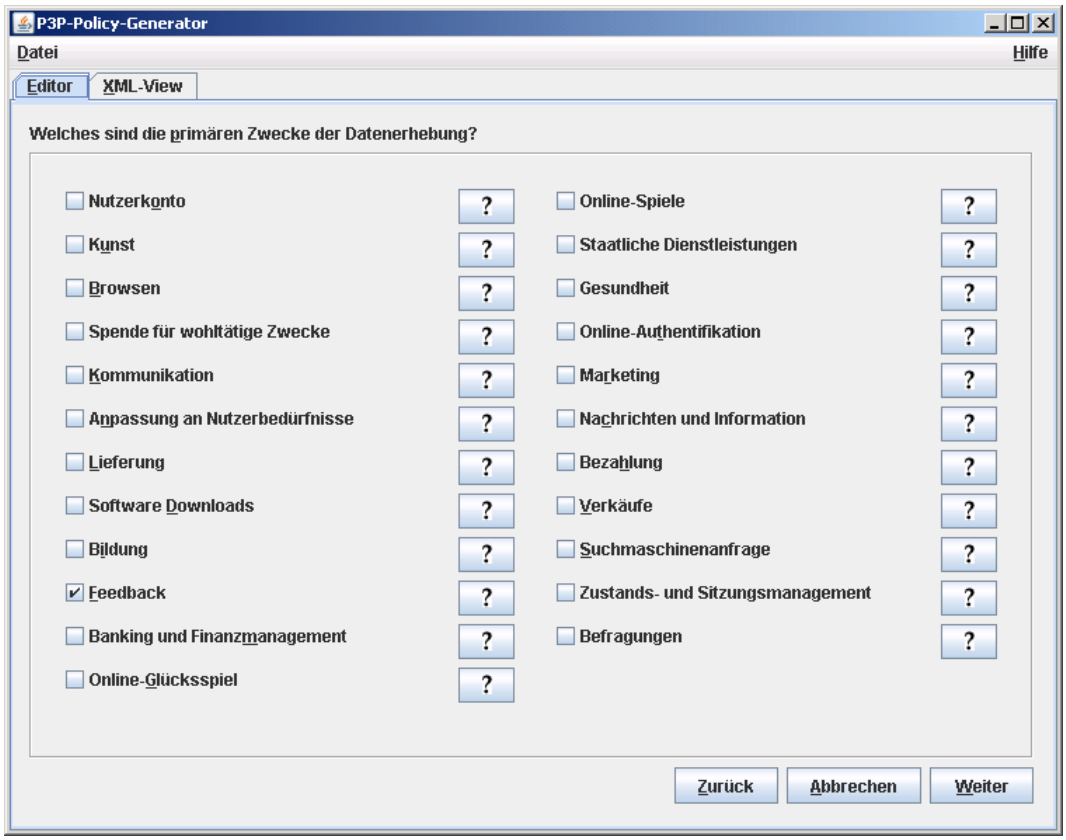

Abbildung 6.15: Fenster zur Angabe der primären Zwecke

Dieses Fenster ist in Abb. 6.15 zu sehen und dient zur Angabe des primären Zwecks der Datenerhebung. So sollen die bisher angegebenen Zwecke detaillierter beschrieben werden. Auch hier sind ein oder mehrere Elemente wählbar. Erläuterungen zu den einzelnen Elementen sind über die Hilfe-Buttons verfügbar.

6.2.9 Bearbeitungsfenster „Löschungsregelung und Empfänger“

Das Fenster zur Angabe von Löschungsregelungen und Empfängern zeigt Abb. 6.16. Die Aufbewahrungsdauer der erhobenen Daten kann im oberen Teil dieses Fensters angegeben werden. Eine Empfehlung wird durch das Template vorgegeben. Gleiches gilt für die Angabe der Empfänger der Daten. Hier kann zu den ausgewählten Elementen eine Beschreibung angegeben werden. Eine Erläuterung der einzelnen Elemente ist über Hilfe-Buttons verfügbar.

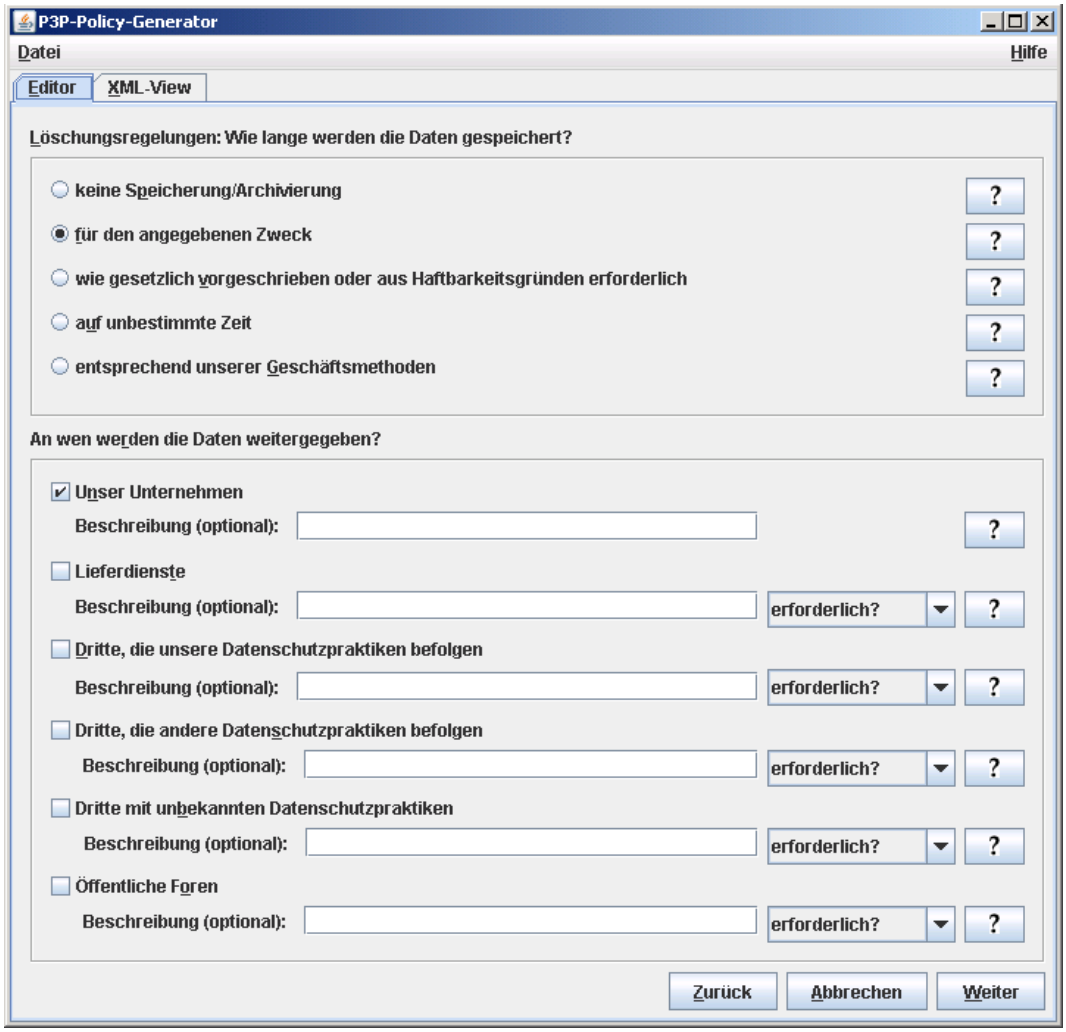

Abbildung 6.16: Fenster zur Angabe von Löschungsregelungen und Empfängern

6.2.10 Bearbeitungsfenster „Datentypen“

Das Fenster zur Definition der Daten, die durch ein Statement erhoben werden, ist in Abb. 6.17 zu sehen. Im linken Teil des Fensters können Datentypen ausgewählt werden. Die verfügbaren Datentypen werden in einer baumartigen Darstellung angezeigt und entsprechen dem P3P *Base Data Schema*. Die Hierarchie dieses Schemas enthält vier Elemente, die durch Unterelemente weiter präzisiert werden sollten:

- *dynamic*: Dieses Element dient zur Beschreibung von Daten, die keine festen Werte haben. Hier sind beispielsweise Cookies, HTTP-Referer-Header oder Suchanfragen, die der Nutzer in eine Suchmaschine eingibt, enthalten.
- *user*: Mit diesem Element werden Informationen über den Nutzer modelliert, dies können z.B. Name, Geburtsdatum oder Kontaktangaben sein.
- *third-party*: Dieses Element wird verwendet, wenn Daten über Dritte erhoben werden, ein Beispiel hierfür könnte sein, dass ein Nutzer über sich und über den Ehepartner Angaben machen muss.
- *business*: Mit diesem Element werden Informationen über juristische Personen angegeben, hier können beispielsweise der Name der Organisation oder Kontaktangaben angegeben werden.

Die vom Template vorgegebenen Datentypen werden in der Übersicht über die getroffene Auswahl in der rechten Hälfte des Fensters dargestellt. Hier kann für jeden Datentyp festgelegt werden, ob seine Erhebung optional ist. Um einen weiteren Datentyp zu dieser Auflistung hinzuzufügen, selektiert man in der Baumansicht ein Element und betätigt den Button in der Mitte des Fensters, der einen Pfeil nach rechts anzeigt. Der Button mit dem Pfeil nach links entfernt ein Element aus der Auswahlliste.

Die meisten Datentypen des *Base Data Schema* sind mit festen, nicht veränderbaren Kategorien verknüpft. Zwei Elemente erfordern jedoch, dass der Nutzer ihnen Kategorien zuordnet: `dynamic.cookies` und `dynamic.miscdata`. Letzteres kann verwendet werden, wenn der gewünschte Datentyp nicht im *Base Data Schema* definiert ist. Wird eines der beiden Elemente ausgewählt, so öffnet sich der Dialog zur Auswahl von Kategorien (siehe 6.18).

In diesem Dialog muss mindestens eine Kategorie angegeben werden, die beschreibt, welche Datentypen durch das gewählte Element erhoben werden. Wenn keine der vorgegebenen Kategorien passend erscheint, sollte die Kategorie „Andere Informationen“ verwendet werden, die eine individuelle Beschreibung der Kategorie ermöglicht.

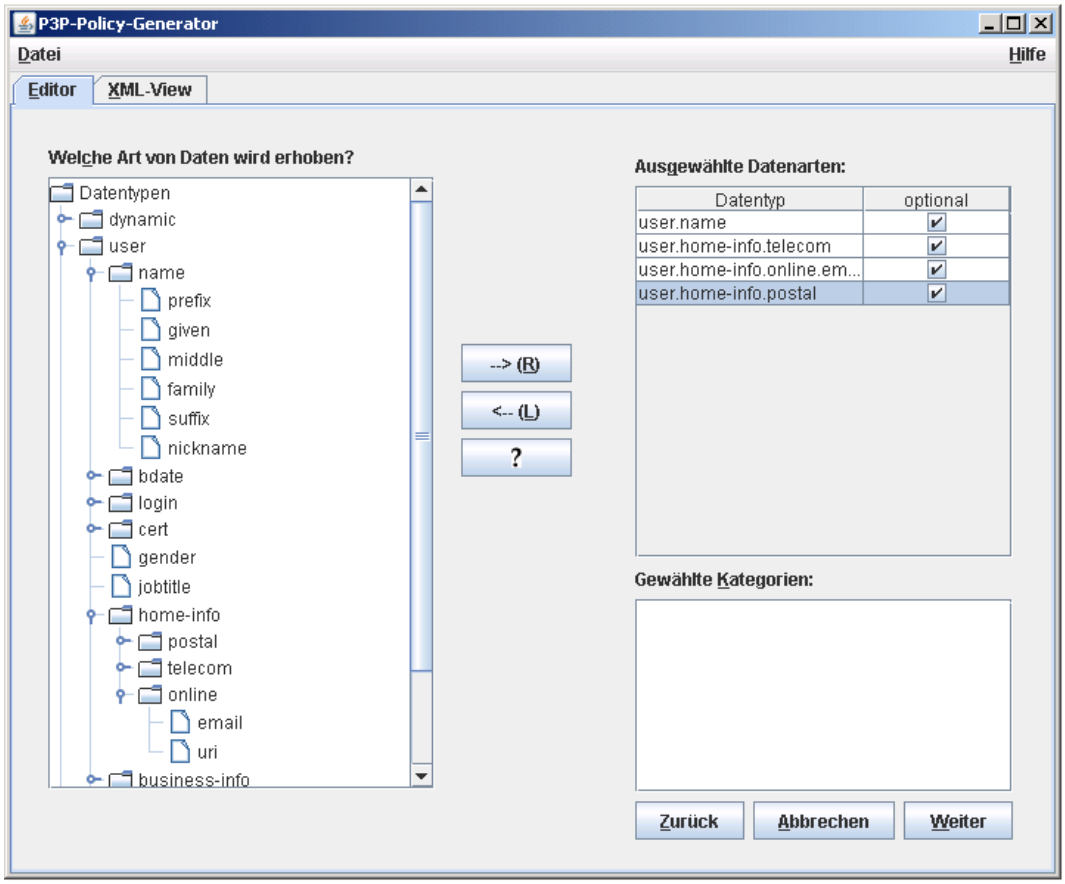

Abbildung 6.17: Fenster zur Definition der Datentypen

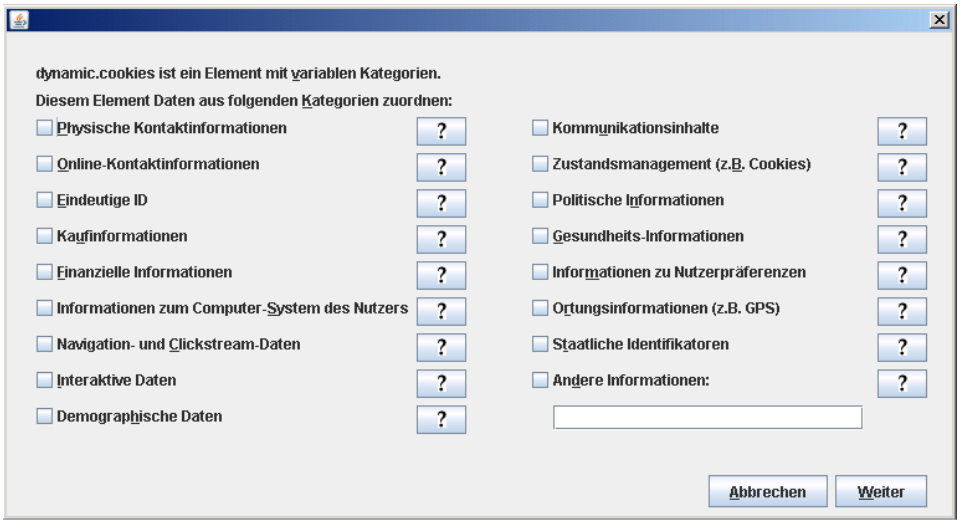

Abbildung 6.18: Dialog zur Auswahl von Kategorien

6.2.11 Veröffentlichung der P3P-Datei

Die mit dem Generator erstellte XML-Datei kann auf verschiedene Arten veröffentlicht werden. Die einfachste und vom W3C empfohlene Methode ist das Ablegen der Datei unter dem Pfad */w3c/p3p.xml*. Für das Webangebot *http://www.beispiel.de* wäre die vollständige Adresse für die P3P-Datei also *http://www.beispiel.de/w3c/p3p.xml*. Diese sog. „well-known location" ist der erste Ort, an dem ein P3P-Nutzertool nach der Datei sucht. Alternative Möglichkeiten zur Veröffentlichung einer P3P-Datei werden in Abschnitt 2.3 erläutert.

6.2.12 Ändern einer P3P-Datei

Soll eine vollständige P3P-Datei im Nachhinein geändert werden, so kann die Datei mit dem Menüpunkt „Öffnen" des „Datei"-Menüs in den Generator geladen werden. Sie kann auf dieselbe Art durchlaufen werden wie ein Template. Die in der Quelldatei ausgewählten Elemente werden in den Bearbeitungsfenstern selektiert angezeigt.

6.3 Erstellen einer Policy ohne Template

Die Erstellung einer Policy ohne Template durchläuft dieselben Arbeitsschritte wie die in Abschnitt 6.2 geschilderte Erstellung mit Template. In den einzelnen Bearbeitungsfenstern sind jedoch keine Empfehlungen zu sehen, die Auswahl der Elemente liegt vollständig beim Benutzer.

6.4 Erstellen von Compact Policies

Der Generator kann zu einer P3P-Datei Compact Policies erstellen. Für jede in der Quelldatei enthaltene Policy wird eine Compact Policy generiert. Diese Funktionalität ist über den Menüpunkt „Compact Policy erstellen" des „Datei"-Menüs erreichbar. Nach Auswahl dieses Menüpunkts wird ein Dateiauswahl-Dialog angezeigt, in dem die Quelldatei angegeben werden kann. Der Generator liest diese Datei ein, generiert zu jeder Policy eine Compact Policy und zeigt das Ergebnis in einem Bearbeitungsfenster an (siehe Abb. 6.19). Hier können die Compact Policies über den „Speichern"-Button in einer Textdatei abgespeichert werden. Näheres zu Compact Policies enthält Abschnitt 2.5.

Eine Compact Policy wird einem Cookie zugeordnet, indem sie in den P3P-HTTP-Header gesetzt wird (siehe Abschnitt 2.3). Falls mehrere Cookies durch eine HTTP-Antwort gesetzt werden, muss die Compact Policy für alle diese Cookies gelten (vgl. [Cranor02a]).

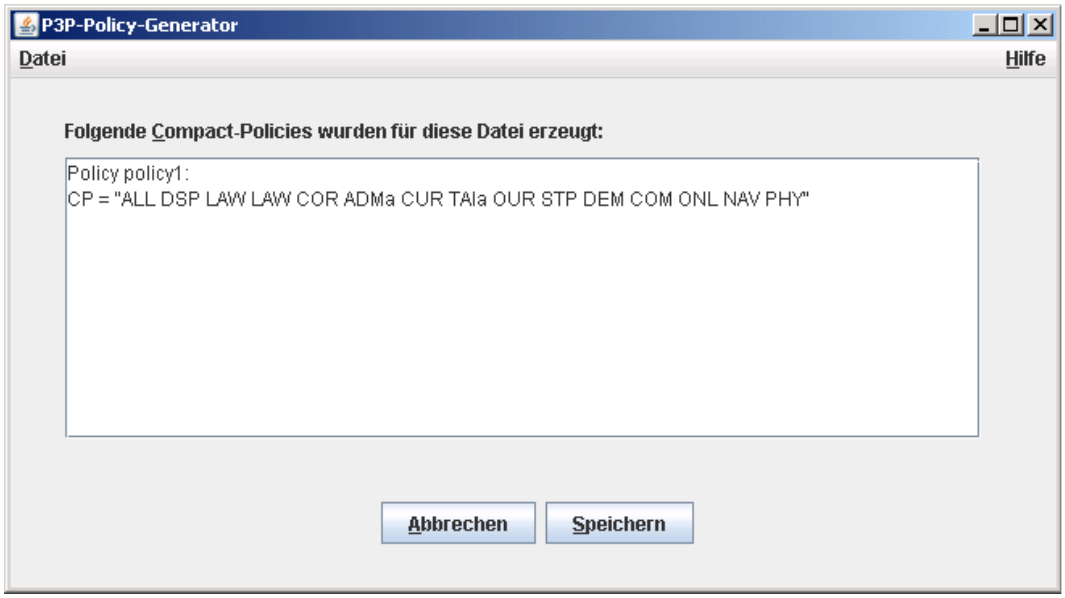

Abbildung 6.19: Anzeigefenster für Compact-Policies

6.5 Erstellen einer Übersichtsdatei

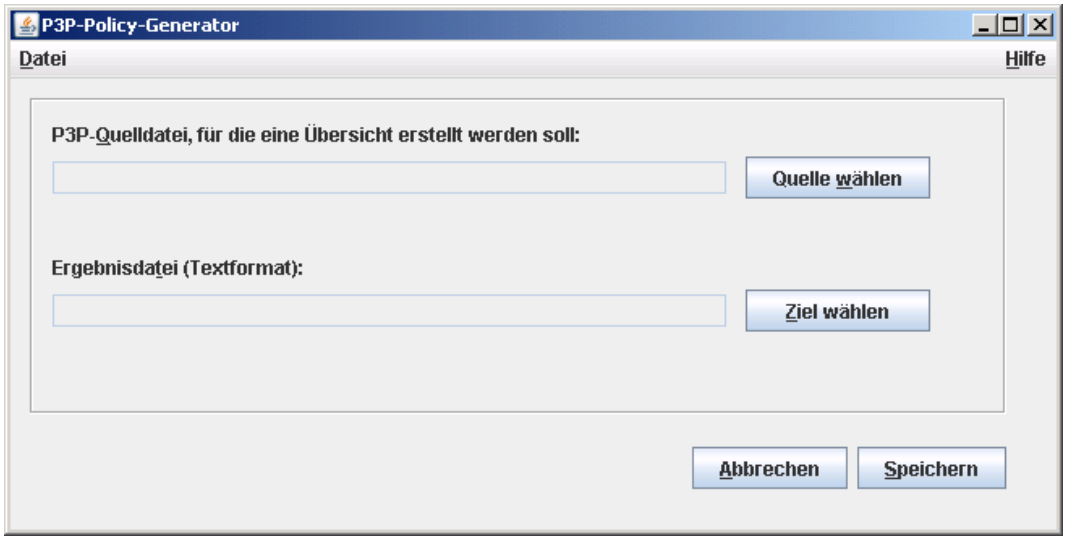

Abbildung 6.20: Fenster zur Erstellung einer Übersichtsdatei

Mit dem Generator kann eine textuelle Übersicht über die in den einzelnen P3P-Elementen enthaltenen Werte erstellt werden. Zu diesem Zweck wählt man im „Datei“-Menü den Punkt „Tabellarische Übersicht erstellen“. Daraufhin erscheint das in Abb. 6.20 gezeigte Bearbeitungsfenster. Im oberen Teil des Fensters wird die Quelldatei ausgewählt, der Button „Quelle wählen“ öffnet einen Dateiauswahl-Dialog.

Auf die gleiche Art wird darunter der Speicherort für die Übersichtsdatei im Textformat definiert. Um die Datei zu erstellen, muss der „Speichern“-Button betätigt werden.

6.6 Erstellen einer Policy-Referenzdatei

Bei der Erstellung von Policies mit oder ohne Template erzeugt der Generator eine P3P-Datei, die sowohl Policies als auch Policy-Referenzen enthält. Die beiden Bestandteile können auch in verschiedenen Dateien abgelegt werden. In diesem Fall enthält die Policy-Referenzdatei Verweise auf die einzelnen Policies in Form einer URL.

Die Möglichkeit, eine Policy-Referenzdatei erstellen zu können, ist wünschenswert, wenn die Policies bereits definiert sind und in einer separaten Datei vorliegen.

Die Erstellung einer Policy-Referenzdatei wird über den Menüpunkt „Referenzdatei erstellen“ des „Datei“-Menüs gestartet. Daraufhin erscheint das in Abb. 6.21 gezeigte Bearbeitungsfenster, das analog zum Bearbeitungsfenster für Policies in Abschnitt 6.2.1 gestaltet ist. Dieses Fenster dient zur Definition von Policy-Referenzen. Mit dem Button „Hinzufügen“ kann die Tabelle mit Referenzen gefüllt werden. Über „Bearbeiten“ gelangt man zum nächsten Bearbeitungsfenster, in dem einer Referenz ein Geltungsbereich zugewiesen werden kann.

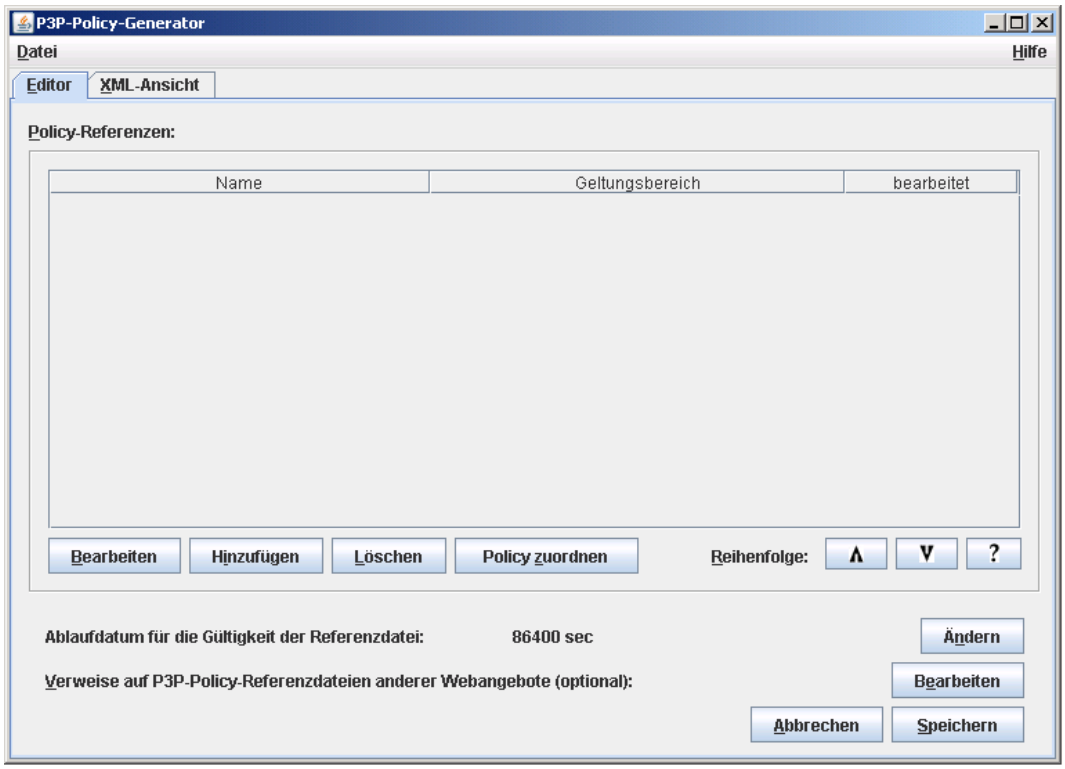

Abbildung 6.21: Fenster zur Erstellung einer Referenzdatei

Der Button „Policy zuordnen“ kann verwendet werden, um eine separate Policies-Datei zu laden und die eindeutigen Namen der darin enthaltenen Policies zu extrahieren. Diese Namen können den Referenzen in der Tabelle zugeordnet werden. Diese Angabe wird ausgewertet, wenn der Generator den Verweis auf die externe Policy erzeugt.

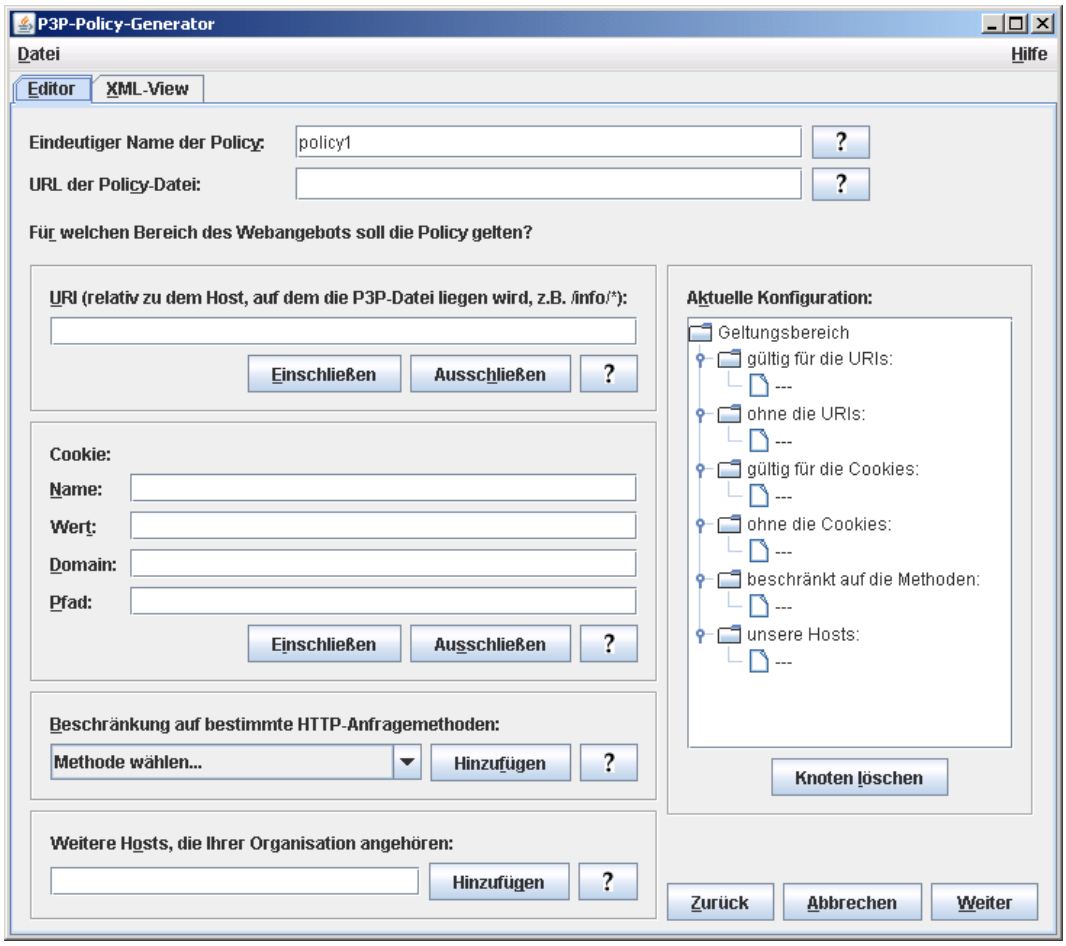

Abbildung 6.22: Fenster zur Definition eines Geltungsbereichs

Zur Bearbeitung der einzelnen Referenzen dient das in Abb. 6.22 gezeigte Fenster. Hier können Geltungsbereiche definiert werden. Dieses Fenster wird genauso bedient wie das in Abschnitt 6.2.2 beschriebene Fenster zur Definition von Geltungsbereichen. Ein Unterschied besteht lediglich darin, dass zur Definition des Verweises auf eine Policy in einer separaten Datei eine URL benötigt wird. Diese URL wird im oberen Teil des Fensters abgefragt. Nachdem der Geltungsbereich definiert wurde, kann mit dem „Weiter“-Button wieder in das Übersichtsfenster gewechselt werden (vgl. Abb. 6.21). Sind alle Referenzen bearbeitet, können sie in einer XML-Datei abgespeichert werden.

Möchte man eine P3P-Datei, die Policies und Policy-Referenzen enthält, auf zwei Dateien aufteilen, so muss zu diesem Zweck das POLICIES-Element mit seinem Inhalt in einer separaten Datei untergebracht werden. Die folgenden Listings geben einen Überblick über die Struktur der verschiedenen Dateien.

Eine P3P-Datei, die sowohl Policies als auch Policy-Referenzen enthält, hat die in Listing 6.1 skizzierte Struktur:

Listing 6.1: Struktur der P3P-Datei mit Policies und Policy-Referenzen

```
<?xml version="1.0" encoding="UTF-8"?>
<META xmlns="http://www.w3.org/2002/01/P3Pv1"
      xmlns:p3p11="http://www.w3.org/2006/01/P3Pv11"
      xmlns:p3p11bds="http://www.w3.org/2006/01/
          P3Pv11BDS">
<POLICY-REFERENCES>
   ...
</POLICY-REFERENCES>
<POLICIES>
   ...
</POLICIES>
</META>
```

Eine Policy-Referenzdatei enthält lediglich das Element POLICY-REFERENCES, nicht aber POLICIES (vgl. Listing 6.2):

Listing 6.2: Struktur der Policy-Referenzdatei

```
<?xml version="1.0" encoding="UTF-8"?>
<META xmlns="http://www.w3.org/2002/01/P3Pv1"
      xmlns:p3p11="http://www.w3.org/2006/01/P3Pv11"
      xmlns:p3p11bds="http://www.w3.org/2006/01/
          P3Pv11BDS">
<POLICY-REFERENCES>
   ...
</POLICY-REFERENCES>
</META>
```

Listing 6.3 zeigt die dazugehörige Policy-Datei, diese hat POLICIES als Wurzelelement:

Listing 6.3: Struktur der Policydatei

```
<POLICIES>
   ...
</POLICIES>
```

Zur Veröffentlichung der Policy-Referenzdatei ist die „well-known location“ zu empfehlen, d.h. die Datei sollte unter dem Pfad */w3c/p3p.xml* abgelegt werden (siehe Abschnitt 6.2.11).

7 Clientseitige Unterstützung von P3P

In den folgenden Abschnitten werden verschiedene Möglichkeiten des Einsatzes von P3P aus Nutzersicht vorgestellt.

7.1 Internet Explorer

Der Internet Explorer von Microsoft unterstützt ab Version 6 den P3P-Standard in der Version 1.0. Die folgende Darstellung dieser Unterstützung basiert auf [Goldfeder01]. Im Internet Explorer enthalten ist die automatisierte Behandlung von Cookies ausschließlich auf Basis der P3P-Compact Policies (siehe 2.5) sowie eine Anzeigefunktion, die die Elemente einer ausführlichen P3P-Policy übersetzt und anzeigt. Nicht integriert ist dagegen die Auswertung einer ausführlichen P3P-Policy.

Das Filtern von Cookies kann durch die Wahl einer von sechs Filterungsebenen erfolgen. Diese Einstellungen sind unter „Extras“ im Dialog „Internetoptionen“ verfügbar, dort kann im Reiter „Datenschutz“ durch einen Schieberegler eine Filterungsebene eingestellt werden. Standardmäßig ist hier die Ebene „Mittel“ eingestellt (siehe Abb. 7.1).

Die unterste Ebene „Alle Cookies annehmen“ bewirkt, dass grundsätzlich alle Cookies angenommen werden und alle bereits auf dem Computer vorhandenen Cookies von den Webangeboten gelesen werden können, von denen sie gesetzt wurden. Die oberste Stufe „Alle Cookies blocken“ lehnt alle Cookies ab und verweigert den Zugriff auf bereits vorhandene Cookies.

Die vier dazwischen liegenden Ebenen „niedrig“, „mittel“, „mittelhoch“ und „hoch“ sind von folgenden Fragestellungen bestimmt:

- Ist das Cookie ein Sitzungscookie oder ein permanentes Cookie?
  Sitzungscookies werden für die Dauer einer Sitzung verwendet und nach dieser Sitzung gelöscht, wohingegen permanente Cookies eine definierte Lebensdauer haben.

- Wird das Cookie vom besuchten Webangebot selbst oder von einem Drittanbieter gesetzt?
  Ein Cookie, das von einer anderen Domain stammt als von der des besuchten Webangebots, wird als Cookie eines Drittanbieters betrachtet. Unter diese Kategorie fallen z.B. Cookies, die von Werbebannern einer Fremddomain gesetzt werden sollen.

- Hat es eine Compact Policy und falls ja, ist diese aus Sicht des Internet Explorers zufriedenstellend?
  Eine Compact Policy ist für den Internet Explorer dann nicht zufriedenstellend, wenn durch das entsprechende Cookie personenbezogene Daten

für bestimmte Zwecke oder bestimmte Empfänger erhoben werden, ohne dass der Nutzer dem zustimmt. Dies ist dann der Fall, wenn die Compact Policy eine Kombination aus Elementen beider Spalten in Tabelle 7.1 enthält. Die Zweck- oder Empfängerelemente müssen dabei als zwingend erforderlich deklariert sein. Ist das *required*-Attribut eines solchen Elements mit „opt-in“- oder „opt-out“ besetzt ist, wird es nicht als inakzeptabel bewertet.

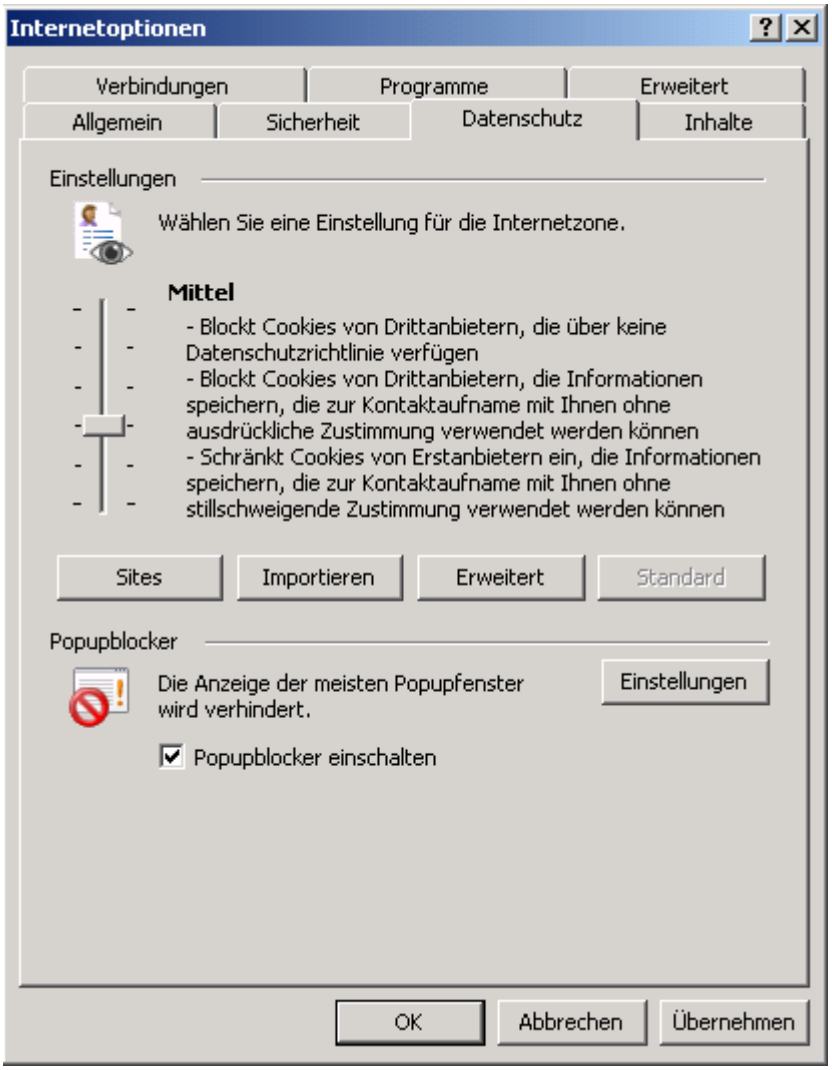

Abbildung 7.1: Einstellungen zur Filterung von Cookies

Durch diese Fragestellungen werden die Eigenschaften eines Cookies herausgearbeitet, die zur Definition der verschiedenen Filterungsebenen dienen. Ein permanentes Cookie, das vom Anbieter selbst stammt und eine nicht zufriedenstellende Compact Policy hat, wird auf der Ebene „niedrig“ angenommen, während es auf der Ebene „hoch“ geblockt wird. Permanente Cookies des Erstanbieters, die keine P3P-Compact-Policy mitbringen, werden nur auf den Ebenen „Alle Cookies blocken“ und „Hoch“ abgelehnt. Eine detaillierte Darstellung der einzelnen Filterungsebenen findet sich in [Goldfeder01].

Kategorien		Zweck/Empfänger	
physical	physische Kontaktinformationen	same	Dritte mit gleichen Datenschutzpraktiken
online	Online-Kontaktinformationen	other-recipient	Dritte mit anderen Datenschutzpraktiken
government	Staatliche Identifikatoren	unrelated	Dritte mit unbekannten Datenschutzpraktiken
financial	Informationen über die Finanzen eines Nutzers	public	Öffentliche Foren
		individual-analysis	Analysen, die auf den einzelnen Nutzer bezogen sein können
		individual-decision	Reaktionen auf das bisherige Verhalten eines Nutzers
		contact	Kontaktieren des Nutzers nicht per Telefon, sondern auf anderem Wege
		telemarketing	Kontaktieren des Nutzers per Telefon
		other-purposes	andere Zwecke

Tabelle 7.1: Inakzeptable Elementkombinationen (vgl. [Goldfeder01])

An den Filterungsebenen des Internet Explorers bemängeln Kritiker beispielsweise, dass Cookies von Werberingen, die das Nutzerverhalten auf der Basis von Pseudonymen analysieren, selbst auf der Filterungsebene „hoch“ klaglos akzeptiert werden (vgl. [JAP06]).

Zur individuelleren Konfiguration der Cookiebehandlung kann der Nutzer seine Präferenzen in einem Microsoft-eigenen Format festlegen und diese in den Browser importieren. Sichere Voreinstellungen in diesem Format, das auf XML basiert, werden von verschiedenen Organisationen wie z.B. vom Unabhängigen Landeszentrum für Datenschutz Schleswig-Holstein (ULD) und von der TU Dresden im Rahmen des AN.ON-Projekts empfohlen und zum Download angeboten (siehe [JAP06]).

Die vollständige P3P-Policy eines Webangebots kann angezeigt werden über das Menü „Ansicht“ und den Menüpunkt „Datenschutzrichtlinie der Webseite...“. Nach Auswahl dieses Menüpunktes erscheint der in Abb. 7.2 gezeigte Dialog.

Hier werden alle Inhalte aufgelistet, die in der Webseite enthalten sind, darunter auch Grafiken, Skripte etc. Wählt man einen der Inhalte aus, kann man über den Button „Zusammenfassung“ die Übersetzung des Internet Explorers für die vollständige P3P-Policy anzeigen lassen, sofern dem Inhalt eine Policy zugeordnet ist (siehe Abb. 7.3).

Im unteren Teil des Fensters kann für den Inhalt, der durch die oben angezeigte Policy beschrieben wird, eine Verfahrensweise definiert werden, Cookies können entweder immer zugelassen oder immer abgeleht werden, oder aber basierend auf den eigenen Einstellungen verarbeitet werden.

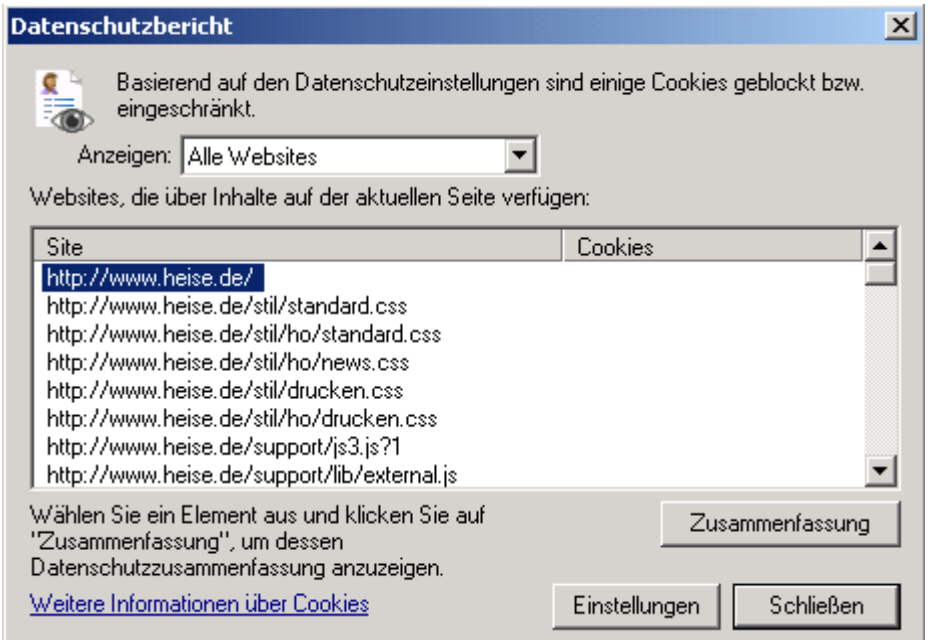

Abbildung 7.2: Dialog Datenschutzbericht

7.2 Privacy Bird

Der Privacy Bird ist ein Plugin für den Internet Explorer ab Version 5.01. Es wurde entwickelt von AT&T, seit Version 1.3 wird es vom CMU Usable Privacy and Security Laboratory der Carnegie Mellon Universität weiterentwickelt (vgl. [Pri08]). Das Plugin ist über der Titelleiste des Browsers in Form eines Vogels sichtbar. Der Privacy Bird überprüft die P3P-Policy jeder angefragten Webseite, vergleicht sie mit den im Plugin abgelegten Nutzerpräferenzen und zeigt je nach Ergebnis der Überprüfung ein entsprechendes Vogel-Symbol an. Dabei verfärbt sich der Vogel analog zu einer Ampel: Bei Übereinstimmung mit den Nutzerpräferenzen ist der Vogel grün, bei fehlender Policy ist der ratlose Vogel gelb, bei Nichtübereinstimmung läuft er rot an. Wenn das Plugin deaktiviert ist, wird ein grauer schlafender Vogel angezeigt (siehe Abb. 7.4).

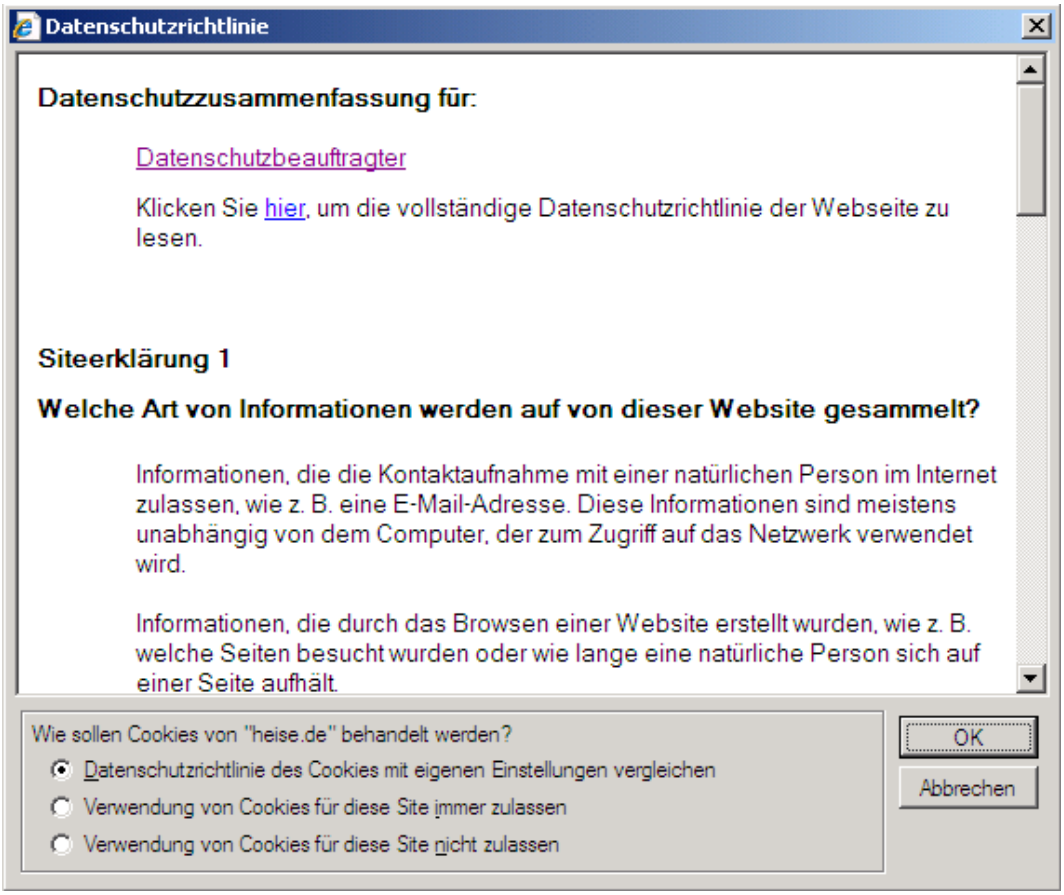

Abbildung 7.3: Ansicht einer Policy

Abbildung 7.4: Privacy Bird Symbole

Neben der Farbsymbolik informiert das Plugin den Nutzer auf Wunsch auch auf akustischem Wege: Bei Übereinstimmung schweigt der Vogel, bei fehlender Policy erklingt ein Vogelzwitschern, bei Nichtübereinstimmung ein wütendes Krähen.

Gewarnt wird der Nutzer zudem, wenn er ein Formular eines Webangebots ausgefüllt hat, das nicht über eine P3P-Policy verfügt oder dessen Policy nicht übereinstimmt mit den Nutzerpräferenzen. Die Warnung erscheint in Form eines Pop-up-Fensters, bevor die in das Formular eingegebenen Daten abgeschickt werden. Die Warnung enthält eine Zusammenfassung der Anbieter-Policy, sofern diese vorhanden ist. Andernfalls erscheint die Frage, ob der Nutzer bei diesem Webangebot in Zukunft weiterhin Warnungen erhalten oder diese lieber abschalten möchte (vgl. [Cranor02a]).

Das Hauptmenü des Privacy Bird erreicht man durch Anklicken des Vogel-Symbols. Das Plugin ist bisher nur in englischer Sprache verfügbar. Um mehr über die Datenschutzpraktiken des besuchten Webangebots zu erfahren, bietet der Menüpunkt „About this Site“ verschiedene Optionen (siehe Abb. 7.5).

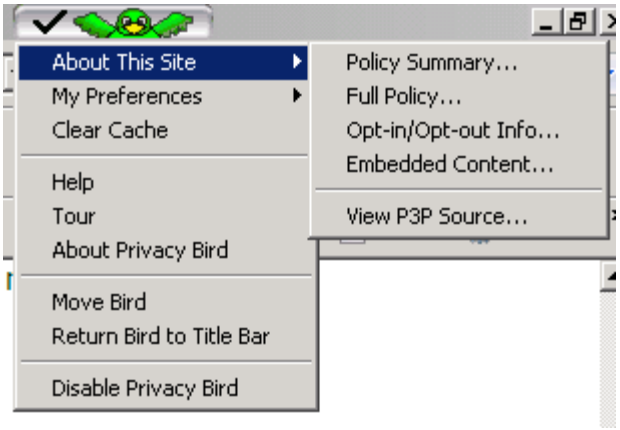

Abbildung 7.5: Hauptmenü und Untermenü „About this Site“

Der Menüpunkt „Policy Summary...“ gibt zum einen eine Übersicht über wichtige Punkte einer Policy, aufgeführt werden hier Kontaktinformationen zum Anbieter (ENTITY-Element), Anlaufstellen bei Streitigkeiten (DISPUTES-Element), Möglichkeiten des Zugriffs auf Daten, die über den Nutzer gespeichert sind (ACCESS-Element) sowie natürlichsprachliche Zusammenfassungen der enthaltenen Statements. Zum anderen findet man hier das Ergebnis des „Privacy Checks“, das darüber informiert, ob die Policy des Webangebots den Nutzerpräferenzen entspricht und falls nein, warum nicht.

Mit Aufruf des Menüpunktes „Full Policy...“ öffnet sich ein separates Browserfenster, das die natürlichsprachliche Datenschutzerklärung des Webangebots anzeigt. Deren URI ist angegeben im Attribut *discuri* der aktuell geltenden P3P-Policy, genauer: im entsprechenden POLICY-Element. Durch den Menüpunkt „Opt-in/Opt-out Info...“ öffnet sich eine URI mit Informationen zum expliziten Widerspruch („opt-out“) bzw. zur expliziten Einwilligung („opt-in“) in bestimmte Praktiken des Anbieters. Diese URI findet sich im *opturi*-Attribut der aktuell geltenden P3P-Policy.

Der Aufruf von „Embedded Content...“ bietet eine Übersicht über alle Inhalte, die in die aufgerufene Webseite eingebettet sind. Zu jedem Inhalt wird das Ergebnis des entsprechenden „Privacy Checks“ angezeigt, zudem kann eine Zusammenfassung der zugehörigen Policy und die P3P-Quell-Datei angefordert werden, sofern dem Inhalt eine P3P-Policy zugeordnet ist (siehe Abb. 7.6).

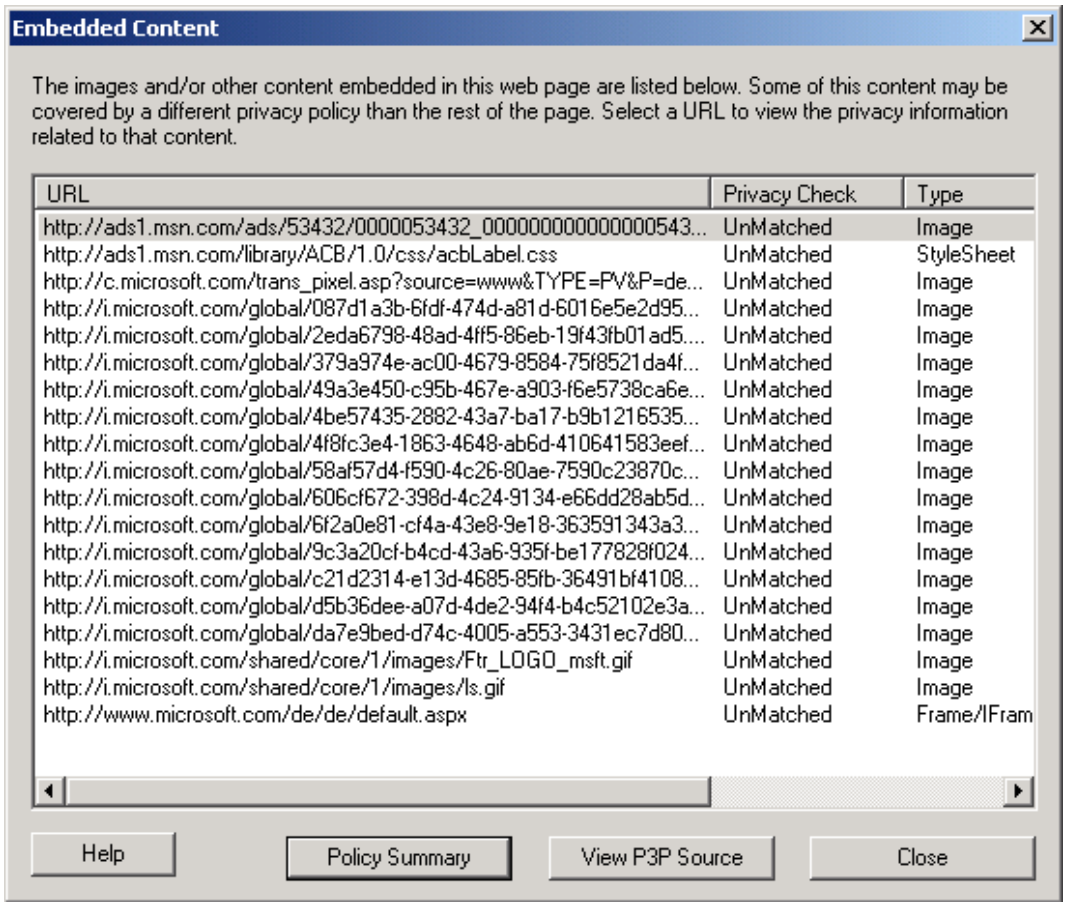

Abbildung 7.6: Übersicht „Embedded Content“

Der Nutzer kann dem Plugin seine Präferenzen mitteilen, indem er im Hauptmenü „My Preferences“ und im zugehörigen Untermenü den Punkt „Privacy. . .“ auswählt. Daraufhin erschient das in Abb. 7.7 gezeigte Dialogfenster.

Die Nutzerpräferenzen können konfiguriert werden, indem eine der drei vorkonfigurierten Datenschutz-Stufen ausgewählt wird: „niedrig“, „mittel“ oder „hoch“. Entsprechend der gewählten Stufe wird nun angezeigt, welche Verfahrensweisen eines Anbieters Warnungen auslösen werden. Wird an den vorkonfigurierten Einstellungen etwas geändert, wird als Stufe „custom“ eingestellt. Neben den vorgegebenen Stufen gibt es die Möglichkeit, Dateien zu importieren, die Präferenzeinstellungen im APPEL-Format enthalten. APPEL ist eine vom W3C entwickelte Sprache zum Austausch von P3P-Präferenzen (vgl. [Cranor02b]). Der Import kann durch den Button „Import Settings“ initiiert werden.

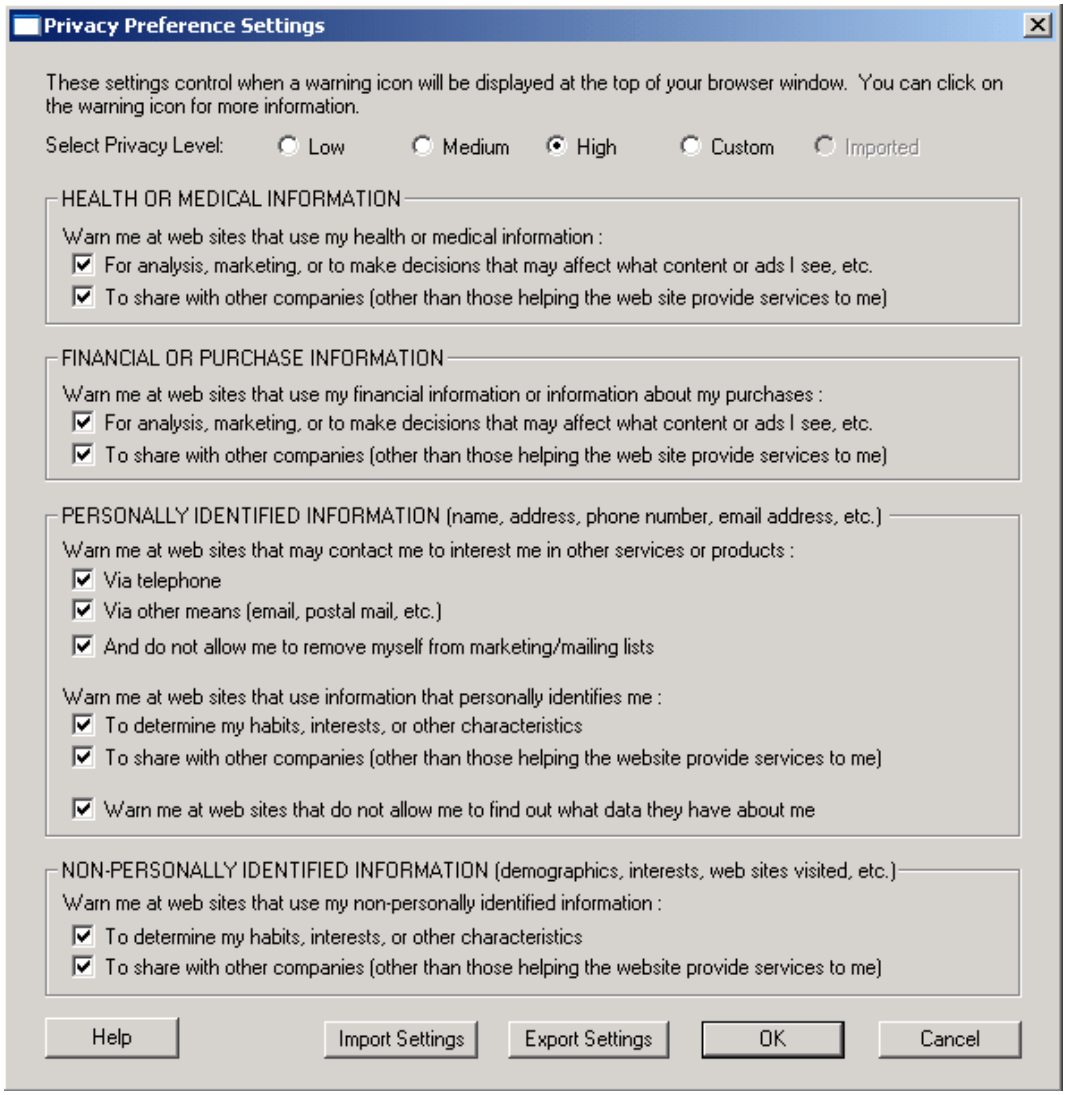

Abbildung 7.7: Dialog zur Konfiguration der Nutzerpräferenzen

7.3 Mozilla

In der derzeit aktuellen Version 1.1.8 von Mozilla SeaMonkey, in der unter anderem ein Web-Browser enthalten ist, wird P3P insofern unterstützt, als Cookies auf Basis ihrer Compact Policies verarbeitet werden können und Zusammenfassungen der vollständigen P3P-Policies angezeigt werden können. Die aktuelle Version 2.0.0.12 des Firefox-Browsers enthält diese Unterstützung nicht. Ob die P3P-Unterstützung auch zukünftig von Seamonkey beibehalten werden wird, ist fraglich, weil der entsprechende Code aus den Mozilla-Repositories entfernt wurde (siehe `https://bugzilla.mozilla.org/show_bug.cgi?id=366611`).

Die Konfiguration der Cookieverwaltung durch P3P ist im Seamonkey-Browser erreichbar über „Bearbeiten“ – „Einstellungen...“, im entsprechenden Dialog ist unter „Datenschutz & Sicherheit“ der Unterpunkt „Cookies“ zu wählen (siehe Abb. 7.8).

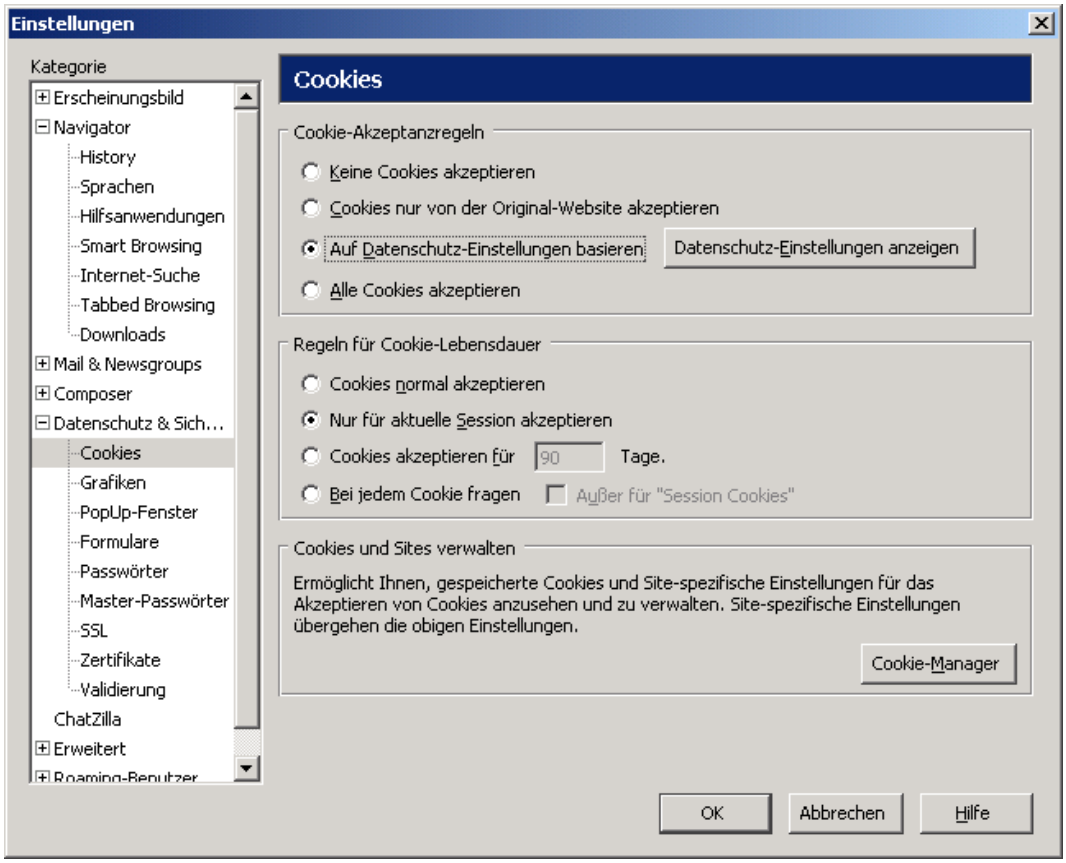

Abbildung 7.8: Seamonkey: Cookie-Einstellungen

Unter „Cookie-Akzeptanzregeln“ kann durch Wahl der Option „Auf Datenschutz-Einstellungen basierend“ individuell definiert werden, welche Art von Cookie wie behandelt werden soll. Mit dem zugehörigen Button „Datenschutz-Einstellungen anzeigen“ öffnet sich der Dialog zur Konfiguration der Sicherheitsstufe. Hier stehen drei vordefinierte Sicherheitsstufen sowie die Stufe „be-

nutzerdefiniert“ zur Verfügung (siehe Abb. 7.9). Ist letztere ausgewählt, kann im unteren Teil des Dialogs das Verhalten des Browsers definiert werden:

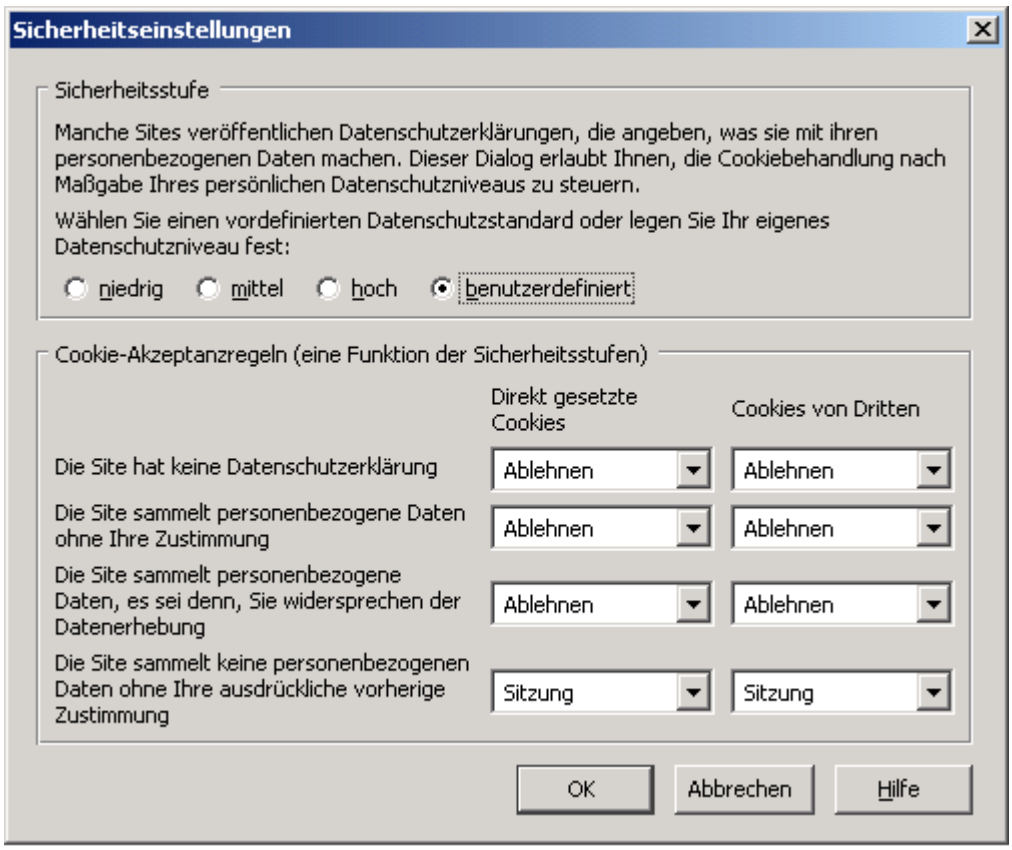

Abbildung 7.9: Seamonkey: Sicherheitsstufen und Cookie-Akzeptanzregeln

Die in Abb. 7.9 angegebenen Empfehlungen sind orientiert an [ULD08b]. In Abhängigkeit von den Aussagen der P3P-Compact Policy (linke Spalte) kann ein Cookie einer bestimmten Kategorie akzeptiert, nur für eine Sitzung akzeptiert, abgelehnt oder markiert werden. Wenn ein Cookie markiert wird, wird es in der Registerkarte „Gespeicherte Cookies“ des „Cookie-Managers“ als markiert angezeigt. Letzterer ist erreichbar über den Dialog aus Abb. 7.8. Zudem wird für markierte Cookies das Cookie-Benachrichtigungssymbol angezeigt: Dieses Symbol ist im rechten unteren Teil des Browserfensters sichtbar und kann angeklickt werden, um weitere Informationen über das Cookie zu erhalten (siehe Abb. 7.10).

Abbildung 7.10: Seamonkey: Cookie-Benachrichtigungs-Symbol

Nähere Informationen zur vollständigen P3P-Policy können über „Ansicht“ – „Seiteninformationen“ im Reiter „Datenschutz“ abgerufen werden (siehe Abb. 7.11). Hier sind alle Inhalte aufgeführt, die in der besuchten Webseite enthalten sind. Einzelne Inhalte oder auch die gesamte Webseite können hier ausgewählt werden. Über die Buttons im unteren Teil des Fensters können Informationen für den selektierten Inhalt angezeigt werden: Über „Richtlinien“ wird die natürlichsprachliche Datenschutzerklärung des entsprechenden Anbieters angezeigt, über „Optionen“ wird eine Webseite aufgerufen, die Informationen zum expliziten Widerspruch bzw. zur expliziten Einwilligung in bestimmte Datenschutzpraktiken des Anbieters enthält. Der Button „Zusammenfassung“ öffnet ein Fenster, das eine natürlichsprachliche Zusammenfassung der vollständigen P3P-Policy anzeigt. Diese ist – wie beim Internet Explorer – in der Sprache des Browsers verfasst (vgl. [Sea06]).

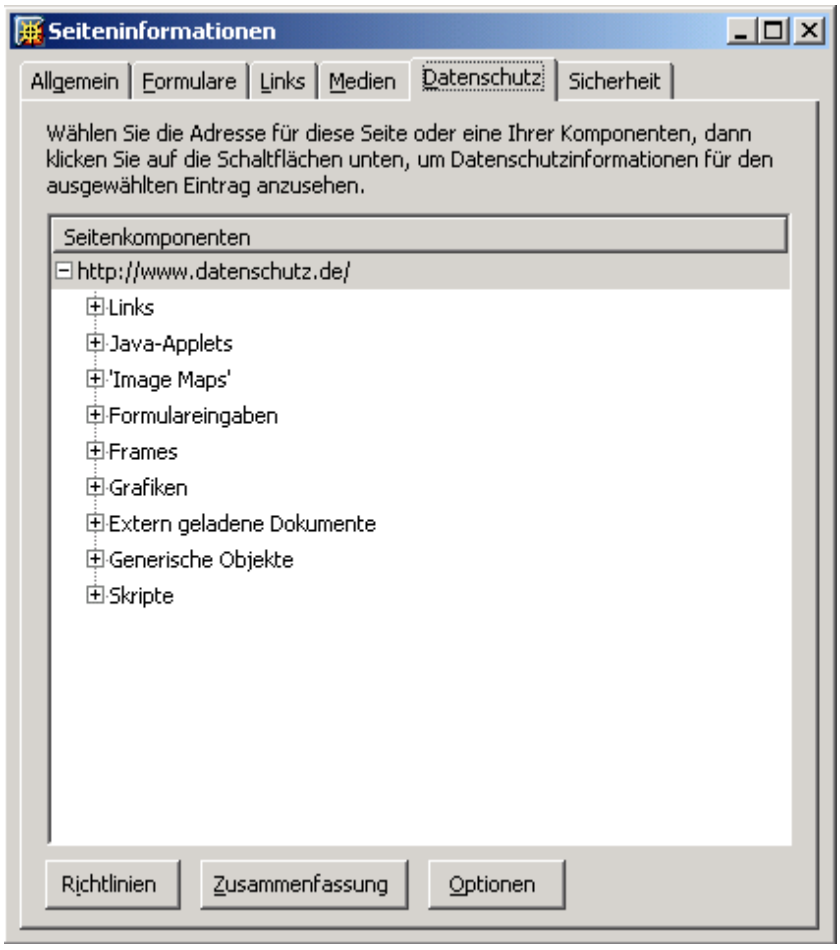

Abbildung 7.11: Seamonkey: Seiteninformationen

7.4 JRC Proxy

Der JRC P3P Proxy Service ist ein Prototyp für einen P3P-Nutzeragenten, der im Gegensatz zu den bisher vorgestellten Ansätzen ein externes Programm startet, das als lokaler Proxy agiert. Dieser verarbeitet HTTP-Anfragen, indem er prüft, ob die angefragte Webseite eine für den Nutzer akzeptable Policy hat. Nur falls diese Prüfung positiv ausfällt, wird die Anfrage an den Zielserver weitergegeben (vgl. [ULD08a]). Der JRC Proxy ist ein Open-Source-Programm. Er wurde vom „Joint Research Centre“ (JRC) in Ispra, Italien entwickelt und implementiert P3P in der Version 1.0. Zur Ausführung des Proxy wird eine Java-Umgebung benötigt.

Der JRC Proxy kann von der Homepage des „JRC P3P Resource Centre“ heruntergeladen werden, unter der Rubrik „Downloads“ findet man das Paket „Personal proxy + Docs + Source Code + Toolkit“ (siehe `http://p3p.jrc.it/downloadP3P.php`). Die heruntergeladene zip-Datei ist in ein beliebiges Verzeichnis zu entpacken.

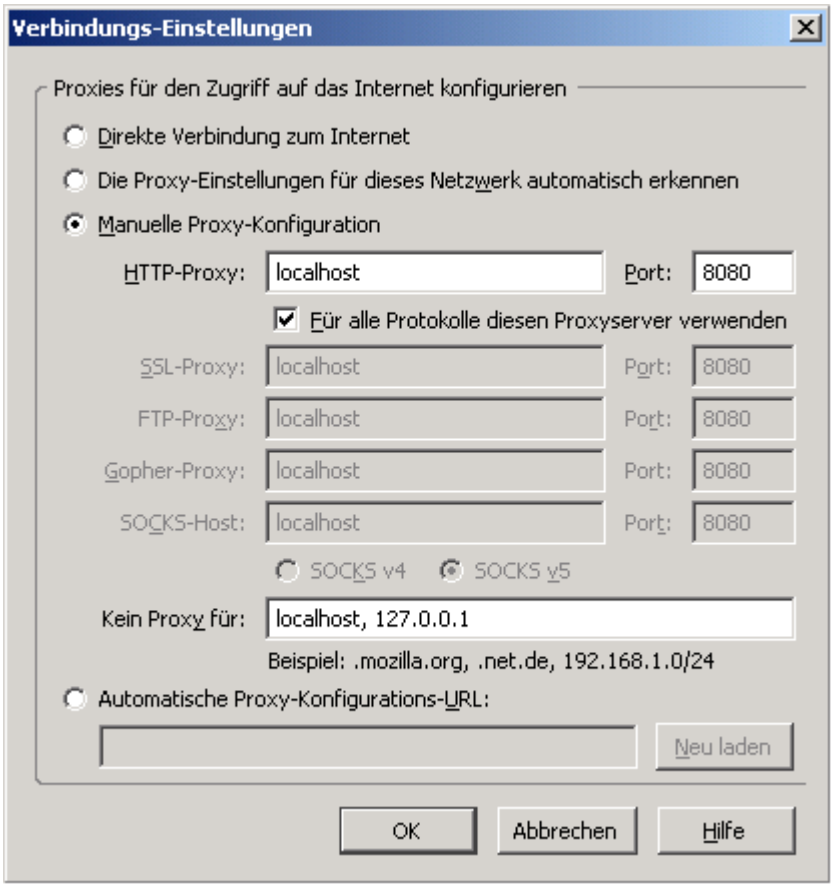

Abbildung 7.12: Einstellungen im Firefox zur Einrichtung eines lokalen Proxy

Um den JRC Proxy verwenden zu können, muss der Benutzer zunächst seinen Browser so konfigurieren, dass dieser sich nicht direkt mit dem Internet verbindet, sondern alle Anfragen zuerst an den Proxy übergibt. Diese Einstellungen können in allen gängigen Browsern auf ähnliche Weise vorgenommen

werden, bei Mozilla Firefox z.B. über „Extras“ – „Einstellungen...“. Dort findet sich unter dem Menüpunkt „Erweitert“ im Reiter Netzwerk ein Feld „Verbindung“, in dem man festlegen kann, wie sich der Browser mit dem Internet verbindet. Der Button „Einstellungen...“, der in diesem Feld enthalten ist, öffnet den in Abb. 7.12 gezeigten Dialog. Hier kann unter „Manuelle Proxy-Konfiguration“ der lokale Proxy wie in Abb. 7.12 eingerichtet werden. Im Internet Explorer kann die entsprechende Konfiguration über „Extras“ – „Internetoptionen“ im Reiter „Verbindungen“ vorgenommen werden (vgl. [ULD08a]).

Zum Starten des Proxy unter Windows ist die Batch-Datei *P3PProxySetup.bat* auszuführen, die im Unterverzeichnis `toolkitplusproxy/proxy` zu finden ist. Durch Doppelklick auf diese Datei öffnet sich ein Kommandozeilenfenster, hier werden Meldungen des Proxy ausgegeben (siehe Abb. 7.13).

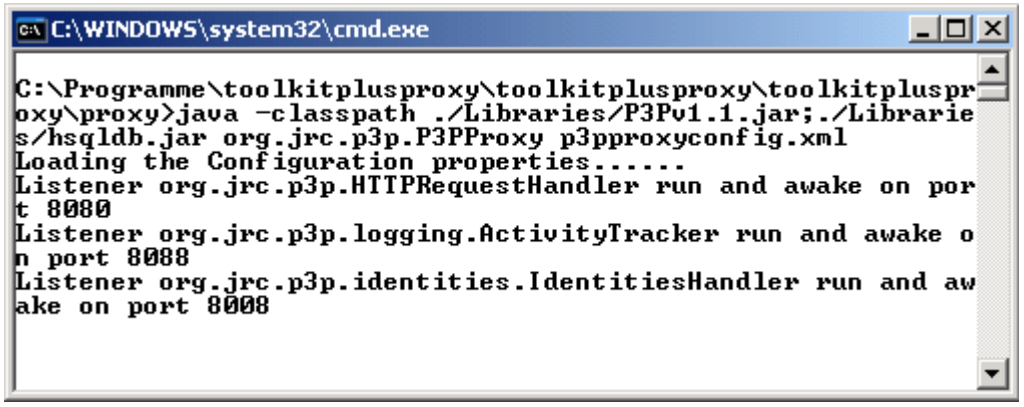

Abbildung 7.13: Start des JRC Proxy

Nun kann im Browser eine beliebige Webseite angefragt werden. Daraufhin erscheint im Browser eine Auswahlseite, die den Nutzer auffordert, ein Profil zu wählen. Ein solches Profil dient dazu, die Datenschutzpräferenzen des Nutzers zu definieren. Ein Nutzer kann auch mehrere Profile anlegen, um situationsabhängig unterschiedliche Präferenzen für den Proxy zu verwenden (vgl. [JRC]).

Die Profil-Auswahlseite bot im Test nicht die Möglichkeit, ein neues Profil zu definieren, dies war erst nach Auswahl eines der vordefinierten Profile möglich. Nach dieser Auswahl wird die angefragte Webseite überprüft, die Policy ausgewertet und das Ergebnis angezeigt. Zu der angeforderten Ressource wird ein Javascript-Menü angezeigt (siehe Abb. 7.14). Über dieses Menü kann ein neues Profil angelegt werden, dazu dient der Menüpunkt „New Profile“. Bei Auswahl dieses Punktes erscheint eine Seite, die den Nutzer nach einer Kennung und einem Passwort für das neue Profil fragt und sechs vordefinierte Präferenzmodelle anbietet. Alternativ kann der Nutzer auch eine eigene Datei importieren. Die Nutzerpräferenzen werden im APPEL-Format abgelegt (siehe [Cranor02b]) und vom Proxy als sogenannte Identitäten verwaltet.

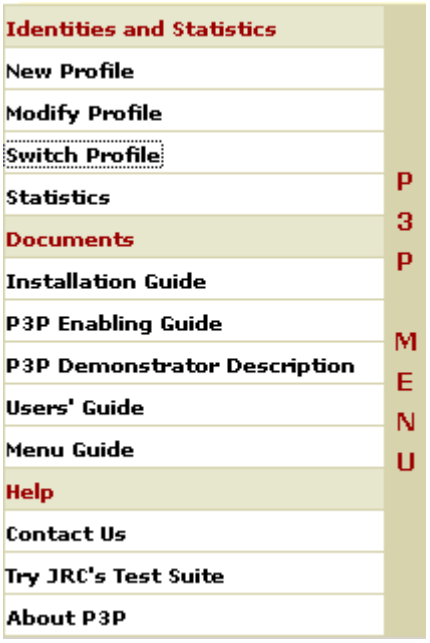

Abbildung 7.14: JRC Proxy: Hauptmenü

Die vordefinierten Präferenzmodelle reichen von „Very Low“ bis „Very High“, auch ein „ComplyWithEUDirective“-Modell wird angeboten. Webangebote ohne P3P-Policy können allerdings selbst mit dem Präferenzmodell „Very Low“ nicht aufgerufen werden. Da P3P bisher noch nicht weit verbreitet ist, ist der Einsatz des JRC Proxy bislang noch nicht praktikabel (vgl. [ULD08a]).

8 Zusammenfassung und Ausblick

Der im Rahmen dieser Diplomarbeit entwickelte P3P-Policy-Generator setzt alle an seine Funktionalität gestellten Anforderungen um. Das Programm ermöglicht die einfache Erstellung von P3P-Dateien für Nutzer ohne Vorwissen auf dem Gebiet des P3P-Standards. Mit Hilfe von Templates können dem Benutzer Belegungen der einzelnen Elemente vorgegeben werden, die konform zu deutschem Datenschutzrecht sind. Neben P3P-Dateien können auch Compact Policies und Übersichts-Dateien erzeugt werden.

An die Gestaltung der Benutzerschnittstelle wurden Anforderungen auf dem Gebiet der Softwareergonomie und der Barrierefreiheit gestellt. Im Bereich der Softwareergonomie erfüllt der Generator viele Anforderungen, Erweiterungen könnten hauptsächlich im Bezug auf Individualisierbarkeit und Lernförderlichkeit vorgenommen werden. Auf dem Gebiet der Barrierefreiheit wurden grundlegende Richtlinien befolgt, der Generator könnte jedoch um einige weitere Maßnahmen erweitert werden.

Um die Umsetzung der Benutzerschnittstelle auf Nutzerfreundlichkeit zu testen und zu überprüfen, ob die Hilfestellungen des Generators und die Anleitung eine erfolgreiche Bedienung ermöglichen, könnte ein Einführungsprojekt durchgeführt werden.

Eine weitere sinnvolle Erweiterung des Generators wäre eine Funktion zum automatischen Speichern der erzeugten Datei auf dem Webspace des Anbieters. Zu diesem Zweck müsste der Nutzer berechtigt sein, auf die „well-known location“ des Webangebots zuzugreifen. Dies könnte durch den Aufbau einer FTP-Verbindung geschehen, zu der der Nutzer die Zugangsdaten kennen müsste.

Auch die Entwicklung von Nutzerpräferenzen, die das Pendant zu den Templates dieser Arbeit bilden, wäre wünschenswert. Diese Präferenzen sollten am deutschen Datenschutzrecht ausgerichtet sein, so dass der Nutzer auf Policies, die diese Vorgaben nicht erfüllen, aufmerksam gemacht werden kann. Die Präferenzen könnten im Hinblick auf die Tätigkeiten entwickelt werden, die der Nutzer ausüben möchte (z.B. Informationsangebote nutzen, Einkaufen, etc.). Zur Formulierung der Präferenzen könnte APPEL verwendet werden, eine vom W3C entwickelte Sprache zur Definition von Präferenzen auf Nutzerseite, dieses Format kann beispielsweise vom Nutzeragenten „Privacy Bird“ importiert werden.

Neben der Entwicklung des Generators wurde im Rahmen dieser Arbeit der P3P-Standard auf seine Vereinbarkeit mit dem deutschen Datenschutzrecht untersucht. Dabei wurde deutlich, dass nicht alle rechtlichen Forderungen eine direkte Entsprechung im P3P-Vokabular haben. Diese Punkte können jedoch

in Elementen, die natürlichsprachliche Beschreibungen enthalten, oder in der Datenschutzerklärung zum Ausdruck gebracht werden. Eine nichtabstreitbare Vereinbarung zwischen Anbieter und Nutzer mit Hilfe von Signaturen ist im P3P-Standard noch nicht integriert, aber für zukünftige Versionen vorgesehen.

Der Einsatz von P3P kann – nicht nur für deutsche Behörden – empfohlen werden, um Transparenz hinsichtlich des Datenschutzverhaltens eines Anbieters zu schaffen und so das Vertrauen der Nutzer in die Seriosität des Angebots zu fördern. Nutzer profitieren vom automatischen Abgleich einer Policy mit ihren Präferenzen. Dadurch wird es ihnen erleichtert, die Kontrolle über ihre Daten zu behalten und Angebote zu erkennen, die sich nicht ihren Präferenzen entsprechend verhalten.

Die Entwicklung von datenschutzfreundlichen Policies und Präferenzen, die von unabhängigen Organisationen empfohlen werden, könnte die Akzeptanz und Verbreitung von P3P fördern und Nutzer bei der Konfiguration ihrer P3P-Tools unterstützen. Für die Anbieterseite liefern die Templates dieser Arbeit einen Ansatz.

Literaturverzeichnis

[Armstrong08] Armstrong, Eric, Santos, Tom, Wilson, Steve. *Understanding the TreeModel.* URL: `http://java.sun.com/products/jfc/tsc/articles/jtree/index.html`, [eingesehen: 4. Februar 2008], 2008.

[Batista00] Batista, Vitor. *Datenschutz im Internet: Kritische Bewertung und Implementierung von P3P gemäß TDDSG.* Diplomarbeit, Johann Wolfgang Goethe-Universität Frankfurt am Main, Juni 2000.

[BDS06] *Bundesdatenschutzgesetz (BDSG).* Ausfertigungsdatum: 20.12.1990. Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), zuletzt geändert durch Artikel 1 des Gesetzes vom 22. August 2006 (BGBl. I S. 1970). URL: `http://www.gesetze-im-internet.de/bdsg_1990`, [eingesehen: 26. Februar 2008], 22. August 2006.

[Berners-Lee05] Berners-Lee, T., Fielding, R., Masinter, L. *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax.* URL: `http://www.rfc-editor.org/rfc/rfc3986.txt`, [eingesehen: 10. Februar 2008], Januar 2005.

[Bray06a] Bray, Tim, Hollander, Dave, Layman, Andrew, Tobin, Richard. *Namespaces in XML 1.0 (Second Edition).* Status: W3C Recommendation, URL: `http://www.w3.org/TR/xml-names/`, [eingesehen: 10. Februar 2008], 16. August 2006.

[Bray06b] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C. M., Maler, Eve, Yergeau, François. *Extensible Markup Language (XML) 1.0 (Fourth Edition).* Status: W3C Recommendation, veröffentlicht am 29. September 2006, URL: `http://www.w3.org/TR/xml/`, [eingesehen: 7. Februar 2008], 16. August 2006.

[Clark99] Clark, James, DeRose, Steve. *XML Path Language (XPath) Version 1.0.* Status: W3C Recommendation, URL: `http://www.w3.org/TR/xpath`, [eingesehen: 10. Februar 2008], 16. November 1999.

[Cranor02a] Cranor, Lorrie. *Web Privacy with P3P.* O'Reilly, Sebastopol, CA, USA, September 2002.

[Cranor02b] Cranor, Lorrie, Langheinrich, Marc, Marchiori, Massimo. *A P3P Preference Exchange Language 1.0 (APPEL1.0).* Hrsg.: Marc Langheinrich, Status: W3C Working Draft, URL: `http://www.w3.org/TR/P3P-preferences/`, [eingesehen: 23. Februar 2008], 15. April 2002.

[Epi00] *Pretty Poor Privacy: An Assessment of P3P and Internet Privacy.* EPIC (Electronic Privacy Information Center) and Junkbusters, URL: `http://www.epic.org/reports/prettypoorprivacy.html`, [eingesehen: 7. Juni 2007], Juni 2000.

[Fallside04] Fallside, David C., Walmsley, Priscilla. *XML Schema Part 0: Primer Second Edition.* Status: W3C Recommendation, URL: `http://www.w3.org/TR/xmlschema-0/`, [eingesehen: 10. Februar 2008], 28. Oktober 2004.

[Geis07a] Geis, Ivo. *Datenschutzrecht.* Online-Publikation, URL: `http://www.ivo-geis.de/veroeffentlichungen/datenschutzrecht.pdf`, [eingesehen: 13. Januar 2008], Februar 2007.

[Geis07b] Geis, Thomas. *(Die neue) DIN EN ISO 9241-110 („Grundsätze der Dialoggestaltung").* In: FIT für Usability, eine Online-Initiative des Fraunhofer-Instituts FIT, URL: `http://www.fit-fuer-usability.de/1x1/knigge/110.html`, [eingesehen: 15. März 2008], 19. September 2007.

[Goldfeder01] Goldfeder, Aaron, Leibfried, Lisa. *Privacy in Internet Explorer 6.* In: *Web Privacy with P3P.* O'Reilly, Sebastopol, Oktober 2001. Abdruck des Artikels „Privacy in Internet Explorer 6" der MSDN Library, online unter `http://msdn2.microsoft.com/en-us/library/ms537343(VS.85).aspx`, [eingesehen: 18. Februar 2008].

[Grimm00] Grimm, Rüdiger, Roßnagel, Alexander. *Weltweiter Datenschutzstandard?* In: *Global@Home* (Herausgegeben von Kubicek, H., et al.), Bd. 8 von *Jahrbuch Telekommunikation und Gesellschaft.* Hüthig Verlag, Heidelberg, Januar 2000.

[Grimm03] Grimm, Rüdiger. *Datenverarbeitung im Internet.* In: *Datenschutz im Electronic Commerce* (Herausgegeben von Wegerich, Thomas, Holznagel, Bernd, Koenig, Christian, Scherer, Joachim, Tschentscher, Thomas), Bd. 18 von *Schriftenreihe Kommunikation und Recht.* Verlag Recht und Wirtschaft, Heidelberg, 1. Aufl., 2003.

[Harold03] Harold, Elliotte Rusty, Means, W. Scott. *XML In A Nutshell.* O'Reilly Verlag, Köln, 2. Aufl., 2003.

[Herczeg05] Herczeg, Michael. *Software-Ergonomie – Grundlagen der Mensch-Computer-Kommunikation.* Oldenbourg Wissenschaftsverlag, München, 2. Aufl., 2005.

[JAP06] *Datenschutzeinstellungen im Internet Explorer 6.0.* URL: `http://anon.inf.tu-dresden.de/ie6_privacy.html`, [eingesehen: 18. Februar 2008], 2006.

[JRC] *How to use the Demonstrator.* In: Bedienungsanleitung des JRC Personal Proxy mit dem Titel „P3P Demonstrator Users' Guide", enthalten im Softwarepaket „Personalproxy+Docs+SourceCode+Toolkit", Quelle: `http://p3p.jrc.it/downloadP3P.php`, [Download am 20. Februar 2008].

[Kersken04] Kersken, Sascha. *Kompendium der Informationstechnik.* URL: `http://www.galileocomputing.de/openbook/kit`, September 2004.

[Krüger07] Krüger, Guido, Stark, Thomas. *Handbuch der Java-Programmierung.* Online-Publikation, HTML-Ausgabe 5.0.1, URL: `http://www.javabuch.de/download.html`, [eingesehen: 14. März 2008], 2007.

[Langheinrich01] Langheinrich, Marc. *P3P – Ein neuer Standard für Datenschutz im Internet.* digma – Zeitschrift für Datenrecht und Informationssicherheit, Bd. 1, (2001), 32–34.

[LfD02] *Hinweise für die Behandlung von Webserverlogdateien vor der Weitergabe an Dritte zur Auswertung.* Hrsg.: Der Landesbeauftragte für den Datenschutz Rheinland-Pfalz, URL: `http://www.datenschutz.rlp.de/materialien/hinweise/info_webserverlogfiles.pdf`, [eingesehen: 3. März 2008], September 2002.

[Lindskog03] Lindskog, Helena, Lindskog, Stefan. *Web Site Privacy with P3P.* Wiley Publishing, April 2003.

[Metsker02] Metsker, Steven John. *Design patterns Java workbook.* The Software patterns series. Pearson Education, Boston, MA, USA, August 2002.

[NBA08] *The Accessibility Developer's Corner.* URL: `http://a11y.netbeans.org/index.html`, [eingesehen: 17. März 2008], 2008.

[Niedermeier06] Niedermeier, Stephan, Scholz, Michael. *Java und XML - Grundlagen, Einsatz, Referenz.* Galileo Press, Bonn, 1. Aufl., 2006.

[Patzak06] Patzak, Andrea. *Datenschutzrecht für den E-Commerce*, Bd. 28 von *Frankfurter Studien zum Datenschutz*. Nomos Verlagsgesellschaft, Baden-Baden, 2006. Dissertation an der Universität Wien, 2004.

[Pri08] Webseite, URL: `http://www.privacybird.org/`, [eingesehen: 20. Februar 2008], 2008.

[Roßnagel03] Roßnagel, Alexander. *Datenschutz in Tele- und Mediendiensten*. In: *Handbuch Datenschutzrecht* (Herausgegeben von Roßnagel, Alexander), S. 1278–1323. Verlag C.H.Beck, München, 2003.

[Schaar02] Schaar, Peter. *Datenschutz im Internet – Die Grundlagen*. Verlag C.H.Beck, München, 2002.

[Sea06] *Datenschutz-Optionen verwenden*. Hilfeseiten von SeaMonkey in der Version 1.1.8, 2006.

[SUN07a] *How to Set the Look and Feel*. In: Creating a GUI with JFC/Swing (The Swing Tutorial) von Sun Microsystems, URL: `http://java.sun.com/docs/books/tutorial/uiswing/index.html`, [eingesehen: 14. März 2008], 2007.

[SUN07b] *How to Support Assistive Technologies*. In: Creating a GUI with JFC/Swing (The Swing Tutorial) von SunMicrosystems, URL: `http://java.sun.com/docs/books/tutorial/uiswing/misc/access.html`, [eingesehen: 16. März 2008], 2007.

[SUN08] *Java Accessibility Quick Tips - Ensuring and Verifying Basic Application Accessibility*. URL: `http://www.sun.com/accessibility/docs/java_access_tips.jsp`, , [eingesehen: 17. März 2008], 2008.

[Thompson04] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah. *XML Schema Part 1: Structures Second Edition*. Status: W3C Recommendation, URL: `http://www.w3.org/TR/xmlschema-1/`, [eingesehen: 10. Februar 2008], 28. Oktober 2004.

[TKG07] *Telekommunikationsgesetz (TKG)*. Ausfertigungsdatum: 22.06.2004. Telekommunikationsgesetz vom 22. Juni 2004 (BGBl. I S. 1190), zuletzt geändert durch Artikel 2 des Gesetzes vom 21. Dezember 2007 (BGBl. I S. 3198). URL: `http://www.gesetze-im-internet.de/tkg_2004/`, [eingesehen: 27. Februar 2008], 21. Dezember 2007.

[TMG07] *Telemediengesetz (TMG).* Telemediengesetz vom 26. Februar 2007 (BGBl. I S. 179). URL: `http://www.gesetze-im-internet.de/tmg/`, [eingesehen: 13. Januar 2008], 26. Februar 2007.

[Ulber04] Ulber, Peter. *P3P - Ein Überblick.* URL: `https://www.dergrossebruder.org/miniwahr/20041216163000.html`, [eingesehen: 7. Juni 2007], 2004.

[ULD08a] *JRC Personal Proxy – Installation & Browsereinrichtung.* URL: `https://www.datenschutzzentrum.de/selbstdatenschutz/p3p/p3p_jrcp.htm`, [eingesehen: 18. Februar 2008], 2008.

[ULD08b] *P3P mit Mozilla Browser 1.5 des Open Source-Projektes „mozilla.org“.* URL: `https://www.datenschutzzentrum.de/selbstdatenschutz/p3p/p3p_moz.htm`, [eingesehen: 18. Februar 2008], 2008.

[Ullenboom07] Ullenboom, Christian. *Java ist auch eine Insel – Programmieren mit der Java Standard Edition Version 6.* Online-Publikation, Galileo „openbook“, URL: `http://www.galileocomputing.de/openbook/javainsel7`, [eingesehen: 11. März 2008], November 2007.

[Wenning06] Wenning, Rigo, Schunter, Matthias, Cranor, Lorrie, Marchiori, Massimo, et al. *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification.* Hrsg.: Rigo Wenning und Matthias Schunter, Status: W3C Working Group Note, URL: `http://www.w3.org/TR/P3P11/`, [eingesehen: 29. Mai 2007], 13. November 2006.

[Wohlgemuth05] Wohlgemuth, Hans H., Gerloff, Jürgen. *Datenschutzrecht – Eine Einführung mit praktischen Fällen.* Luchterhand, München/Unterschleißheim, 3. Aufl., 2005.

A CD-ROM Inhalt

Die beiliegende CD-ROM hat folgenden Inhalt:

- Diese Ausarbeitung im pdf-Format.
- Der P3P-Policy-Generator als Archiv: *generator.zip*
- Der P3P-Policy-Generator als Archiv inklusive Quellcode und Javadoc-Dokumentation: *generatorSrc.zip*