

## Gehören Dimensionen zur „langue“ (Sprache) oder zur „parole“ (Rede)?

### -Sprachphilosophische Bemerkungen-

Vom Schweizer Linguisten Ferdinand de Saussure (1857-1913) stammt die fundamentale Idee, zwischen „langue“ (Sprache) und „parole“ (Rede) zu unterscheiden. Statt „langue“ wurden später auch die Ausdrücke „Schema“ oder „Typ“ bzw. dann bei Chomsky der Ausdruck „competence“ eingeführt. In jedem Fall ist ein universeller und kein partikularer Aspekt eines sprachlichen Gegenstandes gemeint. Universelle und partikulare Aspekte werden in der Regel je nach linguistischem Modell als Begriffspaar eingeführt: langue/parole, Schema/Ausprägung (engl. „instance“, im Deutschen aber auch Aktualisierung oder Instanziierung), type/ token seit Peirce (1839-1914) und competence /performance seit Chomsky (\*1928).

Die Suche nach vollständigen sprachlichen Ausdrücken führte dann in der Linguistik zu den Begriffen „Wort“ und „Satz“, und man ist sich einig, dass Wörter zur „langue“, zum Schema oder Typ, und Sätze zur „parole“, zur Rede oder Ausprägung einer Sprache gehören. Die Wörter einer Sprache können als Schema in einem Wörterbuch (Lexikon) erfasst und manchmal auch erklärt (Referenzsemantik) werden. Bei Sätzen, die zur Ausprägung gehören, ist das nicht möglich. Es gibt kein „Sätzebuch“, allenfalls eine Satzschemelehre, z.B. des Deutschen, in der über typische Satzkonstruktionen wissenschaftlich geredet wird. Eine für die Informatik wichtige Satzschemelehre ist aber das Relationen –Modell (Ted Codd, 1923-2003) der Datenbanksysteme. Es gibt in dem Modell, immer streng getrennt, eine Benennung von Gegenständen (Primärschlüssel) und Aussagen über die benannten Gegenstände. Als „Sätzebuch“ könnte man natürlich ein Relationenbuch über alle z. B. in Deutschland verfügbaren Datenbanken -Schemata aufstellen. Eine Sisyphus-Arbeit, ähnlich wie die Erarbeitung eines Wörterbuches à la Duden. Das À-Jour-Halten ist organisatorisch aber noch fürchterlicher als das Erstellen.

Man sagt auch, Wörter haben als Schema einen signifikativen Aspekt, im Gegensatz zu den Sätzen, denen als Ausprägung ein kommunikativer Aspekt zugeschrieben wird. Über Ausprägungen wird kommuniziert.

Wir betrachten z.B. das Wort „Gewicht“. Es ist für seine schematische Erfassung in einem Lexikon unüblich, auch noch auf seine möglichen Dimensionen, z.B. in kg oder britischen Pounds hinzuweisen. Dimensionen gehören nicht zum Wortschema, sondern zum Verwendungsschema. Erst wenn über „Gewicht“ in Aussagen geredet wird, wird seine Dimension bedeutungsvoll. Betrachten wir unsere klassischen imperativen Programmiersprachen noch aus der nicht-objektorientierten Zeit. Wir finden z.B. in einer Programmdeklaration den Ausdruck „float gewicht“, um darzutun, dass Gewicht als Dezimalzahl abgespeichert wird. Es wird etwas benannt, besagt wird überhaupt nichts. Würde man schreiben „kg gewicht“, so wird besagt, dass die Eigenschaft Gewicht in kg anzugeben ist. Linguistisch schreibt man das wie folgt: „ $\delta$  gewicht“, wenn nur etwas benannt wird, also nur als sprachlicher Namen für etwas angesehen wird (z.B. Gewicht als Bestandteil einer Waage), und „ $\varepsilon$  gewicht“, wenn nur etwas besagt wird (z.B. als Eigenschaft eines Körpers). Das steht so bei Kuno Lorenz „Dialogischer Konstruktivismus“, de Gruyter, 2009 (Seite 48). „ $\varepsilon$  gewicht“ verlangt natürlich nach einem Gegenstand, über den etwas besagt werden soll und der hier mit ‚x‘ benannt wird, also: „ $x \varepsilon$  kg\_ gewicht“. Ohne Benennung von Gegenständen kann kein Bezug zur Welt hergestellt werden. Für den Programmierer in den klassischen Sprachen, aber auch für den Datenbankler der eine Relation „Teil (Teilenummer, kg\_ gewicht, ..)“ hinschreibt, wird ein Standardtyp (float, integer, char..) oder ein benutzerdefinierter Typ eingeführt. Z.B. könnte es heißen:

*Create User Defined Type kg\_ gewicht As float.*

Benutzer Definierte Typen werden auf vorgegebene Typen zurückgeführt.

Datentypen, die nur benennen wie „Teilenummer“, oder besser als „ $\delta$ Teilenummer“ geschrieben, bedürfen gar nicht einer Benutzerdefinition. Sie besagen ja nichts. Sie müssen nur eindeutig sein, was in der Regel von einem System gewährleistet wird. Das Thema „Dimension“ wird unter dem Thema „benutzerdefinierte Datentypen“ abgehandelt.