ML-Methode – einfaches Analyseverfahren zur Erstellung von Programmvorgaben für Datenbank-Software

von Bevier, U. bevier@bussole.de

Abstract:

Basierend auf der axiomatischen Definition der Information (nicht Shannon) und der daraus resultierenden Existenz optimaler Lösungen stellt die ML-Methode einen Ansatz der, der die optimale Lösung durch eine hierarchische Dreieckskonstruktion mit gleichmäßiger Aufgabenverteilung zu bestimmen sucht. Sie ist ein einfaches und direktes Verfahren, um aus verbalen Beschreibungen eine grundlegende Objekthierarchie für die zu lösende Datenbank-Problemstellung zu erzeugen, die als Basis für eine UML-Modellierung dienen kann. Dies geschieht durch die Aufstellung einer Terminologie für das Problem zusammen mit einer Aufstellung binärer Beziehungen in der Terminologie und der Verwendung des Ansatzes der Dreieckskonstruktion. Der Ansatz erlaubt eine Längendefinition analog der Länge auf der Information, die eine Vermessung und Optimierung des Wirkungsgefüges ermöglichen.

1 Einleitung

Moderne Software wird immer anspruchsvoller, moderne Entwicklungsumgebungen immer mächtiger. Immer weniger technische Vorbildung wird heute von Entwicklern verlangt, um ein ansehnliches Resultat in vernünftiger Zeit zu erzielen, immer weiter dringt die Automatisierung auch in die Büros und ermöglicht Fachkräften, ihre Arbeit computer-unterstützt zu erledigen.

Insellösungen, die rasch und gezielt Einzelnen helfen, ihren ständig wachsenden Arbeitsanfall zu bewältigen, wuchern deshalb allerorten wie Pilze aus dem Boden. Viel zu oft werden aber diese kleinen, einfachen Lösungen dann entgegen ihrer ursprünglichen Absicht immer umfangreicher.

Um hier Fachfremden die Möglichkeit zu bieten, dennoch eine problemadäquate Struktur ihrer Insellösungen zu finden, die bei Bedarf auch erweiterbar ist, wurde die Messleine-Methode (ML) entwickelt. Als Voraussetzung verlangt sie nur, ein Problem verbal beschreiben zu können, als Resultat bietet sie den Grundriss der Dateistrukturen mit ihren öffentlichen Methoden an, die der optimalen Lösung unter den angegebenen Voraussetzungen bestmöglich entspricht.

2 Grundlagen und Begriffsbestimmung

Die Existenz der optimalen Lösung ist durch die axiomatische **Definition der Information**[1] als Gruppe der wiederholbaren, zusammenhängenden Wertveränderungen einer Eigenschaft gegeben, da Information somit Wirkung ist und als Wirkung dem Prinzip der geringsten Wirkung unterliegt. Eine

Länge ist auf dieser Menge als die kürzeste Wirkungskette zwischen zwei Werten der Eigenschaften definiert. (Fußnote 1: Herleitung der Definition der Information sowie Betrachtungen über Grenzen und Anwendbarkeit klassischer Mathematik siehe [1].)

Die Messleine-Methode[2] stellt nun den Ansatz dar, die optimale Lösung durch eine hierarchische Dreieckskonstruktion mit gleichmäßiger Aufgabenverteilung zu bestimmen. Die Vermessung erfolgt mithilfe eines Längenbegriffes, der sich eng an die Längendefinition der Information anlehnt. Die Optimierung zur Erreichung des geringsten Wirkungsverlaufs erfolgt unter der Annahme, dass kurze Wege ein Zeichen des Optimums sind. Diese Annahme erfolgt nicht nur aus der Überlegung der geringsten Wirkung heraus, sondern auch unter dem Gesichtspunkt der **Wiederholbarkeitsbedingung**[1] für Information, die an wohldefinierte Zustände geknüpft ist. Kurze Wege erleichtern die Überschaubarkeit der Zustände und gewährleisten über die Kontrollierbarkeit der Zustände somit auch den Erhalt der Information während der Verarbeitung.

Die zentralen Begriffe der Messleine-Methode sind Begriff, Impuls und Beziehung, Länge eines Begriffspaares, Gewicht und Fläche sowie Szenario und Systemgewicht. (Fußnote 2: Darstellung und Definitionen der ML-Methode sowie Strukturanforderungen und Messgrößen siehe [2].)

Begriffe sind dabei verbale Symbole für Problembestandteile mit der wesentlichen Eigenschaft, Wertveränderungen anstoßen und erfahren zu können oder unter aufgabenspezifischen Gesichtspunkten gesehen: sie sind die Elementaraufgaben, jedoch noch ohne Details hinsichtlich der jeweiligen Realisierung der Aufgabe. Ein Impuls ist die Übertragung von Wertveränderungen zwischen Begriffen, wobei der auslösende Begriff „Bedarf“, der aufnehmende Begriff „Verwendung“ heißt. Ein Impuls ist damit immer gerichtet, stellt aber eine (ungerichtete) Beziehung zwischen den beteiligten Begriffen her und ist somit charakteristischer Bestandteil der Kommunikation. Als Umgebung eines Begriffes werden seine gesamten Bedarfe und Verwendungen bezeichnet, seine Fläche ist definiert als das Produkt der Zahl der Bedarfe mit den Verwendungen.

Ein Szenario ist eine Ansammlung von Begriffen eines gemeinsamen Problembereichs, zusammen mit ihren Impulsen. Als Thread wird eine Impulskette bezeichnet. Ein Zusammenhang zwischen zwei Begriffen ist gegeben, wenn wenigstens eine Kette von Beziehungen zwischen beiden Begriffen vorliegt. Die Beziehungskette mit der kürzesten Anzahl beteiligter Begriffe bestimmt die Länge zwischen zwei Begriffen. Ein Impuls hat damit die Länge 1. Nicht zusammenhängende Begriffe haben keine Länge. Das Gewicht eines Begriffes ist die Summe der Längen zu allen übrigen Begriffen des Szenarios, das Systemgewicht ist die Summe der Gewichte aller Begriffe.

Als Objekt wird ein Handlungsträger des Szenarios bezeichnet, also ein Begriff zusammen mit seinen Impulsen, deren Ausgang er ist. Ein Objekt umfasst demnach sowohl Bedarfe als auch Impulse. Wichtig ist, dass im Objekt jede Paarbeziehung gültig ist, unabhängig von der Richtung. In unserem Fall von einfacher Datenbank-Software kann eine Datei als die Variablenstruktur eines Objektes

angesehen werden.

Jede Variable des Objekts, die kein Bedarf ist, ist jedoch nur eine interne Variable und die Methoden, die Wertveränderungen bewirken, sind interne Methoden und beeinflussen Systemgrößen nicht.

Ein System ist ein Szenario von zusammenhängenden Begriffen, ein Subsystem eine Untermenge diese Begriffe. Der zahlenmäßige Wert der optimalen Lösung ist für kleine Problembereiche (Anzahl Begriffe <= 43, Dreiecksstruktur, gleichmäßige Aufgabenverteilung) experimentell ermittelt worden [2]:

1. $S_+ = n*(n-1)*3$ **für** $w <= 6$
   $S_+ = n*(n-1)*(w/2)$ **für** $w > 6$

3 Vorgehen

Die ML-Methode ist der Ansatz, die optimale Lösung des betrachteten Problems durch eine hierarchische Dreieckskonstruktion mit gleichmäßiger Aufgabenverteilung zu bestimmen. Sie ist ein einfaches und direktes Verfahren, um aus verbalen Beschreibungen eine grundlegende Objekthierarchie für die zu lösende Datenbank-Problemstellung zu erzeugen. Dies geschieht durch die Aufstellung einer Terminologie für das Problem zusammen mit einer Aufstellung binärer Beziehungen in der Terminologie.

Mit der Längendefinition analog der Länge auf der Information kann dann das vorhandene Gerüst vermessen werden und auf Probleme untersucht werden. Diese Probleme, aus reinen Strukturmerkmalen ersichtlich, können wieder durch reine Strukturmerkmale behoben werden und so das Grundgerüst der Terminologie-Beziehungen solange „mechanisch“ ändern, bis die Messgrößen sich dem Optimum nähern.

Da das Systemgewicht als Optimum nur eine Strukturgröße ist, eine optimale Lösung sich aber immer am Inhalt ausrichten muss, ist das Systemgewicht leider nur eine notwendige Bedingung für die optimale Lösung, nicht aber eine hinreichende. Dies bedeutet, dass die sich ergebende Lösung inhaltlich immer noch überprüft, angepasst und neu durchgerechnet werden muss.

Erfahrungsgemäß können reale Lösungen durch dieses Verfahren bis auf ca. 2% an die optimale Lösung angenähert werden. Der Vorteil dieser Lösungen zeigt sich dann in der Realisierung, die aufgrund des höheren Verständnisses, das ein solches Vorgehen den Entwicklern bietet, in kürzerer Zeit und ohne Änderung der Grundkonstruktion erfolgen kann und somit die Gefahr konzeptioneller Probleme mindert.

Das gewünschte Ergebnis, auch Nicht-Informatikern die Möglichkeit zu geben, Insellösungen so zu gestalten, dass sie wartbar und erweiterbar sind, ist dadurch erreicht worden, dass die Methode nur

die verbalen Fähigkeiten der Entwickler benutzt und die technische Umsetzung der Algorithmen, die zutiefst mit den technischen Möglichkeiten ihrer Entwicklungsumgebungen verbunden sind, unberücksichtigt lässt.

Für professionelle Entwicklung ist die Methode dahingehend interessant, dass sie ein verbal fundiertes Grundgerüst für eine UML-Darstellung anbietet.

3.1 Begriffsfindung

Die Begriffsfindung kann aus den vorhandenen Dokumenten erfolgen. Günstig ist es, eine verbale Problembeschreibung dessen zu erstellen, was als Aufgabe programmiert werden soll. Dies hilft ganz allgemein, Unklarheiten, Widersprüche und Konflikte bereits im Vorfeld einzukreisen und sich eine schriftliche Unterlage zu beschaffen, die nicht nur als Verständigungsgrundlage, sondern auch als Dokumentation dienen kann. Diese Problembeschreibung ist kurz und präzise zu formulieren, gleiche Sachverhalte nicht blumig zu umschreiben, sondern immer mit den gleichen Worten zu bezeichnen.

Aus dieser Problembeschreibung sind die Hauptworte herauszufiltern, die sich direkt auf das Problem beziehen. Dies ergibt eine Liste von Begriffen. Diese Begriffe sind ebenfalls kurz und präzise zu definieren, wobei zur Erklärung möglichst Begriffe der Begriffsliste zu verwenden sind.

3.2 Impulsfindung

Impulse sind gerichtete, binäre Relationen zwischen zwei Begriffen. Der erste Ansatz kann aus den Begriffsbestimmungen selbst erfolgen. Werden Begriffe der Begriffsliste verwendet, so kann dies als Hinweis auf eine direkte Beziehung zwischen beiden Begriffen vermutet werden. Die Richtung des Impulses, dh. die Bestimmung, welcher Begriff Bedarf und welcher Verwendung ist, kann dann durch die folgenden Fragen ermittelt werden:

- ✓ ist “Begriff I” Bestandteil von “Begriff II”? “Hat” also “Begriff II” auf irgendeine Art “Begriff I”?
- ✓ braucht “Begriff II” Ergebnisse von “Begriff I”?
- ✓ muss “Begriff II” “Begriff I” überprüfen?
- ✓ folgt “Begriff II” aus einem anderen Grund auf “Begriff I”?

Begriffe ohne Bedarfe und Verwendungen sind sinnlos. Es muss deshalb zu jedem zweckdienlichen Begriff auch ein Bedarf und/oder eine Verwendung gefunden werden.

3.3 Strukturelle, automatisierbare Vermessung

Begriffe: Pro Begriff werden alle Bedarfe und Verwendungen gesammelt, das Produkt der Anzahlen bestimmt seine Fläche. Begriffe mit der Fläche Null zeigen an, dass sie entweder Input von außerhalb des Systems aufnehmen oder Output abgeben.

Die gesamte Anzahl der Begriffe bestimmt die Zoneneinteilung des Systemaufbaus [2]:

- ✓ 1 Objekt für n<=7,
- ✓ 2 Objekte für 7 < n < 13
- ✓ 1-Subsystem für n<= 43 endet
- ✓ 2 Subsysteme für 43 < n <= 85

Wir betrachten nur den Fall kleiner Problembereiche (1-Subsystem-Bereich, n <= 43).

Die Zoneneinteilung liefert uns die erwünschte Zahl anzulegender Objekte, auf die die Gesamtzahl aller Begriffe möglichst gleichmäßig aufzuteilen ist. Die Anzahl o der Objekte kann grob abgeschätzt werden durch:

$$o = (n-1)/(w-1) \quad (2.)$$

wobei n die Gesamtzahl der Begriffe ist und w eine natürliche Zahl, die der Quadratwurzel aus n am nächsten kommt.

Distanzenliste: Um die Distanzen der Begriffe zu finden, erstellen wir eine Distanzenliste, eine quadratische Matrix unserer Begriffe. Die Distanz der Begriffe zu sich selbst ist immer Null und bildet die Diagonale. Die restlichen Matrixelemente versehen wir mit dem Unterlassungswert n+1, um definierte Werte zu gewährleisten.

Als nächstes werden die durch unsere Impulsliste vorgegebenen Beziehungen in die Matrix eingetragen mit dem Wert 1. Dann werden alle Begriffe, deren Unterlassungswert noch n+1 ist, dahingehend überprüft, ob sie durch eine Beziehungskette mit der Länge 2 erreicht werden können. Dh. dass nun Begriffspaare untersucht werden, die durch keine direkte Beziehung (Impulse) verbunden sind. Weisen sie eine direkte Beziehung zu einem Begriff auf, der seinerseits eine direkte Beziehung zu einem anderen Begriff hat, so besteht zwischen dem Ausgangs- und dem Endbegriff die Länge 2. Schrittweise wird nun die Distanzenliste überarbeitet, bis die Länge aller Begriffspaare bestimmt ist.

Das Gewicht jeden Begriffes ist die Summe der Längen zu allen übrigen Begriffen des Szenarios. Es teilt uns die „Wichtigkeit“ des Begriffes in diesem Aufgabenbereich mit durch eine Messgröße, wie oft der Begriff gebraucht wird bzw. wie viele andere Begriffe er benötigt.

Systemgewicht: Aus den Gewichten summieren wir das Systemgewicht, das in diesem ersten Anlauf sicher weit vom gewünschten Optimum entfernt sein wird.

Erfahrungsgemäß ist der erste Wert wesentlich geringer als das Optimum, da im ersten Anlauf die Begriffe (Elementaraufgaben) meist noch recht unstrukturiert und deshalb zu dicht vernetzt sind.

Threadliste: Aus den Begriffen mit den Flächen Null lassen sich die durchgängigen Threads bestimmen, also die Impulsketten, die Input von außerhalb des System aufnehmen und als Output aus dem System abgeben.

Objektliste: Diese durchgängigen Threads liefern uns nun die Grundstruktur der Objekthierarchie. Die Begriffe mit Flächen Null, die keine Bedarfe aufweisen, sind per definitionem keine Objekte. Ausgehend von den verbliebenen Begriffen mit Fläche Null, die also keine Verwendung im eigenen System mehr haben, ordnen wir nun unsere Objekthierarchie gemäß der Anordnung der durchgängigen Threads an. Gleiche Threads oder Threads, die Teile anderer Threads sind, übergehen wir dabei.

Einzelobjekt: Aus der Thread- und Objektliste erstellen wir die Einzelobjekte als Begriffe mit Bedarfen, wobei die Bedarfe die externen Variablen darstellen.

3.4 Problemfälle bei Objekten

Aus dem strukturellen Ansatz der ML-Methode, eine hierarchische Dreieckskonstruktion mit gleichmäßiger Aufgabenstellung und kurzen Wegen zu erzeugen, ergeben sich eine Reihe von Anforderungen an ein Objekt, gegen die eine bestehende Lösung geprüft werden kann.

Dominante Begriffe: Dominante Begriffe sind Objekte, die mehr als eine Verwendung (Output) haben, da dies mehrdeutige Threads erzeugt und damit die gewünschte Übersichtlichkeit der Zustände gefährdet.

Klaffende Begriffe: Klaffende Begriffe sind „relativ überlastete“ Begriffe, die zu viele Bedarfe aufweisen.

Unentschiedene Begriffe: Begriffe, die nur einen einzigen Bedarf haben, weisen darauf hin, dass sie keinen Objektcharakter haben, sondern eher Bestandteil eines anderen Objektes sind.

3.5 Problemfälle des Systems

Auch für das System können eine Reihe von Anforderungen aufgestellt werden:

Kreiselnde Begriffe: Objekte, die Impulsketten formen, die wieder auf sich selbst zurücklaufen,

erhöhen Wirkung, jedoch ohne ein entsprechendes Ergebnis, und sind somit ein Anzeichen für eine Struktur, die nicht dem geringsten Wirkungsgefüge entspricht.

Konturlosigkeit: Ein konturloses System ist ein System, dem Input- und/oder Output-Begriffe fehlen. Es ist ein Spezialfall kreiselnder Begriffe, doch da sich in diesem Extremfall keine Objektliste erstellen lässt, ist dieser Fall nicht mehr durch Strukturbetrachtungen lösbar. Einzelne kreiselnde Begriffe dagegen können im Verlauf der Analyse immer wieder auftreten und verschwinden.

Verzettelung: Ein System, das insgesamt mehrere Output-Begriffe hat, kann einerseits aus dominanten Begriffen erzeugt werden, andererseits jedoch aus einer Zusammenhanglosigkeit einzelner Teilbereiche des Systems resultieren. Dies bedeutet, dass Gruppen von Wirkungsketten völlig ohne Kommunikation vorhanden sind und demnach entweder zwei verschiedene, eigentlich unabhängige Aufgabenbereiche vorliegen oder andererseits ein notwendiger Informationsaustausch nicht erfolgt. Im Falle unabhängiger Aufgabenbereiche ist eine Trennung auch in der Analyse zu empfehlen, da eine nicht erforderliche Komplexität unnütze Probleme bereitet, im Falle fehlender Kommunikation ist dieser Fehler zu beheben, was jedoch inhaltsbezogen erfolgen muss.

3.6 Strukturelle Lösungsansätze

Konturlosigkeit: Im Falle der Konturlosigkeit werden 3 Fälle unterschieden. Der erste Fall ist der minimaler Problembereich ($n < 13$). Aus den vorhandenen Begriffen ist keine Vorschlag ersichtlich, somit ist der einzige mögliche Vorschlag, neue Begriffe zu formulieren, die für Input bzw. Output zuständig sind. Der zweite Fall ist der Fall fehlender Output-Begriffe im System. Hier werden die leichtesten, inputfernsten Begriffe vorgeschlagen, da „Leichtigkeit“ bedeutet, dass der Begriff durchschnittlich kurze Impulsketten mit den vielen anderen Systemelementen aufweist, also „hoch vernetzt“ ist und inputfern deutet auf einen Begriff hin, der bereits aufarbeitete Ergebnisse weiterzuverarbeiten hat und damit bereits für das Endergebnis der Verarbeitung in Frage kommt. Fehlen dagegen Registratoren, wird die Suche nach „schweren“, outputfernen Begriffen durchgeführt.

Kreiselnde Begriffe: Das Aufbrechen der Schleifen in den Impulsketten sollte erfahrungsgemäß nicht durch einfaches Löschen von Impulsen erfolgen, da dies möglicherweise notwendige Kommunikation vernichten würde und zumeist zu Verzettelung führt. Auch die Umordnung von Impulsen führte nicht zu der gewünschten klaren Strukturierung der Objekte, sondern im Gegenteil zu einer Verstärkung der Schleifenbildung. Interessanterweise war die Umkehr von Impulsen hier am erfolgreichsten, die zudem das Systemgewicht nicht ändert.

Klaffende und dominante Begriffe: Hier werden zuerst Ähnlichkeiten gesucht in Bedarfen oder Verwendungen. Liegen solche Ähnlichkeiten vor, kann ein daraus resultierender Impuls gelöscht werden, da die betreffende erforderliche Wirkung an anderer Stelle bereits berücksichtigt wird. Liegen keine Ähnlichkeiten vor, müssen Begriffe umgeordnet werden. Hier ist eine Unterscheidung

nach dem Systemgewicht erforderlich. Liegt das Systemgewicht unter dem Optimum, deutet dies auf eine zu hohe „Dichte“ des Systems hin und damit auf überflüssige Kommunikationsvorgänge, es sind also weitere Objekte anzuraten: Objekte dürfen aufgespaltet werden. Liegt das Systemgewicht jedoch über dem Optimum, ist das System zu „locker“ angeordnet und weist auf fehlende Kommunikation hin, die Umordnung sollte deshalb keine weiteren Objekte erzeugen, sondern Objekte „anreichern“. Aufgespalten sollten die leichtesten Begriffe werden, da sie auf eine hohe Kommunikation mit dem übrigen System hinweisen, umgeordnet sollten die schwersten Begriffe werden, da hier geringe Kommunikation erwartet werden kann.

Unentschiedene Begriffe: Unentschiedene Begriffe sind zu Beginn der Verarbeitung uninteressant. Verbleiben sie im weiteren Verlauf der Analyse, sind diese „Fast-Objekte“ aufzuspalten.

Verzettelung: Die Verzettelung, die aus dominanten Begriffen stammt, wurde über die Beseitigung derselben eliminiert. Die Verzettelung, die aus getrennten Wirkungsgefügen erzeugt wurde, erlaubt aufgrund fehlender Überschneidung jedoch keine Vorschläge und muss somit den menschlichen Entwicklern überlassen werden, die über die strukturellen Bedingungen hinaus auch inhaltliche prüfen können.

3.7 Abschluss der strukturellen, automatisierbaren Verarbeitung

Der Abschluss der strukturellen Verarbeitung wird erreicht im Falle:

- ✓ der Konturlosigkeit von Minimalsystem
- ✓ der Verzettelung aus getrennten Wirkungsgefügen
- ✓ der Schleifenbildung von Vorschlägen, wenn also strukturelle Vorschläge nach einigen Durchläufen gleiche Zustände erzeugen
- ✓ der Unentscheidbarkeit von Zuständen, wenn also die Prüfungskriterien nicht eindeutig ein einziges Element selektieren können
- ✓ eine zu hohe Vernetzung des Systems vorliegt, sodass die Rechenkapazitäten überfordert sind
- ✓ keine Strukturmängel mehr vorliegen

Dieser letzte Fall ist derjenige, den das Verfahren als Lösung vorschlägt. Zu beachten ist, dass es nur eine strukturelle Lösung ist, sodass sowohl die Objekthierarchie als auch die Vorschläge des Aufbaus der einzelnen Objekte inhaltlich überprüft werden muss.

Solange hier Widersprüche vorliegen, müssen die Entwickler die Anordnung der elementaren Begriffe

ändern und die strukturelle Vermessung erneut auf die sich ergebende Konstruktion anwenden.

4 Literaturverweis

[1]: „Physik der Information“, ISBN 3-935031-03-03

[2]: „Die Fliege oder Das Handwerk der Datenbank-Programmierung“, ISBN 3-935031-02-05

5 Appendix

Das Buch „Die Fliege“ stellt ein kleines Tool zur Verfügung, mit dessen Hilfe die Rechnereien der strukturellen Vermessung sowie die automatisierbaren Lösungsvorschläge durchgeführt werden können. Das Tool wurde in der 4GL Omnis Studio ® Version 3.0 geschrieben, das Buch selbst führt die Methode an drei ausführlichen Beispielen vor.