

3 Die eigene Anwendung

3.1 Aufgabenstellung

3.1.1 Umfang

Die erste eigene Anwendung will ich nicht gleich durch zu hohe Ansprüche zum Scheitern verurteilen – das bisschen Ruby und Rails, das ich bislang kenne, macht mich noch längst nicht zum Überdrübersuper-Guru. Die Sicherheit langjähriger, exzessiver Programmierung kann schließlich ein einziges erfolgreich abgeleistetes Tutorial und ein interaktiver Debugger auf gar keinen Fall liefern.

Das bedeutet, dass ich die Oberfläche in dieser Anwendung noch nicht „anonymisieren“ möchte, dass ich also jedes Fenster, egal wie ähnlich es den anderen auch sein mag, separat betrachte – was im Klartext heißt, dass ich mich nicht um Mehrsprachigkeit kümmerge. Warum es das heißt? Weil ich Mehrsprachigkeit gerne dazu verwende, die üblichen Datenbank-Verwaltungen zu vereinheitlichen: Mithilfe des CoC-Prinzips ist nämlich eine einzige Fensterklasse für alle Dateien (ohne jegliche programmtechnische Anpassungen wie Vererbung oder virtuelle Methoden) schon fast in Reichweite, weil bei Mehrsprachigkeit jedes Textfeld sowieso angepackt werden muss. Doch wie gesagt – das muss warten.

Auch Protokollierungen wie die allseits beliebten Zeitstempel oder sonstige notwendige Dinge wie Benutzeranmeldungen oder andere Sicherheitsvorkehrungen sollen außen vor bleiben, obwohl Rails [erstere](#) berücksichtigt.

Was ich ausschließlich tun möchte, ist eine bestehende kleine Anwendung – ohne die obigen Bestandteile – nach Rails zu übertragen. Einfach eins zu eins, soweit es möglich ist. Das bedeutet schlicht, dass ich weder Analyse noch Design durchzuführen habe, weil ich bereits weiß, wie die Dateien aussehen sollen, was meine Fenster anzubieten haben und was ich sonst noch an Bearbeitung brauche.

3.1.2 Problemstellung: die 4ff-Methode

Die Anwendung selbst dreht sich um die [4ff-Methode](#), die in „[Die Individualität liegt in den Daten](#)“ beschrieben wird: Sie ist eine einfache kleine Art und Weise, Dateien in einem Datenbankverbund zu bewerten. Das geschieht aufgrund der [Definition der Information](#) als identifizierbarer, wiederholbarer Prozess (sprich nachvollziehbare Zustandsveränderungen, die die möglichen Datei-Feldwerte erzeugen) anhand von vier Angaben:

Das „Eigengewicht“ g_e eines Feldes ist bestimmt durch die Anzahl k möglicher Werte, die der Inhalt des Feldes annehmen kann:

Eigengewicht

$$g_e = 1 / (k-1)$$

Das „Profilgewicht“ g_p eines Feldes kennzeichnet seine „Wichtigkeit“: Wie häufig wird das, was das Feld repräsentiert, in den Dateien des betrachteten Dateisystems verwendet? Wie oft also kommt beispielsweise die Kundennummer vor? Sie taucht sicher in allen Angeboten, Aufträgen, Lieferscheinen und Rechnungen auf und wird wohl noch in den Kundendateien des CRM-Systems verwendet – diese „Ausbreitung“ der Kundennummer über das gesamte vorhandene Dateisystem wird bildhaft „Tentakelzahl“ T genannt. Mit der Anzahl aller Dateien p im System ist das Profilgewicht dann definiert zu:

Profilgewicht

$$g_p = T/p$$

Nebenbemerkung: Bei großen Systemen sollte sich diese Betrachtung nicht nur auf das Gesamtsystem beziehen, sondern auch für die Ebene der Module durchgeführt werden – bei kleinen Systemen kann dies jedoch unberücksichtigt bleiben.

Während diese beiden Werte (g_e , g_p) prinzipiell durch SQL-Befehle aus vorhandenen Datenbeständen ermittelt werden können, bedarf es bei den folgenden beiden „Gewichten“ fundierteren Wissens über das, was die gespeicherten Daten so alles auszusagen haben: Erforderlich wird die Kenntnis, woher der Inhalt stammt, der in diesem Feld hinterlegt ist und wozu er dient, wohin er also „geht“. Dazu gehören natürlich auch direkte Benutzereingaben als „Herkunft“ und ein prüfender Blick als „Ziel“, soll heißen, dass Daten, die auf

dem Bildschirm oder Papier zur Nutzung durch die Anwender gedacht sind, sehr wohl ebenfalls ein Ziel haben – die „direkte Ausgabe“. Bildlich gesprochen soll also sowohl der „Eintritt“ eines Inhaltes in das System als auch sein „Austritt“ erfasst werden, um zu bestimmen, wie lange der „Weg“ durch die eigene Maschinerie hindurch ist, ohne dass ein äußerer Einfluss darauf wirkt – oder mit anderen Worten, wie lange dieser Inhalt mit allen seinen eventuellen Veränderungen unter der ausschließlichen Kontrolle des eigenen Systems steht.

Sind mehrere Quellen oder Ziele vorhanden und damit mehrere Möglichkeiten oder Kombinationen gegeben, die Feldwerte zu ermitteln oder zu verwenden, so ist immer die kürzeste „Distanz“ maßgeblich. Diese ist dabei ganz analog der „[Länge auf der Information](#)“ als „Anzahl von Schritten“ definiert, die ein irgendwie gearteter Inhalt durchläuft, bis er in dem betrachteten Feld dann ankommt (oder ausgehend von diesem Feld, bis er irgendwo und irgendwie ausgegeben wird) – „Schritte“ sind hierbei alle Dateien, in denen der Inhalt oder „Teile“ davon „zwischengelagert“ werden. Sind direkte Benutzeraktionen und -verwendungen beteiligt, heißt das also, dass die maßgebliche kürzeste Distanz „Null“ ist.

Ein einfaches Beispiel sind Identifikatoren. Sie repräsentieren Objekte, die für die zu beschreibenden Geschäftsprozesse von Bedeutung sind und in der Regel in eigenen Stammdateien bearbeitet werden. In allen übrigen Dateien werden die Identifikatoren dieser Objekte nur noch zugeordnet, ihr Inhalt bleibt dabei völlig unverändert und dient nur dazu, die korrekte Auswahl treffen zu können – eine solche Zuordnung hat deshalb im einfachsten Fall die „Länge“ 1.

Als etwas diffizileres Beispiel sei ein Brutto-Positionswert in einem kleinen Verkaufsmodul genannt. Dieser Wert soll innerhalb einer Rechnungsdatei betrachtet werden, wobei zu beachten ist, dass in unserem Fall die Rechnungsdaten streng aus den Auftrags- und Lieferdaten bestimmt werden, also – auf dieser Ebene – nicht mehr willkürlich von Anwendern verändert werden können. Der Wert selbst berechnet sich einfach als Preis, multipliziert mit der Menge (Nettowert) zuzüglich Mehrwertsteuer. Das bedeutet, dass der Wert selbst drei „Quellen“ hat: Preis, Positionsmenge und Mehrwertsteuer. Preise und Mehrwertsteuer sind zwar in weiteren Dateien hinterlegt, doch da nur die „kürzeste“ Distanz zählt, ist demnach für die so genannte Portal-Distanz **P** nur

die Menge der Position maßgeblich. Diese kann nämlich in der Lieferung entsprechend den aktuellen Bedingungen zum Lieferzeitpunkt von den Anwendern noch geändert werden – was eine Portaldistanz von „Null“ für die Lieferdatei bedeutet. Für die Rechnung aber, die ihre Daten streng der Lieferung anpassen muss (weshalb die Portaldistanz also nicht Null sein kann), heißt dies, dass die Portaldistanz der Eingangsquelle „Menge“ gleich 1 ist. Dass die Mehrwertsteuer auch bei Lieferung auf ihre Gültigkeit geprüft werden muss oder ob der Preis bei Lieferung noch geändert werden darf oder nicht, ist deshalb nicht weiter von Belang: Kleiner als 1 kann die Portaldistanz in diesem Fall nicht mehr werden.

Die Exit-Distanz E jedoch schon – denn dieser Brutto-Positionswert mag zwar in weiteren Dateien noch verwendet werden, das aber zählt nicht: Denn in unserem Verkaufsmodul wird er auf der Rechnung ausgedruckt, sein „Weg“ endet also hier.

Exit-Distanz

Beide Distanzen P und E relativ zur Anzahl der Dateien p gesehen, stellen dann Portal- und Exitgewicht dar:

$$g_{pd} = P/p$$

$$g_{ed} = E/p$$

Portalgewicht,
Exitgewicht

Nebenbemerkung: Zu beachten ist, dass „Anwender“ hier als Schnittstellen in und aus dem System heraus zu sehen sind und deshalb nicht nur Menschen sein müssen. Auch Fremdsysteme, die Daten einfüttern und auslesen dürfen, sind „Anwender“ in diesem Sinne.

Doch zurück zu unseren Feldwerten: Interessant sind die vier Angaben deshalb, weil sie es erlauben, die Felder zu klassifizieren und zwar in zweierlei Hinsicht: „feldbezogen“ und „aufgabenbezogen“.

Der feldbezogene Typ wird über die Paarung (g_e, g_p) bestimmt, der aufgabenbezogene über (g_{pd}, g_{ed}) .

Feldbezogener Feldtyp

Damit lassen sich Felder hinsichtlich ihrer „Eigenart“ und ihres „Gebrauchs“ einordnen.

Klassifizierend	„Klassifizierende“ Felder sind solche, bei denen sowohl Eigengewicht g_e als auch Profildgewicht g_p hoch sind, soll heißen, dass sie zwar recht wenig verschiedene Werte aufweisen, aber sehr weit verbreitet sind. Typische Vertreter dieser
Akzentuierend	Gattung sind Mandanten oder Buchungskreise. „Akzentuierende“ Felder dagegen sind solche, bei denen zwar das Eigengewicht hoch ist, das Profildgewicht jedoch nicht. Länder, Sprachen, Geschlecht, Währungen sind bekannte Beispiele dafür – also in der Regel tabellengebundene Werte, die „Akzente“
Dokumentarisch	setzen. „Dokumentarische“ Felder sind die eigentlich interessanten in einer Datenbank, sie „dokumentieren“ die Geschäftsprozesse und zeichnen sich durch niedriges Eigengewicht und hohes Profildgewicht aus, denn sie sind in aller Regel Identifikatoren für bestimmte Objekte (wie Rechnungsnummern, Teile oder Kunden). Das wiederum heißt, dass sie zumindest in ihrer Stammdatei (Objektbeschreibung) eindeutig sein müssen und deshalb sehr viele verschiedene Werte aufweisen, während sie – als Repräsentant von Geschäftsvorfällen – in sehr vielen Dateien des Dateisystems auftauchen. Nach den interessantesten Feldern folgen nun diejenigen, die am wenigsten Aussagekraft
Deskriptiv	haben: Die „deskriptiven“ Felder sind Felder, deren Inhalte frei eingebbar sind und praktisch nicht geprüft werden. Sie zeichnen sich dadurch aus, dass sie sehr viele verschiedene Werte aufweisen (niedriges Eigengewicht), aber in kaum einer anderen Datei wieder verwendet werden (niedriges Profildgewicht). Ähnlich wie Felder nach ihrer Eigenart typisiert werden können, lassen sie sich auch über ihre Aufgabe einordnen.
Aufzeichnend	„Aufzeichnend“ werden Felder genannt, die nur eine kurze Portal- und Exitdistanz aufweisen, es sind also Felder, deren Inhalte flexibel an die Geschäftsprozesse angepasst werden müssen – soll heißen, dass sie von Anwendern nach aktuellen Gegebenheiten immer wieder geändert werden können – und die gleichzeitig von solcher Bedeutung sind, dass sie sofort verwendet werden. Typische Vertreter sind Auftragsdaten wie die Liefermenge, die zwar aus dem CRM-System stammen könnte, in Angeboten fixiert und über Aufträge vertragsrelevant zugesagt wird, aber letztlich bis zur tatsächlichen Lieferung an Kundenwunsch oder den Lagerbestand anpassbar bleiben muss.
Stempelnd	„Stempelnd“ sind dagegen Daten, die zwar eine geringe Portaldistanz aufweisen, dafür aber eine große Exitdistanz haben, das heißt, dass sie zwar das

„erste Mal“ in der betrachteten Datei auftauchen mögen, aber praktisch nie verwendet werden. Typische Vertreter sind die so genannten „Zeitstempel“, also Daten, die protokollieren, wann wer was getan hat, um im seltenen Fehlerfall, nicht aber im „normalen Tagesgeschehen“ dann Hilfe zu leisten. Werden solche Angaben jedoch genutzt, um beispielsweise die Arbeitszeit der Angestellten zu kontrollieren, verlieren sie diesen Charakter – die Exitdistanz schrumpft dann, weil im Tagesgeschäft dieser Feldinhalt irgendwo und irgendwie ganz planmäßig ausgegeben wird. „Auswertend“ sind weiterhin Felder, die eine hohe Portal- und Exitdistanz bei geringer Exitdistanz anbieten. Dies heißt, dass die „letzte Eingabe“ von Anwendern schon mehrere „Dateien“ entfernt lag, soll heißen, dass der betreffende Wert über mehrere Verarbeitungsstufen nur noch von der Software selbst geändert wurde – was wiederum bedeutet, dass einiges an eigener Leistung in diesem Feldinhalt steckt. Die niedrige Exitdistanz dagegen sagt aus, dass dieser Wert „kurz vor der Benutzung“ steht. Typische Vertreter sind, wie der Name bereits sagt, Auswertungen, die Massendaten komprimieren, um bestimmte Zusammenhänge deutlich zu machen und sie anschließend am Bildschirm oder auf Papier den Anwendern zur Verfügung zu stellen. Der letzte Fall des aufgabenbezogenen Feldtyps sind dann solche Felder mit hohem Portal- und Exitgewicht („umleitend“). Es sind also solche Felder, die viel Verarbeitung in sich stecken haben, aber im Alltagsgeschehen der Anwender nicht wirklich von Bedeutung sind, weil diese keinen besonderen Wert auf sie legen – sie wollen sie wohl nicht wirklich sehen. Das sind in aller Regel Felder, die das System für sich selbst nutzt, um seine eigenen inneren Abläufe zu organisieren. Dazu braucht es schließlich kaum Hilfe von den Anwendern, ob es nun Menschen oder Fremdsysteme sind – was ihre Eingaben entbehrlich macht und auch ihr fehlendes Interesse erklärt.

Auswertend

Umleitend

Diese Feldcharakterisierungen erlauben es nun, die gesamte Datei abzuschätzen auf ihre Aussagekraft.

Wird nämlich das Typgewicht einer Datei bestimmt als Summe aller „interessanten“ Feldgewichte (also ohne die wenig aussagekräftigen „deskriptiven“ und die rein organisatorisch interessanten „klassifizierenden“), so errechnet es sich zu:

Typgewicht

$$\Sigma T = \Sigma_{\text{logische,dok,akz}} g_e * g_p$$

Logische Felder

Logische Felder sind dabei solche, die tatsächlich der „Logik“ des Programmes dienen, also vorwiegend zur Aufzeichnung oder Auswertung von Datenmaterial dienen und weniger dazu gedacht sind, das Programm selbst am Laufen zu halten. Um diese dateiübergreifende Charakterisierung durchzuführen, werden Portal- und Exitdistanzen eines Feldes gemittelt und zwar über diejenigen Dateien, die auch die Tentakelzahl bestimmen, weil jede dieser Dateien auf dasselbe „Ursprungsfeld“ zugreift. Diese gemittelten Distanzen erfahren dann wiederum eine Normierung (in unserem einfachen Fall also auf die Anzahl aller Dateien) und ergeben so Gewichte, anhand derer die Bewertung durchgeführt wird. Doch nicht nur „logisch“ sollen die Felder sein, sondern auch „dokumentarisch“ und „akzentuierend“ – dann werden ihre Eigen- und Profildgewichte für das Typgewicht der Datei berücksichtigt.

Dokumentative
Deskriptive

Ist diese Summe im Vergleich der Dateien hoch, so wird die Datei „Dokumentative“ genannt, ansonsten „Deskriptive“. Dokumentative Dateien sind dabei solche, die es erlauben, ihren Datenbestand auch ohne genauere Kenntnis des einzelnen Datensatzes über typisierende Felder so genau zu bestimmen, dass der einzelne Satz über seine Merkmale eingekreist werden kann. Sie ermöglichen es beispielsweise, Kunden aufzufinden aufgrund ihrer Klassifizierung zusammen mit ihrer Landeszugehörigkeit und ihrem Kaufverhalten, ohne dass der genaue Name noch in Erinnerung sein muss – nicht unbedingt, weil es nur einen einzigen Fall geben könnte, der zu dem Muster passt, sondern schlicht, weil das Suchergebnis ausreichen reduziert wird, um noch übersichtlich zu sein oder anders ausgedrückt: Weil es manchmal genügt, suchenden Menschen eine überschaubare Auswahl von Möglichkeiten anzubieten, um ihrem Gedächtnis auf die Sprünge zu helfen.