

Schriftliche Ausarbeitung von
Hanna Hakala

zum Thema

"Fuzzy-Ansatz und Fuzzy-Architekturen"

im Seminar

"Imperfektion und Datenbanken"

im Wintersemester 2003 an der Universität Karlsruhe

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis	2
2	Einführung	3
3	Einführung in die Fuzzy-Theorie	3
3.1	Fuzzy-Mengen	3
3.2	Fuzzy-Repräsentation.....	3
4	Interpretationen von Fuzzy-Mengen	4
4.1	Motivation zu den Fuzzy-Mengen.....	4
4.2	Vagheit vs. Unsicherheit, epistemische Interpretation	4
5	Das Fuzzy-Repräsentationssystem	5
5.1	Grundbegriffe.....	5
6	Fuzzy-Objektorientierung.....	5
6.1	Bekannte Fuzzy-Erweiterungen in der Objektorientierung.....	5
6.2	Dieser Ansatz	6
6.3	Definition des Ansatzes	6
7	Annotationen und Fuzzy-Facetten	7
7.1	Das Entwurfsmuster	7
8	Fuzzy-Bibliothek	8
8.1	Implementierung der Fuzzy-Mengen.....	8
8.2	Implementierung der Fuzzy Konjunktive Normalformen	9
9	Fuzzy-Referenzarchitektur	10
9.1	Fuzzy-Referenzarchitektur.....	10
10	Literatur	11

2 Einführung

Im Vergleich zu klassischen, scharfen Daten gewinnen imperfekte Daten zunehmend an Bedeutung für heutige Informationssysteme. Es existieren mehrere Ansätze, imperfekte (Fuzzy-) Daten in Datenbanken zu integrieren. Diese Seminararbeit befasst sich mit einem Ansatz aus René Wittes Dissertation "**Architektur von Fuzzy-Informationssystemen**" [1].

3 Einführung in die Fuzzy-Theorie

3.1 Fuzzy-Mengen

Zuerst muss der Begriff der Fuzzy-Menge definiert werden. Herr Witte hat ihn in seiner Dissertation folgendermaßen eingeführt:

"Eine Fuzzy-Menge μ von Ω ist eine Funktion von der Referenzmenge Ω in das Einheitsintervall:

$$\mu : \Omega \rightarrow [0, 1].$$

Die Menge aller Fuzzy-Mengen auf Ω sei mit $F(\Omega)$ bezeichnet."

Später wird μ als Zugehörigkeitsfunktion bezeichnet und die Möglichkeit dargestellt, klassische, scharfe Mengen als spezielle Fuzzy-Mengen zu begreifen. Hier wird eine Funktion mit einer Menge gleichgesetzt, was etwas ungeschickt wirkt. Korrekter definiert wäre die Fuzzy-Menge als eine Menge von Paaren $(x, f(x))$, mit x als Element aus der Grundgesamtheit und f als die Zugehörigkeitsfunktion.

Weitere wichtige Definitionen sind die des Schnittes, der Vereinigung und des Komplements:
Der Schnitt:

$$A \cap B = \mu_{A \cap B} = \min\{\mu_A, \mu_B\}.$$

Die Vereinigung:

$$A \cup B = \mu_{A \cup B} = \max\{\mu_A, \mu_B\}.$$

Das Komplement:

$$\neg A = \mu_{\neg A} = 1 - \mu_A$$

3.2 Fuzzy-Repräsentation

Es gibt zwei Arten, Fuzzy-Mengen zu repräsentieren: die vertikale und die horizontale Repräsentation. Die vertikale Repräsentation verwendet parametrisierte Dreiecksfunktionen, parametrisierte Trapezfunktionen oder sogenannte (parametrisierte) s/z-Funktionen. Die horizontale Repräsentation definiert die Fuzzy-Menge über einer Reihe von neuen Mengen, den sogenannten α -Schnitten. Jeder α -Schnitt besteht aus den Elementen der Referenzmenge, die mindestens mit dem Grad α in der Fuzzy-Menge enthalten sind (siehe Abb. 1.2). Durch die Verwendung von immer mehr α -Schnitten kann die Darstellung beliebig genau gemacht werden.

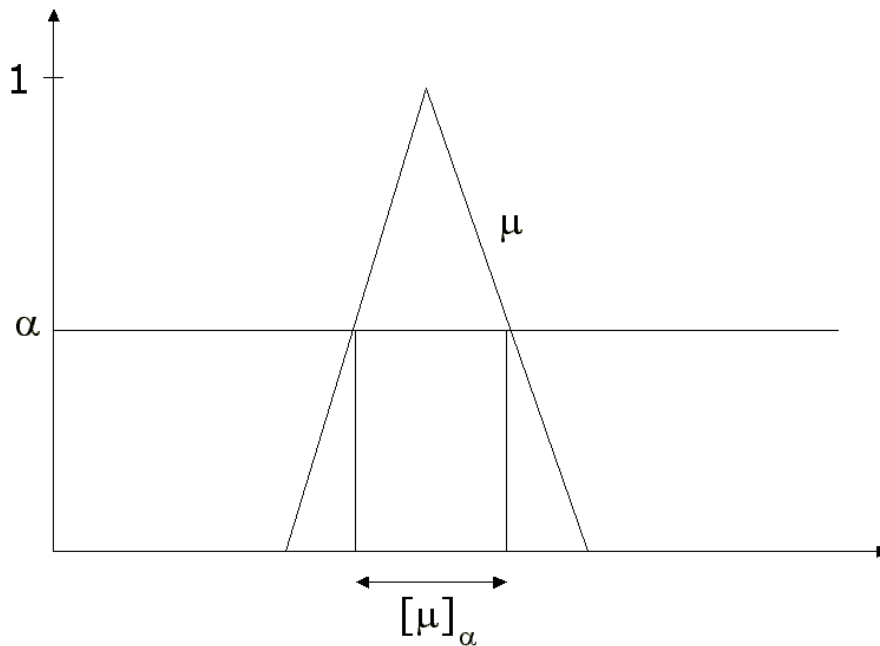


Abb. 3.2: α -Schnitt $[\mu]_\alpha$ einer Fuzzy-Menge μ

4 Interpretationen von Fuzzy-Mengen

4.1 Motivation zu den Fuzzy-Mengen

Ein natürlichsprachiger Begriff wie "große Person" lässt sich nicht zufriedenstellend mit klassischen Mengen beschreiben. Wenn die Grenze für eine "große Person" auf 1,7 m gesetzt wird, ist eine Person mit Größe 1,699 m schon nicht mehr groß – ein Millimeter mehr würde jedoch reichen, um aus ihr eine große Person zu machen. Mit Fuzzy-Mengen versucht man, zu diesem dem "gesunden Menschenverstand" widerstrebenden Konzept eine Alternative zu bieten. Weitere Beispiele, bei denen in der Praxis die Beschreibung durch eine Fuzzy-Menge eventuell derjenigen durch eine klassische Menge überlegen ist, sind "Kreditwürdigkeit einer Person" (für die Banken), "Umweltbelastung einer Region", "Staugefahr" (für ein Verkehrsleitsystem) und "Zuverlässigkeit eines Geschäftspartners".

4.2 Vagheit vs. Unsicherheit, epistemische Interpretation

Fuzzy-Mengen können unterschiedlich interpretiert werden. Es gibt zum einen das Konzept der *Vagheit*, wovon der Fall der großen Person ein Beispiel ist. Hier gibt der Zugehörigkeitsgrad an, wie kompatibel ein Wert aus der Grundmenge mit einem bestimmten Begriff ist. Zum Beispiel könnte (je nach Funktion) eine Person mit Größe 1,72 m zu einem Grad von 0,5 mit dem Begriff "große Person" kompatibel sein.

Bei der Modellierung von *Unsicherheit* hingegen weiß man, dass für gewisses Objekt (z. B. ein Sparschwein) ein unscharfer Begriff ("viel Geld enthaltend") gilt. Nun wird versucht herauszufinden, was "viel Geld" genau bedeuten könnte. 10 Euro wären wohl nicht viel, also setzen wir $\mu(10) = 0$. 100 Euro könnte man uneingeschränkt für "viel Geld" halten, also setzen wir $\mu(100) = 1$. Dazwischen entsteht dann ein gleitender Übergang von Werten, die mehr oder weniger "viel Geld" sind. Es ist zu beachten, dass es hierbei nicht um Wahrscheinlichkeiten geht; im Beispiel gilt für alle Summen x , die größer oder gleich 100

sind, $\mu(x) = 1$. Diese Interpretation einer Fuzzy-Menge nennt man auch *epistemische Interpretation*.

5 Das Fuzzy-Repräsentationssystem

5.1 Grundbegriffe

Ein *Fuzzy-Atom* ist eine syntaktische Beschreibung für eine Fuzzy-Menge und repräsentiert einen imperfekten Begriff wie "große Person" oder "teure Automarke". Die Abbildung μ definiert die Semantik eines Fuzzy-Atoms.

Der Schnitt und die Vereinigung zweier Fuzzy-Mengen und das Komplement einer Fuzzy-Menge wurden schon mit Hilfe der Zugehörigkeitsfunktion definiert. Diese Operationen sollen jedoch auch syntaktisch beschrieben werden, um die Herkunft einer Fuzzy-Menge festhalten zu können; es soll möglich sein, Berechnungen auf Fuzzy-Informationssystemen durchzuführen, die auf einer Menge zu einem Zeitpunkt verfügbarer Atomen aufbauen. Hierfür sollen syntaktische Ausdrücke über Atomen verwendet werden. Die Verknüpfungsoperatoren sind dabei die Negation, die Konjunktion und die Disjunktion.

Die Negation eines Fuzzy-Atoms A wird mit $\neg A$ dargestellt und seine Interpretation ist als

$$\mu_{\neg A} := 1 - \mu_A$$

definiert.

Ein *Fuzzy-Literal* L ist entweder ein Fuzzy-Atom A oder die Negation eines Fuzzy-Atoms $\neg A$.

Eine Fuzzy-Klausel K ist eine Oder-Verknüpfung von Fuzzy-Literalen:

$$K = L_1 \vee L_2 \vee \dots \vee L_n$$

Für die Fuzzy-Interpretation einer Fuzzy-Klausel werden die Fuzzy-Interpretationen der einzelnen Fuzzy-Literale vereinigt. Dies entspricht auf Syntaxebene der possibilistischen (epistemischen) Interpretation auf der Fuzzy-Mengenebene: kein Wert, der nach einem Fuzzy-Literal möglich ist, soll ausgeschlossen werden.

Eine Fuzzy-Formel F ist eine Und-Verknüpfung von Fuzzy-Klauseln:

$$F = K_1 \wedge K_2 \wedge \dots \wedge K_n$$

Die Fuzzy-Formel wird auch *Fuzzy Konjunktive Normalform* (FKNF) genannt. Für die Fuzzy-Interpretation einer Fuzzy-Formel werden die Fuzzy-Interpretationen der einzelnen Fuzzy-Klauseln geschnitten, was wieder der possibilistischen Interpretation der Fuzzy-Menge entspricht: Wenn die Fuzzy-Klauseln einzelne Anforderungen repräsentieren, die gleichzeitig gelten sollen, darf ein Ergebnis keinen Wert enthalten, der nicht mit allen Anforderungen wenigstens einigermaßen kompatibel ist.

6 Fuzzy-Objektorientierung

6.1 Bekannte Fuzzy-Erweiterungen in der Objektorientierung

Jetzt werden die verschiedenen, bereits existierenden Konzepte dargestellt, Objektorientierung Fuzzy-fähig zu machen.

Als "Fuzzy-Objekt" wird in vielen Fällen ein ganz gewöhnliches Objekt mit Fuzzy-Daten in Form von Fuzzy-Attributen verstanden. Eine andere Verwendung für den Begriff liegt vor, wenn damit ein Objekt, das nur teilweise zu einer Klasse gehört, bezeichnet wird. Hier wird also die Klassen-Instanz-Beziehung fuzzyfiziert.

Der Begriff der Fuzzy-Klasse hängt oft eng mit der genannten Klassen-Instanz-Beziehung zusammen. Er wird aber auch verwendet in Zusammenhang mit einem völlig anderen, konzeptionellen Modell, das mit Unsicherheit behaftet ist.

Die Verwendung von Attributen, um Fuzzy-Informationen aufzunehmen, leuchtet schon anhand der oft verwendeten Beispiele der Fuzzy-Logik ein: die Größe "1,75" einer Person würde man in klassischem Fall in einem numerischen Attribut speichern und für die Fuzzy-Größe "mittelgroß" würde man dann wohl Fuzzy-Attribute nehmen. Die Fuzzyfizierung von Attributen ist ein sehr übliches Konzept, Fuzzy-Techniken in Datenspeicherung einzubringen. Viele Fuzzy-Modelle verwenden hierfür einen neuen primitiven Typ oder eine neue Klasse "Fuzzy-Menge".

Fuzzyfizierung von Vererbung beruht auf die gleichen Überlegungen wie die von Objekten im Falle der Klassen-Instanz-Beziehung: es ist möglich, dass eine Klasse nur teilweise eine Unterklasse einer andere Klasse ist. Beispielsweise könnte man ein "Trike" verwenden, das nur mit Grad 0,6 ein Auto und mit Grad 0,8 ein Motorrad ist.

6.2 Dieser Ansatz

Es soll jetzt der Ansatz von Herrn Witte, Fuzzy-Daten in objektorientierte Umgebung zu bringen, in Beziehung zu den vorgestellten Konzepten dargestellt werden.

Für Fuzzy-Attribute lehnt Herr Witte aus Gründen der Interoperabilität einen neuen Datentyp "Fuzzy-Menge" ab. Ein Fuzzy-System soll auch eine gewöhnliche (klassische) Sicht von Daten für Systemteile, die mit Fuzzy-Daten nicht umgehen können, anbieten. Die Lösung von Herrn Witte verwendet zwar auch einen neuen Datentyp, die Fuzzy Konjunktive Normalform, der aber zusammen mit dem Konzept *Annotation mit Facetten* benutzt wird. Mit diesem Konzept ist es möglich für ein Datum verschiedene Facetten (franz. "kleines Gesicht") zu definieren, die z. B. einen klassischen Integer-Wert und einen Fuzzy-Begriff speichern.

Fuzzy-Klassen und Fuzzy-Vererbung auf die dargestellte Art zu verwenden, ist um Einiges problematischer. Solche Änderungen wären erstens nicht kompatibel mit existierenden Laufzeitumgebungen; zweitens würde man hier fuzzy als Selbstzweck verwenden, anstatt bessere und robustere Systeme damit zu realisieren. Die für Fuzzy-Klassen und Fuzzy-Vererbung beschriebenen Konzepte bewegen sich auf einer anderen Abstraktionsebene als das Ziel: sie würden das Fuzzy-Modell an sich unsicher machen, obwohl das Modell nur unsichere Daten repräsentieren sollte.

6.3 Definition des Ansatzes

Das Modell erlaubt es, Strukturkomponenten des objektorientierten Modells zu annotieren. Eine Annotation bildet einen Behälter, in dem eine oder mehrere Facetten eines konkreten Typs Zusatzinformationen speichern können. Eine Facette kann dann z. B. eine Fuzzy Konjunktive Normalform aufnehmen.

Attribut-Annotationen kann man anlegen für

- einfachwertige Attribute, die einen primitiven Datentyp aufnehmen. Eine Fuzzy Konjunktive Normalform könnte man hier beispielsweise für einen voraussichtlichen Wert des Attributs oder für eine einschränkende Bedingung verwenden.

- mehrwertige Attribute, also Vektoren, Strukturen, Felder. Hier gibt es mehrere Möglichkeiten, wie eine Annotation angelegt werden kann: als eine einzelne Fuzzy

Konjunktive Normalform, die Fuzzy-Daten über die ganze Struktur enthält oder als mehrwertige Annotation, die jeweils einen Fuzzy-Datum für einen Element des Strukturs behält.

-objektwertige Attribute, Attribute die Referenzen auf Objekte enthalten. Eine Verwendung für eine Fuzzy-Annotation könnte z. B. eine Qualitätsaussage über die Referenz sein.

Es soll außerdem möglich sein, zusätzlich zu einfachen Attributen Attributmengen zu annotieren. Attributen sollen nicht nur einzeln, sondern in Beziehung zu einander annotierbar sein. Diese Annotationen sollen eine Art ad-hoc-Beziehung zwischen den Daten zustande bringen. Ein paar Beispiele verdeutlichen, was hiermit gemeint ist: eine vage Eigenschaft "gut verschweißbar" bezieht sich immer auf zwei Materialien. Im Modell wird also eine Attributmenge mit den beiden Materialien der beteiligten Objekte definiert. Diese Attributmenge wird annotiert, und eine Facette kann die Information über die Verschweißbarkeit annehmen. Im Bereich der Architektur findet sich eine ähnliche Konstellation mit der Beschreibung "farblich passend", hier könnte die Attributmenge auch aus mehr als zwei Attributen bestehen.

Objekte als Ganzes können auch annotiert werden. Es können dann zusätzliche textuelle Informationen abgespeichert werden, wie "erster Entwurf" oder Bedingungen wie "muss persistent gespeichert werden" oder Metainformationen über z. B. die Wichtigkeit des Objektes.

7 Annotationen und Fuzzy-Facetten

7.1 Das Entwurfsmuster

Der Ansatz wird technisch als Entwurfsmuster umgesetzt (siehe Abb. 7.1). Die Struktur von Objekten soll dynamisch erweitert werden, um Metainformationen einzusetzen. Die Teile des objektorientierten Datenmodells, für die Metainformationen existieren sollen, nennt man Strukturkomponenten. Annotationen werden an diese Strukturkomponenten aufgehängt. Eine Annotation bildet den Container zur Verwaltung einzelner Facetten.

Es werden bei diesem Muster die Entwurfsmuster Kompositum (AnnotationComposite) und Dekorierer (FacetDecorator) verwendet. Mit Kompositum lassen sich auch zusammengesetzte Strukturen annotieren und Dekorierer dient dazu, ein Objekt dynamisch mit zusätzlicher Funktionalität zu erweitern. Die Art der einzusetzenden Metainformation wird durch die vom Benutzer des Musters definierten konkreten Facetten (ConcreteFacet) bestimmt.

AnnotatableObjekt ist die Schnittstelle für ein annotierbares Objekt. Man lässt die Objekte, die annotierbar sein sollen, von dieser Klasse erben.

AnnotatedStructure beinhaltet die vom Benutzer definierten Blätter des Kompositums, beispielsweise Klassen wie "ObjektAnnotation" und "AttributeAnnotation", wovon die Letztere noch in Unterklassen "SingleValued" und "MultiValued" aufgeteilt werden könnte.

FacetDecorator erweitert die Annotationen um Facetten. FacetDecorators Unterklassen definieren die konkreten "Gesichter", wie z. B. eine "StringFacet" oder eine "FKNF-Facet".

Die verschiedenen Iteratoren erlauben es, sowohl die Komponenten einer zusammengesetzten Annotation als auch die Facetten der Komponenten durchzulaufen.

Mit diesem Entwurfsmuster ist es also möglich, Fuzzy-Daten mit Hilfe von Fuzzy-Facetten in objektorientierte Umgebung zu integrieren. Als nächstes wird eine Fuzzy-Bibliothek für die Speicherung der Fuzzy-Daten eingeführt.

Annotations- Entwurfsmuster

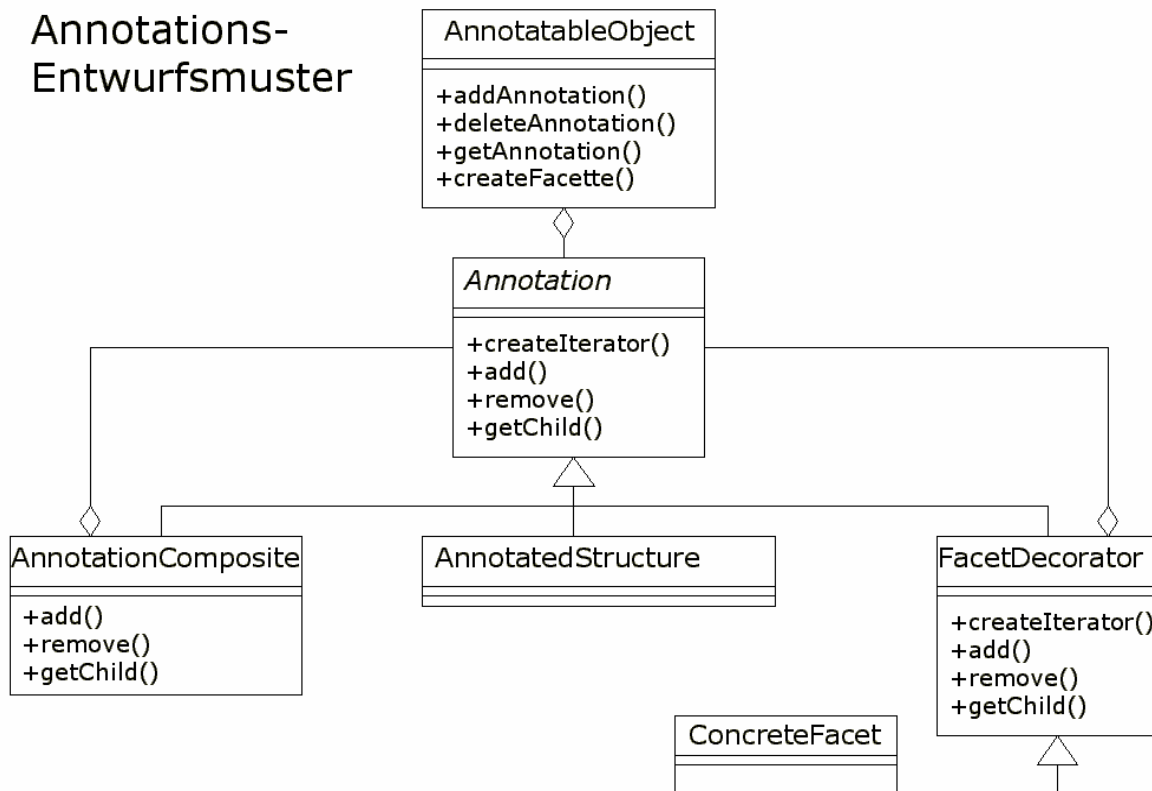


Abb. 7.1: Annotationsentwurfsmuster

8 Fuzzy-Bibliothek

8.1 Implementierung der Fuzzy-Mengen

Zuerst muss eine Struktur für Fuzzy-Mengen und ihre verschiedene Repräsentationen entworfen werden. Hier wird der Entwurfsmuster Brücke verwendet, um eine Fuzzy-Mengen-Schnittstelle von der Implementierung zu trennen.

HFuzzySetImp bietet eine Implementierung der horizontalen und VFuzzySetImp eine Implementierung der vertikalen Repräsentation von Fuzzy-Mengen. Beide Klassen stellen für die Fuzzy-Menge die Operationen *and*, *or* und *not* zur Verfügung; die Klasse LevelCut stellt den α -Schnitt bereit (siehe Abb. 8.1).

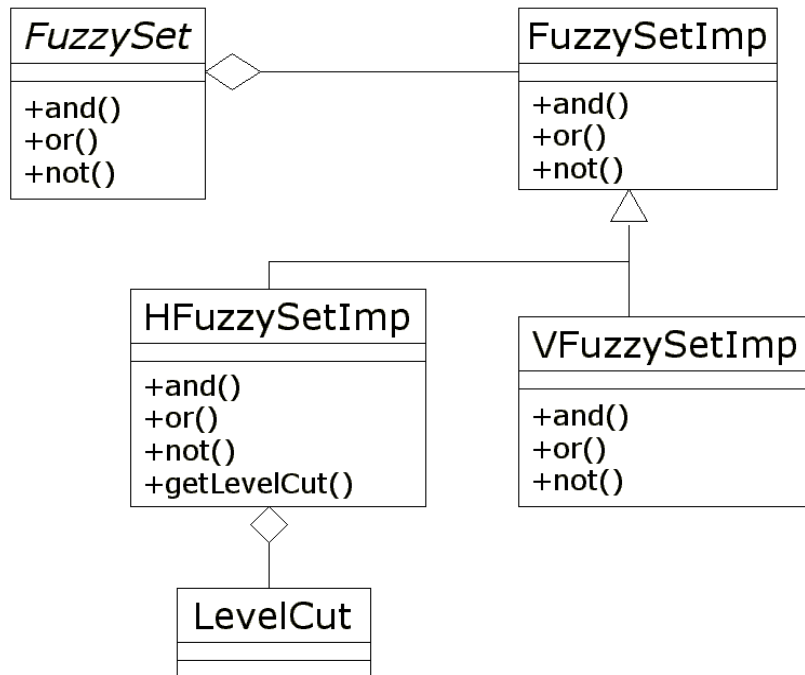


Abb. 8.1: Fuzzy-Mengen-Implementierungsstruktur

Ein Code-Beispiel verdeutlicht die Verwendung dieser Klassen. Ein fiktiver Reengineering-Fall liefert als Fuzzy-Menge den Typ einer nicht näher bekannten Variable i mit μ_i vom $\text{Typ}(\text{"int"}) = 1,0$; μ_i vom $\text{Typ}(\text{"string"}) = 0,25$; μ_i vom $\text{Typ}(\text{"float"}) = 0,5$; μ_i vom $\text{Typ}(\text{"bool"}) = 0,25$.

```

double[] typLevels = {0.25, 0.5, 0.75, 1.0};
HFuzzySetImp typImp = new HFuzzySetImp(typLevels);

// Werte der  $\alpha$ -Schnitte eintragen
typImp.multipleAddValue(1.0, 'int');
typImp.multipleAddValue(0.25, 'string');
typImp.multipleAddValue(0.5, 'float');
typImp.multipleAddValue(0.75, 'bool');

FuzzySet typFuzzySet = new FuzzySet(typImp);
  
```

8.2 Implementierung der Fuzzy Konjunktive Normalformen

Der Entwurf der Fuzzy Konjunktiven Normalform verwendet auch das Entwurfsmuster Kompositum, um Fuzzy-Literale, Fuzzy-Klauseln und Fuzzy-Formeln voneinander zu trennen und trotzdem als Teile der Fuzzy Konjunktiven Normalform gleich zu behandeln.

FuzzyComponent definiert die Schnittstelle für alle Objekte der Komposition. FuzzySet ist das Blatt der Komposition, hier eine einzelne Fuzzy-Menge. FuzzyComposite ist eine abstrakte Klasse und die Oberklasse für die konkreten Klassen Literal, Clause (Klausel) und Formula (FKNF, Formel). Diese Klassen repräsentieren die Fuzzy-Literale, Fuzzy-Klauseln und Fuzzy-Formeln (siehe Abb. 8.2).

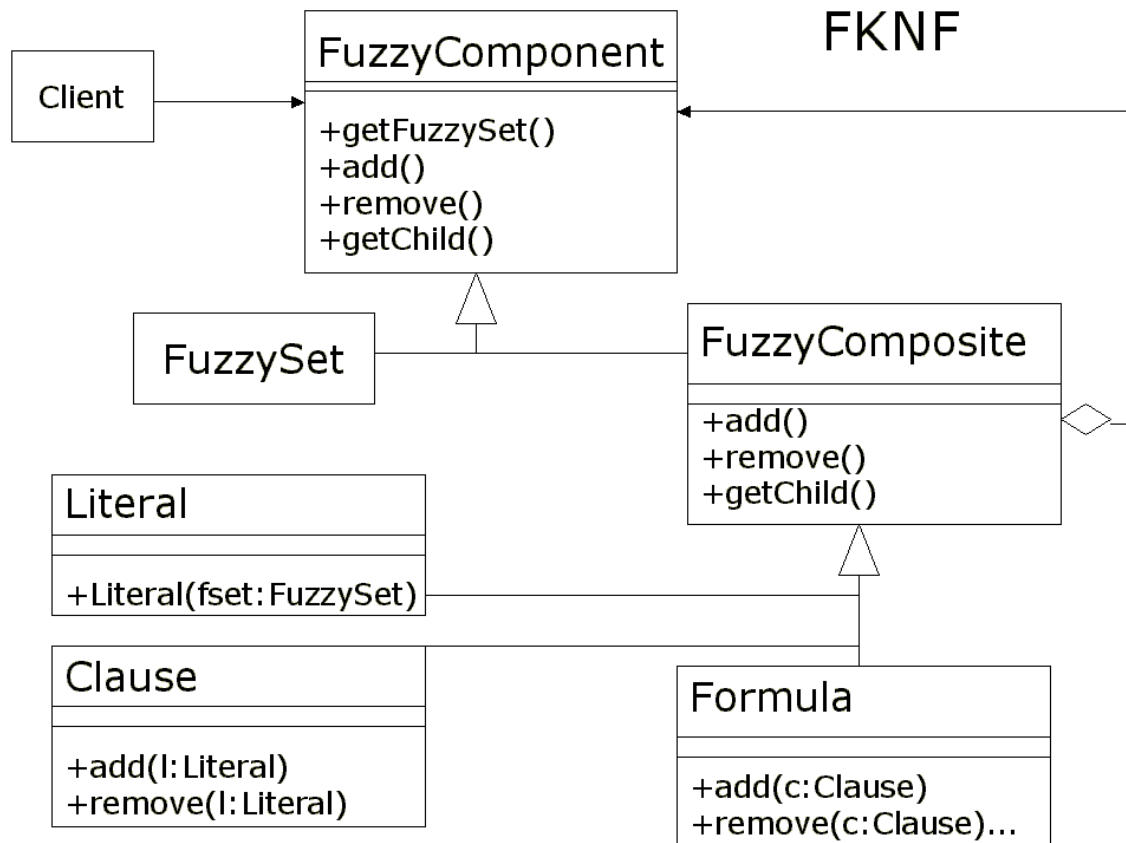


Abb. 8.2: Modellierung von Fuzzy Konjunktiven Normalformen

Mit Hilfe der Entwurfsmuster und diesen zwei Strukturen können jetzt Fuzzy-Daten in objektorientierter Umgebung verwendet werden.

9 Fuzzy-Referenzarchitektur

9.1 Fuzzy-Referenzarchitektur

Auf Basis einer Informationssystem-Referenzarchitektur soll jetzt eine Referenzarchitektur für Fuzzy-Systeme entworfen werden. Die notwendigen Änderungen und deren Gründe für die verschiedenen Stufen der Referenzarchitektur werden dargestellt (siehe auch Abb. 9.1).

Die Klienten-Stufe ermöglicht den Zugang zu einem Fuzzy-Informationssystem. Um den Änderungsbedarf auszuarbeiten, müssen zwei verschiedene Arten von Klienten unterschieden werden: autonome und nicht-autonome. Für nicht-autonome Klienten ergibt sich kein Änderungsbedarf, da für sie die Präsentationsstufe für die korrekte Präsentation der Fuzzy-Daten verantwortlich ist.

Autonome Klienten müssen erstens dem Benutzer Fuzzy-Information verständlich präsentieren, wofür es einer Fuzzy-Präsentationskomponente bedarf. Zweitens müssen sie vom Benutzer imperfekte Information in Form von imperfekten Begriffen (keine Fuzzy-Mengen!) entgegennehmen. Diese Information muss auch in weiteren Stufen zur Verfügung stehen: in der Präsentationsstufe für nicht-autonome Klienten und in der Geschäftsprozessstufe für Berechnungen. Hierfür wird eine neue Ressource namens *Fuzzy-Lexikon* bereitgestellt, in das in der Dissertation genauer eingegangen wird.

Die Präsentationsstufe muss sich um nicht-autonome Klienten kümmern ähnlich wie es die Klientenstufe bei den autonomen Klienten macht. Hier sind die Darstellungsmöglichkeiten jedoch eingeschränkt je nach Fähigkeit des Klienten. Die Präsentationsstufe übernimmt die Aufbereitung und Kontrolle der Benutzerinteraktion. Hier wird die Fuzzy-Bibliothek benötigt. Für Web-Klienten braucht man eine *Fuzzy-HTML*-Komponente, die die Umwandlung zwischen Fuzzy-Daten und (D)HTML vornimmt.

Die Geschäftsprozessstufe, die die Hauptfunktionalität enthält, benötigt alle Konzepte, die bisher vorgestellt wurden: Fuzzy-Formeln, Verarbeitungsoperationen, Annotationen und Fuzzy-Facetten. Hinzu kommen Änderungen in dieser Stufe bei Kommunikation der annotierten Objekte mit der Ressourcenstufe: bei Speicherung imperfekter Informationen in einer Datenbank, Aufruf von externen Diensten (Defuzzifizierung) und bei Datenaustausch mit anderen Systemen (Defuzzifizierung, geeignetes Austauschformat).

In der Ressourcenstufe kann man entweder das existierende Informationssystem mit den vorgestellten Fuzzy-Technologien erweitern oder die Aufrufe aus der Geschäftsprozessstufe anpassen. Stellt die Ressource ein DBMS dar, muss das Schema eventuell auch angepasst werden. Eine neu hinzukommende Ressource ist der schon genannte Fuzzy-Lexikon-Dienst, die von der dritten Stufe benutzt wird.

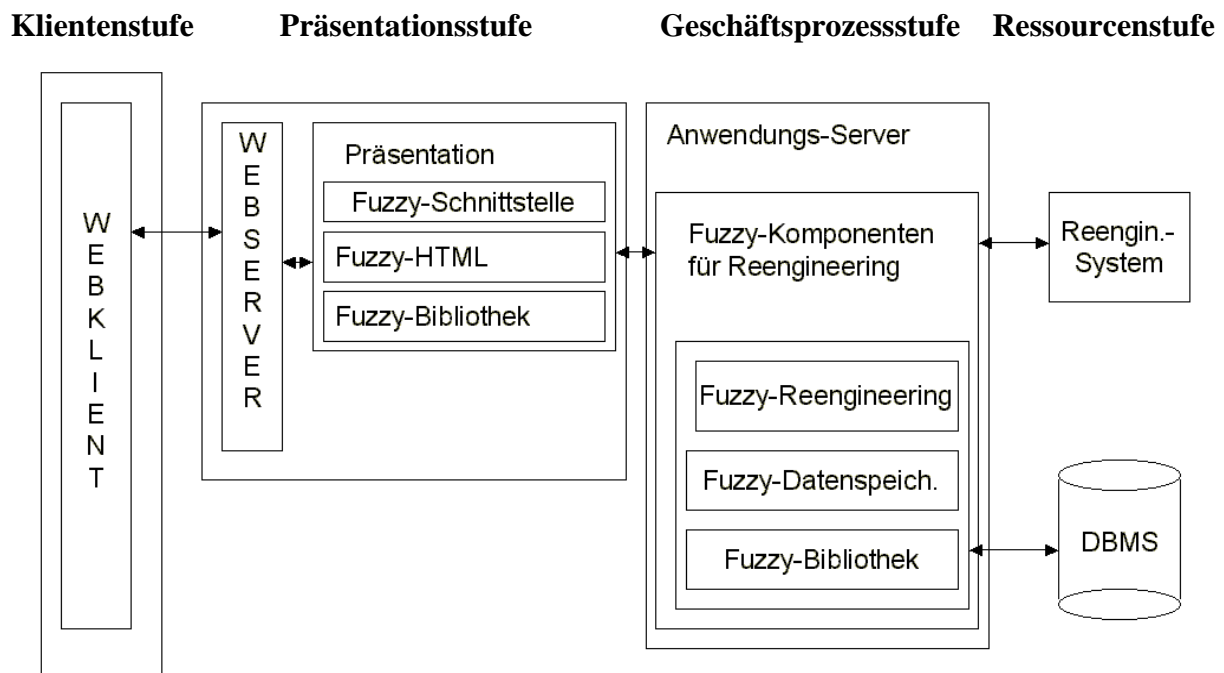


Abb 9.1: Architektur von Fuzzy-Informationssystemen

10 Literatur

- [1] Witte, René: Architektur von Fuzzy-Informationssystemen. Dissertation, Universität Karlsruhe (TH); BoD, Norderstedt, Germany, 2002