

Klaus Prätor

Kartierung - einige Grundkonzepte

Dies ist der Versuch, einige informationstechnische Begriffe und Modelle, die für unser Kartierungsprojekt von Wichtigkeit sind, in einer möglichst einfachen und voraussetzungsfreien Weise darzustellen. Die Betonung liegt auf möglichst ;-)

Kartierung & GIS

Unter "Kartierung" verstehe ich die Erfassung, Handhabung und Auswertung von Informationen über Objekte (Gegenstände oder Prozesse) mit einem geographischen Bezug, d.h. solche, die man in eine Karte *eintragen* kann. (Andere verstehen unter Kartieren (bzw. mapping) auch das Erstellen der Karten selbst.)

Eine Karte kann physisch existieren, zum Beispiel auf Papier, oder virtuell, als Software. Aus informatischer Sicht ist eine Kartierung ein geographisches Informationssystem (GIS). Dieses besteht aus Karte und zugehöriger Datenbank. Dabei ist zu unterscheiden zwischen einer konkreten Datenbank und der Software, die diese handhabt: das Datenbankmanagementsystem (DBMS). Bei Karten ist es entsprechend. Karten wirken attraktiver, weil anschaulicher. Datenbanken dagegen klingen nach Bürokratie. Die wesentliche Aufgabe bei der Kartierung ist aber der Entwurf und dann natürlich die "Füllung" der Datenbanken, denn die Karten gibt es ja in der Regel schon.

Datenbanken

Eine Datenbank (DB) können wir uns als ein Tabellen-System vorstellen.

Relationale Datenbanken (es gibt auch andere, aber dies sind die verbreitetsten) bestehen tatsächlich aus Tabellen. Ehe wir eine konkrete Datenbank füllen können, müssen wir die Struktur dieser Tabellen und eventuell auch das zu verwendende Vokabular festlegen.

Wir würden für ein Projekt zumindest *Name*, *Ort*, *Adresse*, und zum Beispiel *Bedürfnis*, *Interaktionsform* oder *politische Identität* eintragen - und natürlich geografische Koordinaten (*Länge*, *Breite*). Dafür müssen wir jeweils ein Feld, bzw. eine Spalte reservieren. Über die geogr. Koordinaten wird die Verbindung zur Karte hergestellt, d.h. die Kartensoftware sorgt dafür, dass das Projekt an der entsprechenden Stelle eingetragen wird. (In Wirklichkeit würden wir wohl mehr Felder brauchen und die Adresse aufspalten in PLZ, Straße, Hausnummer etc. Aber das Beispiel soll übersichtlich bleiben und die Funktion bleibt bei allen Feldern dieselbe. Diese Darstellungsweise nenne ich die relationale Form.

Das Ganze geht nämlich auch anders. Es ist wichtig, diese andere Variante zu kennen, weil sie auch in OpenStreetMap verwendet wird. Statt für jede Kategorie eine eigene Spalte zu benutzen, nehmen wir (zunächst) nur zwei Spalten und schreiben in die eine den Attributnamen (die Kategorie), also z.B. *Name*, *Ort*, *Breite*, *Tätigkeitsbereich*, und in die andere den Wert des Attributs: also z.B. *Leila*, *Berlin*, *52.4827*, *Leihladen*.

In der Informatik (aber auch in der Logik) spricht man hier von Attributen eines Objekts (in unserem Fall eines Projektes) - und nennt deshalb diese Variante auch objektorientierte Darstellung. Statt von Attributnamen und -wert wird oft auch von key/value - Paaren gesprochen. Um die Attribute dem jeweiligen Objekt zuzuordnen

müssen diese Paare in der Datenbank um einen eindeutigen Bezeichner oder Identifikator für dieses Objekt ergänzt werden. Jede Zeile der Tabelle hat dann drei Felder: Objekt, Attribut(sname), Wert - ein Tripel. (Ein Tripel oder 3-Tupel ist für drei das, was ein Paar für zwei ist.) Wenn wir für unser Beispiel Leila als Identifikator die Webadresse von Leila nehmen (sie ist garantiert einmalig), ergibt sich Folgendes:

Objekt	Attribut	Wert
www.leila-berlin.de/	Name	Leila
www.leila-berlin.de/	Ort	Berlin
www.leila-berlin.de/	Breite	52.4827
www.leila-berlin.de/	Tätigkeitsbereich	Leihladen

(Mögliche) Komplikationen und Komplexität

All das kann auch komplizierter werden, sowohl hinsichtlich der Schwierigkeiten wie auch der Lösungsmöglichkeiten. Dazu ein wichtiger Vorspruch:

Keep it simple!

Wenn ich im Folgenden auf zunehmend komplexere Möglichkeiten eingehe, heißt das nicht, dass wir (d.h. unser Kartierungsprojekt) das alles (gleich) machen sollen. Ganz im Gegenteil bin ich sehr dafür, mit überschaubaren, machbaren Strukturen anzufangen - auch wenn die nicht alles Gewünschte abbilden können. Zugleich meine ich, dass es sinnvoll ist, sich bereits am Anfang mit den Komplexitäten vertraut zu machen, um sich nicht durch falsche Anfangsentscheidungen spätere Erweiterungs- oder Optimierungsmöglichkeiten zu verbauen.

Zugriff auf unsere Karte über verschiedene "Klassifikationsachsen"

("Facettenklassifikation" nennen das die Dokumentare.) So schrieb Silke in einer Mail (sinngemäß): "Wir brauchen eine 3-fach überlappende Struktur, die den Zugriff über Bedürfnisse, Interaktionsform und politische Identität erlaubt." Das ist kein Problem. In der relationalen Form wird jeweils ein neues Feld in die Tabelle eingefügt. In der objektorientierten fügen wir für jeden neuen Aspekt, den wir auf der Karte abbilden wollen, eine neue Zeile an. wo z.B: *Bedürfnis (fulfills_needs)*, *Interaktion (mode_of_interaction)* und *politische Identität (political_identity)* als Attributname und daneben der entsprechende Wert eingetragen wird.

Mehrfachzuordnungen

Brigitte hatte diese Frage in einer Mail aufgeworfen: Mitunter soll ein Projekt / Objekt mehrfach zugeordnet werden. Die Kommune Niederkaufungen z.B. ist Tagungshaus, Kindertagesstätte, Altenbetreuung, Werkstatt, Bioladen, Landwirtschaft etc. Sie hat also mehrere Tätigkeitsfelder, bzw. erfüllt mehrere Bedürfnisse. Das sollte natürlich eingetragen werden, ist aber schon nicht mehr ganz so einfach. In einer "kunstgerechten" relationalen Datenbank ist dafür eine eigene Tabelle mit zwei Spalten anzulegen. In jeder Zeile enthält ein Feld *eines* der Tätigkeitsfelder, das andere einen *Schlüssel* für das Projekt, d.h. einen Ausdruck, der sich auf genau einen Datensatz der Ursprungstabelle bezieht, z.B. den Namen.

Exkurs: OpenStreetMap

Michael hat uns gesagt, dass in OpenStreetMap Mehrfachwerte dargestellt werden können. Das stimmt im Ergebnis, ist aber nicht die

ganze Wahrheit. Diese sieht etwa so aus: Eigentlich können Mehrfachwerte **nicht** dargestellt werden. OpenStreetMap benutzt nämlich für die Darstellung seiner Objekt-Attribut-Wert-Tripel eine relationale Datenbank (Postgres), die in jedem Feld nur *einen* Wert aufnehmen kann/soll. Der Aufbau zusätzlicher Tabellen ist in OSM nicht vorgesehen. Es gibt aber einen "Trick": Wir fügen den Attributwert, durch ':' getrennt, an den Attributnamen an, also z.B.

Bedürfnis:Kinderbetreuung, und tragen im Feld für den Wert nur den so genannten logischen Wert 'ja' oder 'nein' ein. Ihr könnt hier sehen, wie wir das in unserer Taxonomie machen:-

https://wiki.openstreetmap.org/wiki/User:KartenKarsten/futureMap#Needs_fullfilled_by_a_community

Das löst das Problem, allerdings um den Preis eines schwierigeren Umgangs mit der Datenbank, z.B. muss der Wert, den ein Programmierer eigentlich im Wertefeld vermutet, jetzt als ein Teilausdruck im Attributsfeld gesucht werden.

Hierarchien (Taxonomien)

Manche Betriebe (z.B. Karlshof Berlin) liefern nur Kartoffeln, ich würde sie aber auch gern finden, wenn ich nach Gemüse suche. Eine Lösungsmöglichkeit dafür sind hierarchische Klassifikationen oder Taxonomien. (Unter Taxonomien werden in der Regel hierarchische Klassifikationen verstanden wie z.B. die der biologischen Arten. Man kann aber auch, wie in unserem Projekt, Klassifikationen mit nur einer Ebene als einfache Taxonomie auffassen. Der Begriff selbst besagt, dass die Klassifizierung (Taxa) gewissen Regeln (Nomoi) folgt.)

Eine kleine Beispielhierarchie:

Kartoffeln ArtVon Gemüse.

Blumenkohl ArtVon Feingemüse.

Feingemüse ArtVon Gemüse.

Gemüse ArtVon Nahrungsmittel.

Ei ArtVon Nahrungsmittel.

Sie steht neben den eigentlichen Projektdaten. Mit ihrer Hilfe lässt sich, neben dem Blumenkohl, auch das Kartoffelkombinat finden, wenn ich nach Gemüse suche. Neben geeigneten Programmen ist eine Datenstruktur erforderlich, die die Hierarchiebeziehungen ausdrücken kann. In einer relationalen Datenbank wären das spezifische Tabellen für diesen Zweck.

Exkurs: OpenStreetMap

Da keine separaten Tabellen zur Verfügung stehen, werden auch Hierarchien in OSM nicht "kunstgerecht" dargestellt, sondern wieder mit "Trick" Sie lassen sich wiedergeben durch ':'-Trennung im Attributnamen. Das ist aber keine wirkliche Datenstruktur. Außerdem hat dieser Doppelpunkt verschiedene Funktionen (z.B. die vorhin erwähnte Trennung von Attributname und -wert), was für den programmierenden Umgang mit ihm äußerst unangenehm ist. Denn Programmierer lieben Eindeutigkeit - und das nicht ohne Grund.

Semantic Web

Eine Alternative zu Taxonomien bzw. hierarchischen Klassifikationen sind **Thesauri**, sozusagen strukturierte Wörterbücher. Damit kommen wir zu dem wichtigen Thema des Vokabulars, und wollen es behandeln im Zusammenhang eines aktuell sehr beachteten Themas: dem Semantic Web. Texte, die für uns Menschen vielfältig strukturiert und

komplex sind, waren für die Informatik zunächst einfach "Fließtext". Das heißt: unstrukturierte Zeichenketten, mit denen der Computer nichts anfangen konnte, weil ihm die Bedeutung (Semantik) nicht zugänglich war, es sei denn über spezielle linguistische Software. Das änderte sich etwas mit dem Aufkommen von Textauszeichnung (markup), insbesondere mit HTML, der Auszeichnungssprache des Web. Überschriften, Hervorhebungen, Listen gaben den Texten eine auch maschinenlesbare Struktur. Mit XML konnte man darüber hinaus Elementen beliebige Bedeutungen zuordnen. Tim Berners-Lee, der uns nahe steht, weil er das World Wide Web nicht nur erfunden, sondern der Menschheit als Allmende überlassen hat, hatte dann auch die Idee des Semantic Web. Seine Vision - und die vieler anderer - ist, weitere bedeutungstragende (semantische) Elemente für den Computer zugänglich und handhabbar, sozusagen "verstehbar" zu machen. Dazu gehören semantische Informationen im engeren Sinn wie z.B. *Synonymität*, aber auch sachliche Beziehungen, wie z.B. dass eine Person *Autor* eines Buchs ist oder *A Kunde* von B, oder z.B. *Kollege*. Für die Gesamtheit der Beziehungen eines Bereichs hat sich der Ausdruck "Ontologie" eingebürgert. In unserem Projekt wären z.B. die Attribute *Bedürfnis*, *Interaktion* und *politische Identität* Bestandteile unserer Ontologie (und als solche haben wir sie auch diskutiert). Als Grundstruktur für die Ontologie des Semantic Web wird das Resource Description Format (RDF) benutzt. Damit möchte man eine einfache Datenstruktur zur Verfügung stellen, die es erlaubt, die Vielfalt der erforderlichen Beziehungen auszudrücken. Ein Element dieses Formats ist quasi ein Satz mit Subjekt, Prädikat und Objekt. Also z. B. die Autorin von ISBN 978-3-86581-133-2 ist Elinor Ostrom. Mit der ISBN soll

angedeutet werden, dass es auch beim RDF darum geht, *eindeutige* Bezüge herzustellen, nämlich durch sogenannte URIs, für die die üblichen Webadressen (URLs) ein Beispiel sind. Interessant für uns ist ein formaler Aspekt: das RDF-Format besteht aus Tripeln (Subjekt, Prädikat, Objekt). Wer diesen Text aufmerksam gelesen hat, wird sich erinnern, dass auch OpenStreetMap solche benutzt, die wir als Objekt, Attribut, Wert interpretiert hatten. Nun lassen sich aber Objekt, Attribut, Wert - Tripel als ein Spezialfall von Subjekt, Prädikat, Objekt - Tripeln auffassen. Das eröffnet gute Perspektiven zwischen dem (relativ einfachen) Datenmodell von OSM und dem (sehr ambitionierten, aber in der Praxis auch nicht unproblematischen) Modell des Semantic Web zu vermitteln und sie bei Bedarf wechselseitig zu übersetzen. Die Ausdrucksmöglichkeiten des letzteren könnten uns dort helfen, wo wir mit denen von OpenStreetMap nicht auskommen.