7.8 Entwurf einer Rechtespezifikation

Dieses Kapitel betrachtet einige ausgewählte praktische Fragen zum Entwurf einer Rechtespezifikation.

7.8.1 Symbolische Prioritäten

Während des Entwurfsprozesses entsteht erst nach und nach der Bedarf für verschiedene Prioritäten.

Es ist daher praktisch, zunächst aussagekräftige, **symbolische Prioritäten** zu benutzen und die Ordnung zwischen diesen Symbolen explizit festzulegen.

Dadurch wird z.B. das Einfügen von neuen Prioritäten in diese bestehende Prioritätenmenge vereinfacht und ist dann i.a. ohne (textuelle) Änderung der bisher entworfenen Rechte möglich, insbesondere ohne Umnummerierungen der Prioritäten.

Nach Ende des Entwurfsprozesses sind eher wenig Rechteänderungen zu erwarten.

Nun möchte man (z.B der Rechteverwalter oder ein Benutzer für die Beurteilung seines Rechtestatuses) schnell die Priorität eines Rechts einschätzen können.
Dies geht schneller, wenn -wie in unseren Beispielen- eine gut bekannte, lineare Ordnung benutzt wird.
Im diesem Fall ersetzt man am Ende des Entwurfsprozesses die symbolischen Prioritäten durch eine bekannte Ordnung.

7.8.2 Klassen und Klassenhierarchie

Sind beim Entwurf nicht-disjunkte Klassen entstanden und bevorzugt man disjunkte Klassen, so kann man iterativ je zwei nicht-disjunkte Klassen A und B, durch die Klassen ersetzen, die durch die Mengen A–B, B–A, A∩B beschrieben werden, soweit diese Mengen nicht-leer sind.
(Allerdings müssen dann auch die Unterklassen von A und B entsprechend behandelt werden.)

Möchte man nur den gemeinsamen Elementen von A und B besondere Eigenschaften zuordnen können, bietet sich an, die durch A∩B beschriebene Klasse als gemeinsame Unterklasse der Klassen A und B zu modellieren.

Spezifizierte und hierarchiefreie Rechte fallen zusammen, wenn man die leere Hierarchie spezifiziert, d.h. alle Klassen (einer Kategorie) sind paarweise unvergleichbar.
Falls man also hierarchiefreie Rechte statt spezifizierter Rechte spezifizieren möchte (zumindest für eine Kategorie), so bietet die leere Hierarchie die Möglichkeit dazu.

7.8.3 Konfliktvermeidung

Bei linear geordneten Prioritäten können Konflikte trivialerweise vermieden werden, wenn man für jedes Recht eine andere Priorität benutzt.

Ebenfalls unkritisch ist es, wenn man Rechte mit gleicher Priorität, aber inverser Rechtekennung nur verwendet, wenn sie unter sich gegenseitig ausschließenden Bedingungen stehen (vgl. z.B. die Rechte für das Überholen im Verkehrsbeispiel), oder ganz verschiedene Teile der Klassenhierarchien bei zumindest einer Handlungskategorie betreffen.

Man könnte sich den Fall vorstellen, dass man durch ein neues Recht mit höherer Priorität einen Konflikt schlichten will. Fast immer ist es dann aber sinnvoll, die höhere Priorität direkt dem ursprünglichen Recht zuzuordnen, das jetzt bzgl. der überlappenden Handlungen dominieren soll.

Beispiel: Es besteht ein Konflikt zwischen den beiden Rechten

{(Verbot,20,Internist,o,g), (Erlaubnis,20,Chirurg,o,g)},

da anne zu beiden Klassen gehört. Diesen aktuellen Konflikt könnte man beheben, indem man ein drittes Recht

(Erlaubnis,30,anne,o,g)

einführt (allerdings damit einen latenten Konflikt belässt). Besser wäre es, die Priorität des Rechts

(Erlaubnis,20,Chirurg,o,g)

geeignet (d.h. i.a. geringfügig) hochzusetzen. Dabei muss allerdings überprüft werden, dass die Prioritätenerhöhung nicht noch weitere Auswirkungen hat.

Prioritäts-induzierte Konflikte kann man vermeiden, indem man die Ordnung auf den Prioritätsbereichen so wählt, dass unvergleichbare Prioritätsbereiche keine gemeinsamen Unterprioritätsbereiche haben.
Das ist trivialerweise der Fall bei linearen Ordnungen auf Prioritätsbereichen.

Konkret heißt das, dass unvergleichbare Prioritätsbereiche für verschiedene, disjunkte Teilwelten zuständig sein sollten.

7.8.4 Kleinste und größte Prioritäten: default-Werte und unüberwindbare Schranken

Wir haben bisher nicht festgelegt, ob es kleinste oder größte Prioritäten bzw. Prioritätsbereiche gibt.
Für einige Anwendungen liegt es jedoch nahe, solche zu verwenden.

Der Rechteansatz mit unüberwindbaren (Rechte-)Schranken (*mandatory access control*) legt z.B. nahe, zur Modellierung dieser Schranken die größte Priorität bzw. -wenn man zur Modellierung der Schranken mehrere Prioritäten benötigt- den größten Prioritätsbereich zu verwenden.

Umgekehrt können Standard-Rechte (sog. default-Werte) sehr einfach durch Verwendung der kleinsten Priorität bzw. des kleinsten Prioritätsbereichs modelliert werden.

7.8.5 Benutzergesteuerte Rechteverwaltung

In dem Ansatz der benutzergesteuerten Zugriffskontrolle (*discretionary access control*) ist jeder Eigentümer von Granulen gleichzeitig Rechteadministrator für die Rechte, die seine Granule betreffen.
Am einfachsten ist diese Situation modellierbar, indem jedem Eigentümer ein Prioritätsbereich zugeordnet wird.
I.a. sind dann die Prioritätsbereiche verschiedener Eigentümer unvergleichbar, nur der Prioritätsbereich des Systemadministrators ist größer als die der Eigentümer.

7.9 Rückblick

Unser Sicherheitsansatz kennt implizite Rechte für die Mitglieder von Klassen, fordert jedoch explizite Verbote und explizite Prioritäten.
Die objektorientierte Modellierung der Handlungen der (Anwendungs–) Welt erlaubt uns, die Rechte für Klassenmitglieder kurz und übersichtlich auszudrücken.
Mit Hilfe eines Prioritätensystems (zusammen mit expliziten Verboten) lassen sich mehrstufig allgemeine Regeln und Ausnahmen zu diesen Regeln kompakt beschreiben.

Die einfachen, total geordneten Prioritäten sind gut verwendbar, solange nur ein Rechteverwalter für die Konsistenz der Rechte verantwortlich ist.
Sind mehrere Rechteverwalter gleichzeitig tätig, so kann die unkoordinierte Verwendung von global gültigen Prioritäten verheerend wirken.
Insbesondere für diesen Fall haben wir Prioritätsbereiche eingeführt und die Wirkung eines Rechteverwalters auf seinen Gültigkeitsbereich beschränkt.

Darüberhinaus stellt sich heraus, dass mehrere Prioritätsbereiche auch für einen Rechteadministrator sinnvoll sind, um die Strukturierung der von ihm verwalteten Rechte zu erhöhen.
Diese Strukturierungsmöglichkeit erleichtert auch die Zuordnung der korrekten Prioritäten während des Entwurfs.
Für diesen Zweck haben sich außerdem auch symbolische lokale Prioritäten bewährt.

Wir haben das beschriebene Rechtesystem möglichst als Baukasten aufgebaut:

Wir können
- die Bedeutung der Klassenhierarchien und entsprechend der Bedeutung (und abhängig von der Rechtekennung) gleich- oder gegenläufige Rechtevererbung frei wählen,
- wahlweise jedes Objekt nur einer Klasse zuweisen oder mehreren (oder gar keiner),
- wahlweise jeder Klasse höchstens eine Oberklasse zuordnen oder mehrere,
- die Zustands- oder die Struktursemantik benutzen,
- Basiskonflikte oder aktuelle Konflikte betrachten.

Damit bleibt das Rechtesystem weitgehend konfigurierbar.

Wir haben dabei eine Gratwanderung gemacht zwischen der präzisen Darstellung der Vielzahl der Kombinationsmöglichkeiten und der suggestiven Darstellung der grundlegenden Ideen.
Die nicht für eine explizite Darstellung ausgewählten Kombinationsmöglichkeiten sollten aber vom Leser bei Bedarf problemlos ergänzt werden können.
So ließe sich z.B. auch eine "gemischte" Zustands- und Struktursemantik betrachten, bei der für einige Kategorien die Zustands- und für die restlichen Kategorien die Struktursemantik verwendet wird.

7.9.1 Pflichten und Freiheiten

Betrachtet man nicht nur statische Konzepte (wie z.B. Datenstrukturen), sondern auch dynamische Konzepte (wie z.B. Transaktionen), so möchte man oft nicht nur spezifizieren können, dass eine Handlung geschehen darf, sondern dass sie wirklich geschieht. Dies geschieht typischerweise so, dass einem Subjekt die **Pflicht** auferlegt wird, eine bestimmte Operation bei einem bestimmten Granul durchzuführen.

Syntaktisch haben wir es hier wieder mit einer Handlung zu tun, die mit einer neuen Rechtekennung als Pflicht gekennzeichnet wird.

Analog zu Erlaubnissen und Verboten wollen wir auch die Möglichkeit haben, die Abwesenheit von Pflichten festzulegen, also dass für eine Handlung keine Pflicht spezifiziert werden soll. Dieses Recht nennen wir **Freiheit**.

Die einzige syntaktische Erweiterung, die wir dafür in unserem Rechteansatz vornehmen müssen, ist die Erweiterung der Menge der Rechtekennungen:
TAG={Verbot, Erlaubnis, Freiheit, Pflicht}.

Die booleschen Operationen auf dieser Menge können weiterhin verwandt werden mit der Bedeutung
¬ Verbot=Erlaubnis, ¬ Freiheit=Pflicht.

Die Konflikte zwischen Pflichten und Freiheiten lassen sich dann ganz analog behandeln wie die zwischen Erlaubnissen und Verboten. Neu hinzu können allerdings Konflikte zwischen diesen beiden Gruppen kommen, z.B. Konflikte zwischen Pflichten und Verboten.

Auf ähnliche Art und Weise ließen sich ergänzend auch **Fähigkeiten** und **Unfähigkeiten** spezifizieren.