Reverse Engineering deutschsprachiger Fachkonzepte

Harry M. Sneed
ANECON GmbH, Wien
Universität Regensburg

Abstrakt: Der folgende Beitrag beschreibt ein Werkzeug gestütztes Verfahren für die Analyse, Prüfung und Reverse Engineering deutschsprachiger Fachkonzepte. Der Auslöser für diese Arbeit war die Notwendigkeit fachlicher Testfälle aus Fachkonzepten abzuleiten. Hinzu kam die Notwendigkeit Entwicklungsprojekte an Hand von Konzepten zu kalkulieren. Schließlich, gab es die Anforderungen Konzeptdokumente formal auf Konsistenz und Vollständigkeit zu kontrollieren und deren Inhalt bezüglich Quantität, Qualität und Komplexität zu messen. Diese drei Anforderungen führten zur Entwicklung eines Textanalysators mit dem Ziel gewisse Information aus Prosatexten zu extrahieren. Vorausgegangen sind zahlreiche Studien zum Thema Reverse Engineering von Source Code und Sprachverarbeitung.

Keywords: *Natural Language Processing, Textanalyse, Key Words in Context, Fachkonzeptprüfung, Messung fachlicher Anforderungen, Reverse Engineering*

1 Problemstellung

Trotz aller Versuche, die Anwender zu bewegen, ihre Anforderungen zumindest semi-formal zu beschreiben, bleiben die meisten Fachkonzepte in der natürlichen Sprache des Anwenders, sei dies English, Deutsch, Italienisch, oder was immer. Möglicherweise werden die Prosatexte durch einzelne Diagramme wie Use-Case, Struktur- oder Datenflussdiagramme ergänzt, aber der Kern der Konzepte bleibt in Prosa. Auch wenn die Anwender Anwendungsfälle verwenden, sind die wichtigen Angaben – Schritte, Akteure, Ausnahmen und Vor- und Nachzustände mit Texten beschrieben.

Anderseits bildet das Fachkonzept, in dem die Benutzeranforderungen zusammengetragen sind, die Baseline oder Basis für alles was danach folgt. Die Systemspezifikation, die Aufwandsschätzung, der Systementwurf, die Programme und die Testfälle beziehen sich auf das Fachkonzept. Falls es unvollständig oder inkonsistent ist, leiden alle darauf folgenden Aktivitäten darunter. Die Schätzer, die Entwickler und die Tester können zunächst nur so viel wissen, wie sie aus dem Konzepttext entnehmen können. Natürlich, können sie nachfragen, was der Verfasser des Konzeptes mit der einen oder der anderen Aussage meint, aber ihre Fragen werden sich auf den Text beziehen. Das was im Text nicht vorkommt, wird nicht befragt. Außerdem kostet das Nachfragen sehr viel Zeit und hält das Projekt auf, so dass immer weniger nachgefragt wird.

1.1 Die Aufwandsschätzung

Eine der wichtigsten Rollen eines Fachkonzeptes ist als Bezugsdokument für die Aufwandsschätzung. Egal ob nach Function-Point, Object-Point, Use-Case oder Analogie geschätzt wird, muss der Schätzer die Größe und Komplexität des Vorhabens sowie auch die Qualitätsanforderungen aus dem Fachkonzept ableiten. Auch er kann nachfragen, aber sofern er kein Sachgebietsexperte ist, wird er nur Fragen bezogen auf den Text stellen. Demzufolge kann die Aufwandschätzung nur so wie das Fachkonzept sein.

1.2 Die Systemspezifikation

Die softwaretechnische Lösung wird aufgrund des Fachkonzeptes spezifiziert. Insofern als es sich um eine objektorientierte Lösung handelt, muss das Konzept in eine objektorientierte Struktur verkleidet sein. Das heißt, die Objekte müssen die Grundelemente bilden. Die Funktionen, Zustände und Regel müssen um die Objekte herum gruppiert sein. Sollte das Fachkonzept nicht ursprünglich diese Struktur haben, muss die Struktur transformiert werden. Jedenfalls, müssen die Vorgänge, bzw. Anwendungsfälle, die Objekte, die Attribute, die Operationen, die Zustände und die Geschäftsregel aus dem Fachkonzept hervorgehen.

1.3 Die Systemtestfälle

Jeder Test ist ein Test gegen etwas, d.h. der Abgleich zweier Gegenstände. Im Falle des Systemtests, wird das System gegen das Fachkonzept getestet. Dazu müssen die Systemtestfälle aus dem Fachkonzept abgeleitet werden. Jede Aktion, jeder Zustand und jede Bedingung von jedem Anwendungsfall sollte getestet werden. Die Spezifikation des Tests setzt jedoch voraus, dass diese im Text erkennbar sind. Denn das was dort nicht beschrieben ist, wird auch nicht getestet. Deshalb ist es so wichtig die Konzepte im Hinblick auf die Testbar-

keit zu formulieren. Sollte dies nicht der Fall sein, muss der Text dementsprechend aufbereitet werden.

2 Ein automatisierter Ansatz zur Textanalyse

Eine Möglichkeit das Problem der Konzeptanalyse zu lösen, ist die automatisierte Reverse Engineering der Fachkonzepte. Demnach werden die Prosatexte gescanned und mit bestimmten vorgegebenen Wörtern und Textmuster verglichen. Es komme darauf an, bestimmte Textentitäten, die für die Projektschätzung, die Systemspezifikation und die Testfallableitung von Bedeutung sind zu erkennen und zu extrahieren.

Die Verarbeitung der natürlichen Sprache ist so alt wie die Informationstechnologie. Daran wird seit Mitte der 50er Jahren in der einen oder der anderen Richtung gearbeitet. „Natural language processing" ist ein fester Bestandteil der Informationstechnologie und hat schon erstaunliche Erfolge erreicht, z.B. das Elisa Projekt des Professor Weizenbaumes in den 60er Jahren. Ob Spracherkennung oder Textanalyse, es ist schon alles da gewesen. Umso erstaunlicher ist es, dass sie nie angewendet wurde um Fachkonzepte zu analysieren. Das liegt wahrscheinlich daran, dass noch keiner es gewagt hat, die Zielergebnisse zu definieren. Da natürliche Sprachen so vielfältig sind, muss der Automat genau wissen, was es zu suchen hat. Die Sprache wird danach gefiltert und nur die relevanten Passagen extrahiert. Je präziser die Ziele und je feinmaschiger das Filter, desto ergiebiger der Ertrag aus der Sprachanalyse.

In der vorliegenden Anwendung der Sprachverarbeitung sind die Ziele relativ genau definiert. Es geht darum

- zu prüfen ob gewisse Textangaben vorhanden sind,
- zu zählen wie oft bestimmte Spracheigenschaften vorkommen,
- zu ermitteln welche relevante Hauptwörter als Objekte verwendet werden,
- zu ermitteln welche Aktionen auf diese Objekte stattfinden,
- zu ermitteln welche Zustände dieser Objekte abgefragt werden und
- zu ermitteln welche sonstige Bedingungen gestellt werden.

2.1 Identifizierung der Attribute

In jedem standardisierten Dokument gelten bestimme Angaben als Pflichtangaben. In der Regel sind sie in dem Dokumentenschablone bereits mit dem Titel eingebaut. Der Titel bzw. der Übertitel einer Tabellenspalte oder einer Tabellenzeile identifiziert den Attributtyp. Der Titel „*Datum:*" zeigt an, dass hier ein Datum einzutragen ist. Der Titel „*Vorbedingung:*" zeigt an, dass eine Vorbedingungsbeschreibung erfolgen sollte. Vor der Analyse des Textes müssen alle solchen Attributtitel in einer Tabelle mit der String Länge und dem String Inhalt als Textmuster festgelegt werden. Die Attribute werden als solche erkannt wenn ein String in der Textzeile mit dem Musterstring übereinstimmt. Wenn kein typgerechter Eintrag zu dem Titel gehört, wird es als ein fehlendes Attribut bezeichnet. Man gehe davon aus, dass alle Attribute auszufüllen sind, zumindest mit der Anmerkung „nicht zutreffend".

2.2 Identifizierung der Abschnitte und Anwendungsfälle

Dokumente sind in Kapitel und Abschnitte aufgeteilt. Bei der Analyse eines Dokumentes, ist es erforderlich diese Untergliederung zu erkennen, um die Inhalte dem Textabschnitt zuweisen zu können. Wichtig ist, dass jeder Abschnitt auch eine Bezeichnung hat, nämlich der Übertitel. Textgliederungen sind in der Regel nummeriert, so dass sie an deren Nummer erkennbar sind. Es kann aber vorkommen, dass sie nur mit einem Stichwort markiert sind, z.B.: *Anwendungsfall.* Für diesen Fall braucht der Textanalysator eine Liste aller solcher Stichwörter, die auf eine Untergliederung hinweisen.

2.3 Identifizierung der Objekte

Objekte sind per Definition alle Hauptwörter eines Textes. In der deutschen Sprache beginnen alle Hauptwörter mit einem Großbuchstaben. Das erleichtert die Erkennung. Dennoch gebe es jede Menge Hauptwörter die keine echten Objekte sind. Um die Suche einzuschränken muss der Toolbenutzer wie bei den Attributen, eine Tabelle aller relevanten Objektnamen für jedes Fachkonzept vor der Analyse erstellen. Das Analysewerkzeug vergleicht alle Hauptwörter mit dieser Tabelle um zu erkennen ob sie echte Objekte sind. Dabei kann ein Substring verglichen werden. Wenn der Substring mit dem entsprechenden Hauptwortteil übereinstimmt, wird das Objekt registriert. Da die Objekte das Skelett der Systemspezifikation bilden, ist es wichtig sie erkennen zu können.

2.4 Identifizierung der Aktionen

Aktionen sind Handlungen auf ein Objekt. Irgendetwas wird mit dem Objekt gemacht, z.B. es wird erstellt, verändert oder gelöscht. Zunächst wird das Objekt als solches erkannt. Dann wird geprüft ob das Objekt im Zusammenhang mit einem Aktionsverb steht. Z.B. in dem Satz „*das Konto wird geschlossen*" ist wird geschlossen die Aktion. In dem Satz „*der Kunde eröffnet ein Konto*" ist eröffnet die Aktion auf das Objekt Konto. Da in einem Fachkonzept fast ausschließlich die dritte Person verwendet wird, lassen sich die Verben an der Endung

„t“ oder „en“ erkennen. Wenn außerdem ein Objekt vor oder nach dem Verb steht ist die Aktion als solche erkannt.

2.5 Identifizierung der Zustandsabfragen

Objekte sind nicht nur der Gegenstand von Handlungen, sie können auch der Gegenstand einer Abfrage sein, z.B. in dem Satz „*Es ist zu prüfen, ob das Konto überzogen ist*“. Das Wort „ob“ lässt schließen, dass es sich hier um eine Abfrage handelt. Überzogen ist der Zustand, der abgefragt wird. Ein weiteres Beispiel ist die Bedingung „*Wenn der Frist abgelaufen ist...*“ Hier deutet das Wort „Wenn“ auf eine Abfrage des Zustandes abgelaufen. Zustandsabfragen sind also an gewissen Bedingungswörter wie „*wenn, sofern als, ob, solange als, gegebenenfalls, usw.* „ in Verbindung mit einem Objekt zu erkennen. Diese Bedingungswörter sind in einer internen Tabelle eingebaut.

2.6 Identifizierung sonstiger Bedingungen

Auch wenn sie keine Zustandsabfragen sind, können Bedingungen für die Testfallspezifikation von Interesse sein. In dem Satz „*Der Benutzer gibt solange Transaktionen ein bis er keine Lust mehr hat*“ ist kein explizites Objekt vorhanden. Dennoch sollte es für diese Bedingung „*bis er keine Lust mehr hat*“ einen Testfall geben. Deshalb werden solche Bedingungen auch registriert. Sie werden an der Satzstellung erkannt.

3 Die Ergebnisse der Textanalyse

Beim derzeitigen Stand des Analysewerkzeuges werden fünf Ergebnistypen erzeugt:

- ein Bericht über die formalen Mängel im Fachkonzept,
- ein Metrikbericht mit den wichtigsten Kennzahlen zur Größe, Komplexität und Qualität des Fachkonzeptes,
- ein Objektverzeichnis mit allen verwendeten Objekten samt deren Verwendungsarten,
- eine Tabelle der fachlichen Testfälle und
- eine Exportdatei für den Aufbau eines Spezifikations-Repositories.

3.1 Mängelbericht

Formale Mängel sind fehlende Pflichtangaben, fehlende Inhalte und Mängel in der Struktur des Fachkonzeptes. Beispielhaft für das Erste sind fehlende Attribute wie Vorbedingungen und Nachbedingungen zu einem Anwendungsfall oder fehlende Einträge in einer Datenbeschreibungstabelle. Beispielhaft für das Zweite sind Anwendungsfälle ohne Aktionen oder Zustandsabfragen. Beispielhaft für das Dritte sind zu lange Sätze, zu wenig Gliederungen und zu große Abschnitte. Alle Mängel werden hier mit dem Typ, dem Textverweis und der Seitennummer aufgelistet.

3.2 Metrikbericht

Der Metrikbericht umfasst die Quantitäts-, Komplexitäts- und Qualitätsmetriken des Fachkonzeptes. Die Quantitäten sind die wichtigsten Größenmaße. Dazu zählen u.a:

- die Anzahl Textzeilen,
- die Anzahl Anwendungsfälle,
- die Anzahl Sätze,
- die Anzahl Aktionen,
- die Anzahl Wörter,
- die Anzahl Zustände,
- die Anzahl Objekte,
- die Anzahl Bedingungen,
- die Anzahl Textabschnitte und
- die Anzahl Testfälle.

Mit Hilfe dieser Größen werden die Function-Points, Data-Points, Object-Points und UseCase Points für die Aufwandsschätzung der Entwicklung errechnet.

3.3 Objektverzeichnis

Das Objektverzeichnis bildet die Brücke zu einer objektorientierten Systemspezifikation. Es listet alle verwendeten Objekte in alphabetischer Reihenfolge auf und gibt an in welchen Anwendungsfällen, bzw. Vorgänge, sie wie verwendet werden. Sie können das Objekt einer Aktion, das Objekt einer Zustandsabfrage oder das Prädikat einer Bedingung sein. Diese Liste kommt einer objektorientierten Spezifikation nahe weil sie die Funktionen und Regel nach den Objekten ordnet.

3.4 Testfalltabelle

Die Testfalltabelle enthält alle Testfälle, die erforderlich sind um die erkannten Aktionen, Zustandsabfragen und Bedingungen zu testen, gruppiert nach Abschnitt, Vorgang und Anwendungsfall. Die Testfälle sind als relationale Tupel mit einem eindeutigen Identifizierungsmerkmal dargestellt, so dass sie ohne weiteres in eine Testfalldatenbank übernommen werden können.

3.5 Repository Exportdatei

Die Repository Exportdatei ist eine CSV – Comma Separated Value – Datei mit den erkannten Konzeptbeziehungen in fünf Spalten:

- der Textabschnittstyp,
- die Textabschnittsbezeichnung,
- die Beziehungsart,
- der Zielentitätstyp und
- die Zielentitätsbezeichnung.

Jeder Textabschnitt bzw. fachliche Funktion oder Anwendungsfall, die Objekte benutzt, Zustände abfragt oder Bedingungen stellt wird hier als Basis-

entität aufgelistet. Die Zielentitäten sind die benutzten Objekte, die ausgeführten Aktionen, die abgefragten Zustände, die gestellten Bedingungen und die zugewiesenen Attribute. Die Beziehungen werden hier als binäre Relationen zwischen einem Basisobjekt – der Textabschnitt oder der Anwendungsfall - und dem Zielobjekt – das Objekt, Attribut, Aktion, Zustand und Bedingung.

4 Zusammenfassung

Dieser Versuch deutschsprachiger Fachkonzepte automatisch zu analysieren hat zum Vorschein gebracht, dass es sehr wohl möglich ist, wertvolle Information aus dem Konzepttext zu gewinnen, vorausgesetzt die Textanalyse ist auf eine wohl definierte Untermenge der Sprache ausgerichtet und der Anwender die Objekte und Attribute vorher identifiziert. Es komme darauf an, die wesentlichen Schlüsselwörter und die Grammatik der Texte durch den Textscanner zu erkennen. Mit Hilfe weniger Sprachmuster ist es gelungen, die wesentlichen Sprachkonstrukte wie Objektveränderungen, Ereignisse, Zustandsabfragen und Bedingungen zu erkennen. Mit Hilfe des Benutzers ist es weiterhin gelungen Objekte und Anwendungsfälle zu erkennen. Die gewonnen Information hat jedenfalls ausgereicht, um die Qualität des Konzeptes zu beurteilen, den Aufwand zu schätzen und die fachlichen Testfälle zu extrahieren.

Natürlich kann dies nur als Anfang betrachtet werden. Es gebe noch viel zu tun, ins besonders was die Definition von Textmuster und Mustererkennung an betrifft. Das Netz zur Filterung der Semantik muss viel feinmaschiger werden, aber das Tool, das hier geschildert wird, hat schon brauchbare Ergebnisse in zwei Projekten zustande gebracht.