



Textuelle Analyse – eine Arbeitstechnik für die visuelle Modellierung

Version 2.0

Dieses Dokument wurde verfasst von ...

Dr. Jürgen Pitschke, BCS-Dr. Jürgen Pitschke, www.enterprise-design.eu

Diese Unterlagen können intern frei für nicht-kommerzielle Zwecke benutzt werden. Die kommerzielle Nutzung oder Weiterverbreitung jeglichen Teils dieser Unterlagen ist ohne Zustimmung von BCS - Dr. Jürgen Pitschke nicht gestattet. Für Lizenzen und Weiterverwendung sprechen Sie uns bitte an. Kopieren Sie diese Notiz in jede Reproduktion.

Inhaltsverzeichnis

| | | |
|-------|---|----|
| 1 | Benötigen wir „Textuelle Analyse“? | 3 |
| 2 | Ursprung der textuellen Analyse | 4 |
| 2.1 | Textuelle Analyse nach Abbott | 4 |
| 2.2 | Textuelle Analyse und UML-basierte Modelle | 4 |
| 2.3 | Textuelle Analyse und BPMN-basierte Modelle | 6 |
| 2.4 | Textuelle Analyse und andere Modellelemente | 6 |
| 3 | Die textuelle Analyse im Modellierungsprozess | 7 |
| 3.1 | Das Zachman-Framework | 7 |
| 3.2 | Vorgehensmodelle und der BCS-Modellierungsprozess | 8 |
| 3.2.1 | Identifikation von Konzepten | 9 |
| 3.2.2 | Aufbau eines Faktenmodells | 11 |
| 3.2.3 | Textuelle Analyse und RuleSpeak® | 12 |
| 3.2.4 | Textuelle Analyse und Prozessdekomposition | 13 |
| 4 | Wie textuelle Analyse wirklich funktioniert | 15 |
| 4.1 | Bewerten der identifizierten Kandidaten | 15 |
| 4.2 | Sprache und textuelle Analyse | 16 |
| 5 | Werkzeugunterstützung | 17 |
| 6 | Zusammenfassung | 18 |
| | Literatur | 19 |
| | Anhang A: Beispiel Service Desk | 20 |
| | Ungeordnete Informationssammlung (Auszug) | 20 |
| | Begriffssammlung (Makroebene, Auszug) | 22 |
| | Faktenmodell (Makroebene) | 22 |
| | Prozessmodell (Makroprozess Incident Management) | 23 |

1 Benötigen wir „Textuelle Analyse“?

„Textuelle Analyse“ ist eine Arbeitstechnik, die angewandt wird, um aus mehr oder weniger unstrukturierten Informationen *systematisch* Modellelemente zu extrahieren.

Anwender, die zum ersten Mal mit der textuellen Analyse in Berührung kommen, sind oft skeptisch. Benötigen wir das? Weshalb sollen wir diese Technik einsetzen? Meine Antwort ist sehr einfach: Jeder Modellierer wendet diese Technik an. Wenn wir Modelle entwickeln, liegen uns zuerst unstrukturierte, nicht formalisierte Informationen vor, meist in Form von Text. Diese Informationen müssen analysiert und in Modelle umgesetzt werden. Jeder Modellierer nimmt diesen Schritt also zumindest im Kopf vor. Meist wird dieser Schritt aber unbewusst vorgenommen und vor allem nicht dokumentiert. Daher ist er später nicht nachvollziehbar, die erstellten Modelle scheinen aus dem Nichts gekommen zu sein.

Wenn wir diesen Schritt auf jeden Fall durchführen, dann macht es Sinn ein **systematisches Vorgehen** zu wählen und das Ergebnis **explizit darzustellen**. Durch eine Systematik, die alle Projektbeteiligten einsetzen, wird dieser Schritt planbar und wiederholbar. Ergebnisse sind vergleichbar. Durch die explizite Darstellung wird dieser Schritt nachvollziehbar.

Sicher spielt auch die Werkzeugfrage eine Rolle. Ein Werkzeug, das die textuelle Analyse unterstützt, vereinfacht die Anwendung und die Dokumentation.

2 Ursprung der textuellen Analyse

Die textuelle Analyse geht zurück auf einen Artikel von Russel J. Abbott¹ in Communications of the ACM, November 1983 mit dem Titel „Program Design by Informal English Descriptions“. Aber auch davor gab es bereits Ansätze zur systematischen Analyse von textueller Information.

Verkürzt gesagt hatte Abbott die Idee aus natürlich-sprachigem Text Datentypen, Variablen, Operatoren und Steuerstrukturen abzuleiten, um Ada-Programme zu entwickeln. Es ist mir nicht bekannt, wie erfolgreich diese Versuche waren, aber 20 Jahre später, mit der stärkeren Verbreitung der visuellen Modellierung, fand die Idee neue Anwender und wurde weiterentwickelt. So referenziert z.B. Richard Dué 2002 in „Mentoring Object Technology Projects“² den Artikel von Abbott.

2.1 Textuelle Analyse nach Abbott

In seinem Artikel schreibt Abbott:

„We identify the data types, objects, operators, and control structures by looking at the English words and phrases in the informal strategy.

1. A common noun in the informal strategy suggests a data type.
2. A proper noun or direct reference suggests an object
3. A verb, attribute, predicate, or descriptive expression suggests an operator.
4. The control structures are implied in a straightforward way by the English.“

Abbott betont auch, dass der Formalisierungsprozess zumindest gegenwärtig nicht automatisch erfolgen kann. Es sind also Kreativität und Erfahrung gefragt.

Tabelle 1 fasst den Ansatz von Abbot im Überblick zusammen.

| Part of Text | Model Component | Example |
|-------------------|---|--|
| Proper Noun | Instance, Object | J. Smith, Euro |
| Common Noun | Class, Type, Role | toy, currency, seller |
| Doing Verb | Operation | buy |
| Being Verb | Classification | is an |
| Having Verb | Composition | has an |
| Stative Verb | Invariance-Condition | are owned |
| Modal Verb | Data Semantics, Pre Condition, Post Condition or Invariance Condition | must be |
| Adjective | Attribute Value or Class | unsuitable |
| Adjective Phrase | Association, Operation | The customer with children, the customer who bought the kite |
| Transitive Verb | Operation | enter |
| Intransitive Verb | Exception or Event | depend |

Tabelle 1: Textuelle Analyse nach Abbott

2.2 Textuelle Analyse und UML-basierte Modelle

Die Unified Modeling Language (UML) ist heute das Standardbeschreibungsmittel für objekt-orientierte Systeme. Die ursprüngliche Idee von Abbott lässt sich unmittelbar auf die

¹ Russel J. Abbott, Program Design by Informal English Descriptions, Communications of the ACM, Volume 26, Number 11, November 1983

² Richard T. Dué, Mentoring Object Technology Projects, Just Enough Series / Yourdon Press, Prentice Hall, 2002

Entwicklung von UML-Modellen, insbesondere Klassenmodellen, übertragen. Mit dem heutigen Wissen lässt sich der Ansatz aber verbessern und für weitere Diagrammartentypen erweitern.

Zuerst drehen wir die Frage von Abbott um, stellen den Ansatz für die Modellierung quasi vom Kopf auf die Füße. Abbotts Ansatz ging von den Satzelementen aus. Danach argumentierte er, welches Programmelement dem jeweiligen Konstrukt entspricht. Für die Modellierung wollen wir genau anders herum vorgehen: Wir überlegen, welche Modellelemente wir suchen (z.B. Klassen) und legen dann fest, wie wir diese Elemente in einem natürlich-sprachigem Text erkennen können. Wir tauschen die Reihenfolge der Spalten in der Tabelle. Das scheint eine unbedeutende Formalie zu sein, vereinfacht in meiner Erfahrung aber die praktische Anwendung. Tabelle 2 zeigt den erweiterten Ansatz für UML-basierte Modelle auszugsweise.

| Modellelement | Satzkonstrukt | Beispiel |
|--------------------------------|--|--|
| Use Case Diagram | | |
| Actor | Substantiv (aktives Objekt) | Der Service-Desk-Ingenieur, Der Endanwender ... |
| Use Case | Substantiv – Verb Verb – Substantiv | ... Störung melden meldet Störung ... |
| Association | Kontext | Der Endanwender meldet die Störung ... |
| Class Diagram / Object Diagram | | |
| Class | Unbestimmtes Substantiv | Währung |
| Object | Bestimmtes Substantiv | Euro |
| Association | Adjektiv, Adjektivphrase | ... der für die Störung verantwortliche Servicedesk- Ingenieur ... |
| Composition / Aggregation | Besitzanzeigendes Verb | ... hat ein ..., ... gehört zu ... |
| Generalization | Verb „sein“ | ... ist ein ... |

Tabelle 2: Textuelle Analyse für UML (Auszug)

Wir können diesen Ansatz mit anderen Techniken kombinieren. Für die Entwicklung von UML-Diagrammen haben sich z.B. „Zielfragen“³ bewährt. Diese helfen den richtigen Elementtyp zu identifizieren, wenn es mehrere Möglichkeiten gibt. Tabelle 3 fasst die Zielfragen für UML-Diagramme zusammen.

| Diagramm | Zielfrage |
|------------------|-------------------------|
| Use Case Diagram | WER macht WAS? |
| Sequence Diagram | WER machte WAS WANN? |
| Activity Diagram | WER liefert WAS an WEN? |
| Class Diagramm | (Wer macht was) WOMIT? |

Tabelle 3: Zielfragen für UML-Diagramme

Vergleichen wir die ersten drei Zeilen in Tabelle 2 für Elemente des Use-Case-Diagramms und die Zielfrage für das Use-Case-Diagramm sehen wir den engen Zusammenhang. Durch die Zielfrage „Wer macht Was?“ wird insbesondere der Zusammenhang zwischen Akteur und Use-Case und damit die Assoziation deutlich. Wir betrachten nicht mehr nur die Wortart, sondern auch den Satzbau und den Zusammenhang zwischen den einzelnen Satzgliedern. Die Zielfrage hilft auch bei der Bewertung, ob das gefundene Substantiv wirklich ein aktives Objekt ist und damit als Akteur in Frage kommt.

3 vgl. Hans J. Günther, Der 360 Grad Entwurf, Lulu Press 2008

2.3 Textuelle Analyse und BPMN-basierte Modelle

Für die Darstellung von Geschäftsprozessen hat sich die Business Process Modeling Notation (BPMN) als Darstellungsmittel etabliert. Wie für die UML lässt sich eine Tabelle aufstellen, die uns hilft, die benötigten Modellkonzepte in unstrukturierten Informationen zu identifizieren.

Tabelle 4 zeigt den Ansatz für BPMN-basierte Modelle.

| Modellelement | Satzkonstrukt | Beispiel |
|---------------|--|--|
| Pool / Lane | Substantiv (aktives Objekt) | ... die Entwicklungsabteilung..., ... der Service-Desk-Ingenieur ..., ... der Zollverantwortliche |
| Event | Zeitangabe, Konditionalsatz, Intransitives Verb | Um zwei Uhr ..., Bei Erhalt der Bestätigung, wenn die Bestätigung nicht innerhalb von 24 Stunden eintrifft ... |
| Activity | Substantiv – Verb | ... Ware kaufen Dokument archivieren Zollerklärung abgeben ... |
| Gateway | Konditionalsatz | Wenn für einen Incident eine feste Lösungsfolge bekannt ist, ... Wenn der Warenwert über 20 TEuro liegt, ... |
| Data Object | Substantiv (passives Objekt) | ... die Rechnung das Service-Level-Agreement die Ware ... |

Tabelle 4: Textuelle Analyse für BPMN-Modelle

Die Tabellen lassen sich noch weiter verfeinern. Z.B. können wir die verschiedenen Ereignistypen der BPMN unterscheiden. In der Praxis verursacht das zu viel Formalismus. Daher ist Vorsicht angebracht.

2.4 Textuelle Analyse und andere Modellelemente

Die OMG hat neben der UML und der BPMN eine Vielzahl weiterer Standardnotationen und Standards für die Unternehmensmodellierung hervorgebracht. Interessant sind besonders der Standard „Semantics of Business Vocabularies and Rules (SBVR)“ und die „System Modeling Language (SysML)“. Der Ansatz lässt sich auf weitere Notationen erweitern. Abschnitt 3.2.1 diskutiert z.B. die textuelle Analyse im Kontext des SBVR-Standards. Unterscheiden Sie dabei immer Inhalt und Darstellungsmittel.

3 Die textuelle Analyse im Modellierungsprozess

Häufig denken Anwender, dass die textuelle Analyse im gesamten Projekt nur am Anfang des Modellierungsprozesses angewandt wird. Das ist grundsätzlich falsch. Im Kapitel 2 haben wir gesehen, dass wir die textuelle Analyse anwenden können, um sehr unterschiedliche Modellelemente zu identifizieren. Es wäre unsinnig zu versuchen all diese Elemente in einem Durchgang zu identifizieren.

Wann und wie oft wir die textuelle Analyse anwenden, hängt von unserem Vorgehensmodell ab. Ich gehe hier nicht im Detail auf verschiedene Vorgehensmodelle ein. Unser Vorgehensmodell (BCS-Modellierungsprozess) orientiert sich am Zachman-Framework. Dadurch ist gewährleistet, dass alle relevanten Ergebnisse bedacht und erstellt werden. Unser Vorgehensmodell ist iterativ und nutzt Arbeitstechniken, Darstellungsmittel und Vorgehensweisen aus anderen bekannten Methoden, wie dem Unified Process.

Ein wichtiger Unterschied zu den im Kapitel 2 aufgeführten Tabellen für die textuelle Analyse liegt darin, dass wir im Rahmen unseres Projektes bestimmte **Inhalte** suchen, die durch bestimmte **Modellelemente** dargestellt werden. Wir suchen also nicht Klassen, Akteure, Use Cases oder ähnliches, sondern Begriffe, Fakten, Rollen, Geschäftsaktivitäten, Systemfunktionen usw. Leider geht diese Unterscheidung in vielen Modellierungsprojekten verloren. **Die Form ist nicht der Inhalt!** Wir können dieselbe Darstellungsform nutzen, um verschiedene Inhalte darzustellen.

3.1 Das Zachman-Framework

Das Zachman-Framework ist kein Vorgehensmodell, sondern ein Architekturmuster, dargestellt durch eine Matrix. Es definiert verschiedene Perspektiven (Zeilen) und verschiedene Sichten in jeder Perspektive (Spalten). Abbildung 1 zeigt das Framework.

| | What | How | Where | Who | When | Why | |
|----------------------------|---|---|---|--|---|--|--------------------------------------|
| Scope Contexts | Inventory Identification e.g. Inventory Types | Process Identification e.g. Process Types | Network Identification e.g. Network Types | Organization Identification e.g. Organization Types | Timing Identification e.g. Timing Types | Motivation Identification e.g. Motivation Types | Strategists as Theorists |
| Business Concepts | Inventory Definition e.g. Business Entity Business Relationship | Process Definition e.g. Business Transform Business Input | Network Definition e.g. Business Location Business Connection | Organization Definition e.g. Business Role Business Work | Timing Definition e.g. Business Cycle Business Moment | Motivation Definition e.g. Business End Business Means | Executive Leaders as Owners |
| System Logic | Inventory Representation e.g. System Entity System Relationship | Process Representation e.g. System Transform System Input | Network Representation e.g. System Location System Connection | Organization Representation e.g. System Role System Work | Timing Representation e.g. System Cycle System Moment | Motivation Representation e.g. System End System Means | Architects as Designers |
| Technology Physics | Inventory Specification e.g. Technology Entity Technology Relationship | Process Specification e.g. Technology Transform Technology Input | Network Specification e.g. Technology Location Technology Connection | Organization Specification e.g. Technology Role Technology Work | Timing Specification e.g. Technology Cycle Technology Moment | Motivation Specification e.g. Technology End Technology Means | Engineers as Builders |
| Component Assemblies | Inventory Configuration e.g. Component Entity Component Relationship | Process Configuration e.g. Component Transform Component Input | Network Configuration e.g. Component Location Component Connection | Organization Configuration e.g. Component Role Component Work | Timing Configuration e.g. Component Cycle Component Moment | Motivation Configuration e.g. Component End Component Means | Technicians as Implementers |
| Operation Instance Classes | Inventory Instantiation e.g. Operations Entity Operations Relationship | Process Instantiation e.g. Operations Transform Operations Input | Network Instantiation e.g. Operations Location Operations Connection | Organization Instantiation e.g. Operations Role Operations Work | Timing Instantiation e.g. Operations Cycle Operations Moment | Motivation Instantiation e.g. Operations End Operations Means | Workers as Participants |
| Released October 2008 | Inventory Sets | Process Transformations | Network Nodes | Organization Groups | Timing Periods | Motivation Reasons | Normative Projection on Version 2.01 |

Abbildung 1: ZachmanTM-Framework⁴, Quelle: www.zachmaninternational.com

Ein häufiges Missverständnis über das Zachman-Framework ist, dass die Zeilen von oben nach unten mehr Detail enthalten. Das ist falsch. Jede Zeile definiert eine neue Perspektive.

⁴ John Zachman, The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing, Zachman International, 2006, Electronic Book

Das Anwachsen von Detail geschieht innerhalb einer Zelle, das Framework ist eigentlich 3-dimensional. Abbildung 2 versucht das zu verdeutlichen.

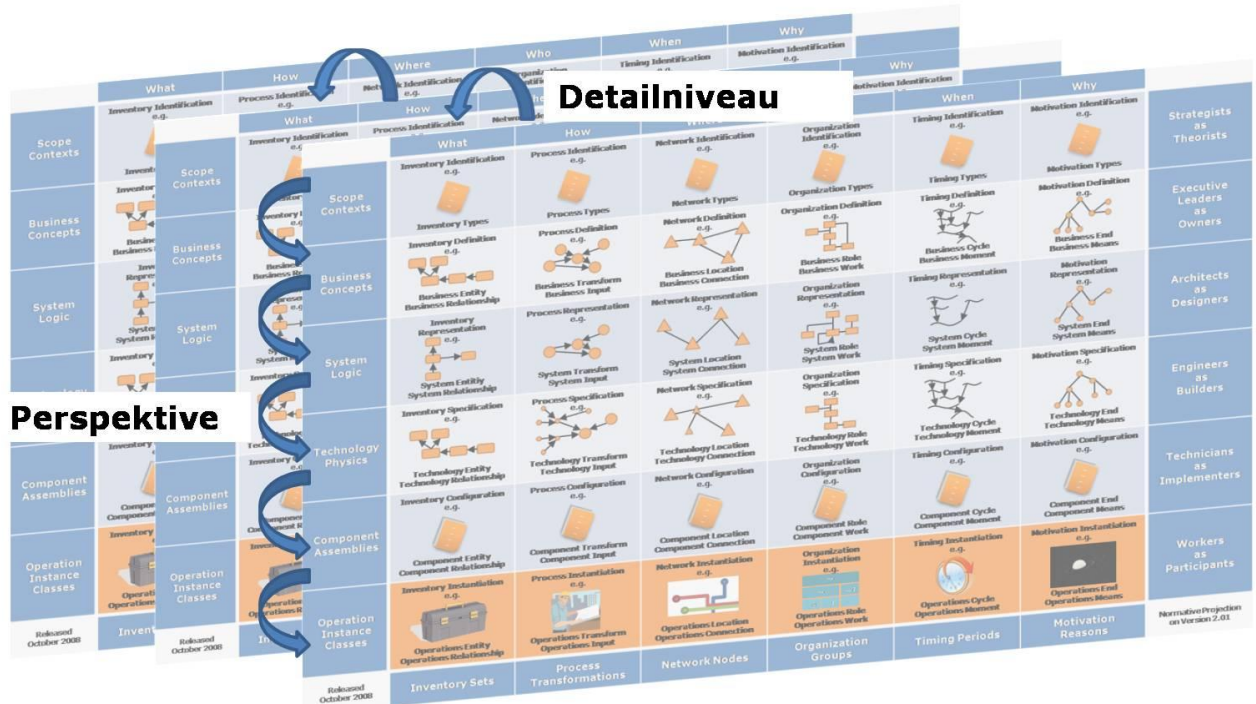


Abbildung 2: Zachman™-Framework – Perspektiven und Detailtiefe

Zwischen den Perspektiven bestehen Zusammenhänge. Die Perspektive „Business Model“ (Zeile 2) definiert Anforderungen und Voraussetzung für die Perspektive "System Logic" (Zeile 3).

Die textuelle Analyse ist prinzipiell für das Erstellen der Modelle für jede Zelle im Framework und für jede Detailstufe denkbar. In der Regel wird die textuelle Analyse jedoch für die Erstellung der Modelle in Zeile 1 und 2, beim Übergang von Zeile 1 zu Zeile 2 und von Zeile 2 zu Zeile 3 angewandt. Für die anderen Perspektiven werden andere Analyse-Methoden angewandt, z.B. beim Übergang vom Informationsmodell zum Datenbankmodell (Stichwort Normalisierung; Spalte 1, Zeile 3 zu Zeile 4).

3.2 Vorgehensmodelle und der BCS-Modellierungsprozess

Egal welchem Vorgehensmodell Sie folgen, werden Sie die Zellen des Zachman-Frameworks Schritt für Schritt füllen. Es spielt keine Rolle, ob Sie einem Wasserfallmodell folgen oder den Unified Prozess anwenden, ob Sie klassisch arbeiten oder ein „agiles“ Team leiten. Sie müssen entscheiden, welche Zellen oder besser welche Ergebnisse für Ihr Projekt relevant sind und diese systematisch erstellen. Beginnen werden Sie wahrscheinlich in Zelle 1,6 – „Scope, Why“. Dabei werden das zu lösende Problem und die Motivation für unser Projekt benannt und untersetzt. Wir formulieren den Projektauftrag. Die gesamte Zeile 1 – Scope – dient der Einordnung und Abgrenzung unseres Projektes. Wir benennen Prozesse, Funktionen, Akteure, Rollen und andere Konzepte, die in unserem Anwendungsbereich relevant sind. Diese bewerten wir danach, ob sie Gegenstand unseres Projektes sind oder nicht. Wir beschreiben hier auch Bereiche, die in unserem Projekt nicht analysiert, optimiert oder implementiert werden. Z.B. stellen wir alle relevanten Geschäftsprozesse grob dar, um eine Abgrenzung der zu betrachtenden Prozesse vorzunehmen. Nur wenn wir diese benennen, können wir sagen, dass sie nicht Gegenstand unseres Projektes sind (Negativabgrenzung).

Die Disziplin „Business Modeling“ schließt sich an. Ziel dieser Disziplin ist die Ausarbeitung der Zeile 2. Im Deutschen benennen wir diese Disziplin oft „Fachmodellierung“. Mir persönlich gefällt der Begriff „Geschäftsmodellierung“ besser. Wir stellen den Gegenstandsbereich aus „geschäftlicher“ Sicht dar. Sagen wir „Fachmodellierung“ ist das Ergebnis meist ein Teilmodell, das nur einen isolierten Bereich darstellt und dann in einer Silolösung endet. Oft verschwimmt die Abgrenzung zu Design und Implementation. „Geschäftsmodellierung“ meint die Modellierung der gesamten Perspektive „Business Model“, nicht nur der Geschäftsprozesse und Geschäftsaktivitäten.

Unser Vorgehensmodell, der BCS-Modellierungsprozess, ist ein prozessgetriebenes Vorgehensmodell. D.h. zentrales Konzept der Modellierung und Analyse ist der Geschäftsprozess. Identifizierte Geschäftsaktivitäten werden durch Systemfunktionen unterstützt oder automatisiert.

Der erste Schritt ist jedoch nicht die Identifizierung von Prozessen und darin enthaltenen Geschäftsaktivitäten, sondern die Definition einer ersten Version des benutzten Vokabulars⁵ (Business Vocabulary).

Oft können wir eine erste Version des Vokabulars „erben“. Wenn Sie sich z.B. mit Logistik beschäftigen finden Sie im Internet, in Wörter- oder Lehrbüchern Begriffe, die alle Anwender in diesem Bereich gebrauchen. Jeder Logistiker weiß, was ein Container, ein LCL-Container oder ein FCL-Container ist. Wenn Sie sich mit der Organisation eines Service-Desk befassen, können Sie den ITIL-Standard zu Rate ziehen, um grundlegende Begriffe für Ihr Projekt zu erhalten. Darstellen können Sie das Vokabular in Form eines Glossars oder - besser - Sie folgen dem SBVR-Standard und definieren Begriffe und Fakten (Terms and Facts).

Eine gute Praxis ist es, zeitig mit der Erstellung des Vokabulars zu beginnen, da Sie damit viel unnütze Diskussion und Arbeit bei der Modellierung und Analyse von Prozessen, Regeln, Funktionen vermeiden. Wir beginnen dabei mit einem kleinen, initialen Vokabular. Es ist nicht Ziel alle Begriffe und Fakten zu definieren, bevor wir mit der Modellierung von Prozessen oder anderen Artefakten beginnen.

Der BCS-Modellierungsprozess folgt wie andere moderne Ansätze einem iterativen Vorgehen. Unser Vokabular, unser Prozessmodell, usw. entstehen also nicht als „Big Bang“, sondern in (meist) drei Stufen. Zuerst betrachten wir die Struktur der Prozesse und die Definition der wichtigsten Konzepte. Danach verfeinern wir, z.B. in die Management-Sicht. Diese wiederum in die Aufgabensicht.

3.2.1 Identifikation von Konzepten

Im Anhang A finden Sie den Auszug eines Textbeispiels, wie wir es in unseren Kursen benutzen. Es beinhaltet Informationen über den Anwendungsbereich „Service Desk“ in völlig unstrukturierter Form. Es enthält Informationen über Ziele, über den eigentlichen Geschäftsprozess, über Systemfunktionen, über Implementationsdetails, über Stakeholder und Randbedingungen.

Wir können die oben benannten Techniken anwenden und uns interessierende Konzepte identifizieren – Klassen, Objekte, Akteure, Funktionen, Ziele, Anforderungen, usw. Wir erhalten damit eine erste Sammlung von Modellelementen, die wir nach der Einordnung in das Zachman-Framework und nach Ihrem Detaillierungsgrad bewerten müssen. Das ist eine typische, aber nicht optimale Vorgehensweise bei der Modellierung von Prozessen und Systemen.

Das Beispiel zeigt keine gute Informationssammlung. Diese Informationssammlung ist offensichtlich nach dem Muster „Schreib bitte alles auf, was du zu diesem Thema weißt.“

⁵ Teilnehmer unserer Kurse kennen die Bibel-Anekdote und wissen, warum das Vokabular zuerst entwickelt wird.

entstanden. Das ist keine gute Idee. Wenn wir uns auf die Suche nach Informationen begeben, müssen wir immer gezielt Fragen. Wir stellen also wieder Zielfragen. Diesmal zielen diese jedoch auf Inhalte und nicht auf Darstellungsformen.

Für einen ersten Schritt wären folgende Fragen zielführend:

- Benenne bitte die Begriffe, die im Anwendungsbereich „Service Desk“ relevant sind.
- Gib eine kurze Definition dieser Begriffe an.

Wenn wir bereits ein Vokabular z.B. aus dem ITIL-Standard übernommen haben, ist folgende Frage eine notwendige Ergänzung:

- Ordne die Begriffe den folgenden gegebenen Definitionen zu.

Eine sehr wichtige Zusatzfrage ist:

- Benenne dir bekannte alternative Bezeichnungen für die erfassten Konzepte.

Haben Sie die Fragen aufmerksam gelesen? Ist Ihnen aufgefallen, dass die erste Frage nicht lautet:

- Benenne bitte **ALLE** Begriffe, die im Anwendungsbereich „Service Desk“ relevant sind. Die zweite Frage zielt auf eine **kurze** Definition ab.

Bedenken Sie hier unser iteratives Vorgehen. Im ersten Schritt reicht uns eine Definition der wichtigsten Konzepte des Anwendungsbereiches. Für das Beispiel „Service Desk“ sind das vielleicht die Begriffe „Service“, „Incident“, „Problem“, „Change Request“, „Configuration Item“, „User“, „Service Desk Ingenieur“. Sicher werden Ihnen die Anwender weitere Begriffe liefern. Beschränken Sie aber im ersten Schritt die Anzahl der gesuchten Begriffe. Ich habe Projekte erlebt, die als Ergebnis der ersten Begriffssuche ca. 30 Seiten Text geliefert haben. Diese Projekte verzetteln sich endlos.

Eine gute Modifikation unsere Frage wäre daher:

- Benenne bitte **die 10 wichtigsten Begriffe**, die im Zusammenhang mit dem Thema „Service Desk“ relevant sind.

Statt 10 können Sie auch andere Größen für Ihr Projekt wählen. 10 ist aus unserer Erfahrung eine praktikable Zahl. Der Anwender ist gezwungen, zu strukturieren. Oft sagen uns Anwender, dass sie viel mehr Konzepte haben. Das ist natürlich richtig. Wir wollen aber die 10 wichtigsten Konzepte wissen. Wenn ein Anwender uns sagt, dass er mehr als 10 Konzepte hat, die wirklich wichtig sind, dann wurde der erste Schritte „Abgrenzung“ („Define Scope“) nicht erfolgreich abgeschlossen. Ich bevorzuge genau aus diesem Grund in meinen Projekten eher eine niedrigere Zahl. Erhalten Sie wesentlich mehr Konzepte und sind diese Konzepte wirklich alle derselben logischen Ebene zuzuordnen, dann versuchen Sie die Anwendungsbereiche in der nächsten Iteration besser zu strukturieren und abzugrenzen. Die gefundenen Begriffe lassen sich dazu gut nutzen.

Die so erhaltenen Begriffslisten können wir nun analysieren. Das ist nicht sonderlich schwer. Wir folgen dem ursprünglichen Ansatz von Abbott und unterscheiden Eigennamen (Proper Noun), die individuelle Konzepte bezeichnen (z.B. Euro, Pound Sterling), und Substantive, die generelle Konzepte (Common Noun) bezeichnen (z.B. Währung).

Ein großes Problem ist die Verwendung von Synonymen. Für ein und dasselbe Konzept werden in der täglichen Praxis verschiedene Bezeichnungen verwendet. Wir müssen hier abwägen. Einerseits wollen wir nur wenige Synonyme führen, um den Aufwand zu verringern. Andererseits müssen wir die Sprache des Zielpublikums berücksichtigen. SBVR bietet dafür das Konzept der „Communities“

Die identifizierten Konzepte können in Form eines Glossars oder als Einträge in einem Repository dargestellt werden. Das ist abhängig von den verwendeten Werkzeugen.

3.2.2 Aufbau eines Faktenmodells

Bestandteil eines Vokabulars nach SBVR ist ein Faktenmodell, das auf den gefunden Begriffen (Terms) aufbaut und Zusammenhänge zwischen diesen darstellt. Besser gesagt: Das Faktenmodell etabliert das Vokabular.

Im Faktenmodell werden Fakttypen (Fact Types; z.B. Land benutzt Währung) und Fakten (Deutschland benutzt Euro) dargestellt.

D.h. der nächste logische Schritt für die textuelle Analyse ist die Identifikation der Fakttypen und Fakten. Dieser Schritt kann durchaus gemeinsam mit dem Finden der Begriffe (Abschnitt 3.2.1) erfolgen. Es empfiehlt sich diese Schritte zu trennen, wenn die Begriffsbasis noch sehr unklar ist. In späteren Iterationen werden die Schritte typischerweise zusammen ausgeführt.

Wir ergänzen unseren Fragenkatalog um die Fragen:

- Benenne bitte **Beziehungen** zwischen den Konzepten.
- Beschreibe diese Beziehungen kurz.

Ein Faktenmodell kann z.B. durch ein UML-Klassendiagramm dargestellt werden (siehe SBVR, Annex H). Ein Faktenmodell ist aber kein Informationsmodell oder Datenmodell. Es enthält keine Kardinalitäten, keine Datentypen oder andere implementations- oder designspezifischen Informationen.

Ron Ross charakterisiert Fakten in „Business Rule Concepts“ als: „Fact types recognize what is possible to know, but given that, no other constraints.“⁶

Wir können Fakten und Fakttypen als Ergebnisse von Geschäftsaktivitäten sehen. Z.B. kennen wir einen Fakttyp „Service-Desk-Ingenieur ist verantwortlich für Incident“. Damit wissen wir, dass es eine oder mehrere Geschäftsaktivitäten geben muss, die dieses Ergebnis hervorbringen. Das Faktenmodell hilft uns später Geschäftsregeln, Geschäftsprozesse, Geschäftsaktivitäten und Systemfunktionen zu identifizieren und zu strukturieren. Daher ist das Erstellen eines Begriffs- und Faktenmodells für jede Art der visuellen Modellierung ein essentieller erster Schritt.

Überlegen wir, was notwendig ist, um das Ergebnis „Service-Desk-Ingenieur ist verantwortlich für Incident“ zu produzieren, dann finden wir z.B. folgende Geschäftsaktivitäten:

- Klassifizieren des Incidents (Klassifizieren bedeutet zuordnen des Incidents zu einer Hauptkategorie)
- Suchen eines verfügbaren Service-Desk-Ingenieurs, der Wissen für die Incident-Kategorie besitzt
- Zuweisen des gefunden Service-Desk-Ingenieurs als verantwortlichen Bearbeiter für den Incident

Hinterfragen und analysieren wir diese Geschäftsaktivitäten, finden wir neue Informationen für das Faktenmodell:

- Incident ist einer Kategorie zugeordnet
- Service-Desk-Ingenieur hat die Eigenschaft „verfügbar“
- Service-Desk-Ingenieur hat Wissen über eine Incidentkategorie
- Ein Service-Ingenieur, der für einen Incident verantwortlich ist, wird „verantwortlicher Bearbeiter“ genannt

Selbstverständlich haben wir hier – implizit oder explizit – wieder die textuelle Analyse angewandt. Bedenken Sie, dass unser Prozess iterativ ist. Dieser Schritt gehört zur zweiten Iteration.

⁶ Ronald G. Ross: Business Rule Concepts, Second Edition, Business Rules Solutions, LLC 2005, Seite 16

Das Faktenmodell kann textuell oder grafisch dargestellt werden. Ich bevorzuge eine grafische Darstellung. Der SBVR-Standard zeigt in den Anhängen G und H dafür zwei Varianten auf. Anhang G zeigt die „Concept Diagram Graphic Notation“, Anhang H zeigt die Darstellung des Faktenmodells mit UML-Klassendiagrammen. Welche Variante man wählt ist meist eine Werkzeugfrage. Wird die „Concept Diagram Graphic Notation“ unterstützt, ist diese für das Faktenmodell besser geeignet und vermeidet, dass bereits Implementationsdetails in das Faktenmodell eingestreut werden.

Die textuelle Darstellung wird genutzt, wenn im Projekt kein Werkzeug für die visuelle Modellierung eingesetzt wird.

Die textuelle Darstellung ist aber de facto auch die Vorlage für die textuelle Analyse. Wir drehen für unsere Zwecke das Ganze wieder um. Der SBVR-Standard sagt, wir stellen einen binären Fakttypen in der Form <<Begriff>> <<Prädikat>> <<Begriff>> dar. Für die textuelle Analyse sagen wir, wenn wir ein Konstrukt der Form <<Begriff>> <<Prädikat>> <<Begriff>> in unseren textuellen Informationen finden, nehmen wir an, zwei Begriffe und einen binären Fakttypen identifiziert zu haben. Tabelle 4 gibt eine Übersicht für die verschiedenen Formen von Fakttypen.

| Fakttyp | Konstrukt | Beispiel |
|---------------------|---|---|
| Unärer Fakttyp | Subjekt Prädikat | Service-Desk-Ingenieur ist verfügbar. |
| Binärer Fakttyp | Subjekt Prädikat Objekt | Anwender berichtet Incident. |
| Ternärer Fakttyp | Subjekt Prädikat Objekt Präpositionales Attribut | Anwender berichtet Incident für Asset. |
| Quaternärer Fakttyp | Subjekt Prädikat Objekt Attribut Adverbiale Bestimmung | Anwender berichtet Incident für Asset am Datum. |

Tabelle 5: Fakttypen und Beispielkonstrukte⁷

In der Praxis werden vorrangig unäre und binäre Fakttypen dargestellt. Gelegentlich sehen wir 3-stellige Fakttypen. In praktischen Projekten ist mir noch kein 4-stelliger Fakttyp begegnet. Meist werden solche Relationen im Zuge der Vereinfachung und Normalisierung in binäre Relationen aufgelöst.

Die dargestellten Techniken funktionieren mit Englisch meist besser als mit einem „normalen“ deutschen Text. Das liegt daran, dass der Satzbau im Deutschen mehr Varianten aufweisen kann. Es empfiehlt sich daher bereits bei der Informationserfassung, wenn möglich, einen einfachen Satzbau zu wählen⁸. Zugleich soll der Text durch einen Leser als normaler, verständlicher Text wahrgenommen werden.

Die textuelle Analyse ist eine kreative Tätigkeit, keine pure Mechanik.

3.2.3 Textuelle Analyse und RuleSpeak[®]

RuleSpeak[®] ist die Methode zur Darstellung von Geschäftsregeln in natürlicher Sprache⁹. Die Satzmuster zur Darstellung von Regeln und Empfehlungen sichern einerseits die Verständlichkeit der Statements für den Geschäftsanwender. Zugleich wird durch die Nutzung eines formalisierten Satzbaus und eines reglementierten Vokabulars eine Prüfung der Qualität der Regeln möglich.

⁷ siehe Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, OMG Document Number: formal/2008-01-02, Anhang I

⁸ vgl. Pitschke, RuleSpeak[®] - Kommentare zu den Basisdokumenten, Version 1.1, 2009, <http://www.enterprise-design.eu/de/knowhow/ressources>

⁹ Details zu RuleSpeak[®] in Deutsch sind auf www.enterprise-design.eu oder auf www.rulespeak.com zu finden.

Bei der textuellen Analyse zur Findung von Geschäftsregeln werden folgende Schritte ausgeführt:

- Aussagen identifizieren, die Regeln oder Empfehlungen darstellen
- Umformen (oder besser „Normalisieren“) dieser Aussagen, so dass sie den RuleSpeak-Vorschriften entsprechen (Verwendung der Schlüsselworte, Dekomposition der Regeln)

Das Dokument „RuleSpeak® Guidelines - Grundlagen, Geschäftsregeln in natürlich sprachlichem Deutsch darstellen – was man tun sollte und was nicht“ ist eine Einführung in diese Thematik und stellt eine vereinfachte Anleitung zur textuellen Analyse für Geschäftsregeln dar.

3.2.4 Textuelle Analyse und Prozessdekomposition

Der BCS-Modellierungsprozess ist ein prozessgetriebener Ansatz. Der Geschäftsprozess ist zentrales Element des Modellierungsansatzes. Der Prozess ist das „motivierende“ Element. Andere Modelle und Modellelemente werden aus der Analyse der Geschäftsaktivitäten abgeleitet.

Die Darstellung der Prozesse erfolgt in drei Ebenen:

1. Struktur-Sicht: Grobe Prozessstruktur
2. Management-Sicht: Verfeinerung der Subprozesse der Struktursicht, abstrakte Darstellung des Prozesses
3. Aufgabenebene: Verfeinerung der Subprozesse aus der Managementsicht, Detailbeschreibung des Prozesses

Diese Darstellungsform kann durch Dekomposition aber auch durch Komposition erreicht werden. Die Dekomposition von einer groben Prozessstruktur zur Aufgabenebene ist unser gewolltes Vorgehen.

Als Darstellungsmittel nutzen wir die BPMN. In Abschnitt 2.3 wurde die textuelle Analyse für BPMN bereits dargestellt.

Wir gehen jedoch wieder zielgerichtet vor. Wir fragen nach Inhalten, nicht nach Modellelementen. In unseren Projekten identifizieren wir die Inhalte in folgender Reihenfolge:

- (allgemeine) Prozessteilnehmer (dargestellt durch Pools)
- Rollen (dargestellt durch Lanes)
- Prozessereignisse (dargestellt durch Start-, Zwischen-, End-Ereignisse)
- Geschäftsaktivitäten (dargestellt durch Subprozesse, Aufgaben)
- Sequenzfluss und Verzweigungen (dargestellt durch Sequenzfluss, Gateways)
- Nachrichtenaustausch und Schnittstellen (dargestellt durch Nachrichtenfluss)
- Daten-/Informationsobjekte (dargestellt durch Datenobjekte).

Die Reihenfolge der Inhalte ist kein Zufall. Aus theoretischen und praktischen Gründen hat sich dieses Vorgehen bewährt, um zu einem gut strukturierten und pflegbaren Prozessmodell zu kommen.

Der Fragenkatalog für die Prozessmodellierung lautet daher bezogen auf einen zu betrachtenden Prozess:

- Welchen Nutzen bringt der Prozess dem Prozesskunden? Was ist das gewollte Ergebnis des Prozesses?
- Benenne bitte die Prozessbeteiligten.
- Welche dieser Prozessbeteiligten sind externe, welche interne Prozessbeteiligte?
- Benenne bitte Rollen¹⁰ innerhalb der internen Prozessbeteiligten.
- Welche Ereignisse verursachen einen Start des Prozesses?

¹⁰ Diese Frage setzt das Verständnis des Begriffs „Rolle“ voraus. Das wird hier nicht diskutiert.

- Was ist das gewünschte Ergebnis des Prozesses? Welche Endereignisse repräsentieren das Erreichen des gewünschten Ergebnisses?
- Gibt es wichtige Meilensteine innerhalb des Prozesses?
- Was sind die Aktivitäten innerhalb des Prozesses?
- Ordne die Aktivitäten in den Prozess ein (zwischen Startereignis und Meilenstein 1, zwischen Meilenstein 1 und Meilenstein 2, ..., zwischen Meilenstein n und dem Endereignis).
- Beschreibe die Aktivitäten kurz: Was ist das erwartete Ergebnis der Aktivität¹¹? Was ist Voraussetzung, damit die Aktivität ausgeführt werden kann? Welche Ausnahmen können bei der Ausführung der Aktivität auftreten?

Je nach Abstraktionsebene (Makroprozessebene, Subprozessebene, Aufgabenebene) werden wir Vorgaben machen, wie viele Aktivitäten wir darstellen wollen. Für die Makroprozessebene lautet die Vorgabe z.B. dass nicht mehr als 10 Subprozesse benannt werden. Aufgaben sind auf der Makroprozessebene nicht zulässig.

Zwischen den Fragen sind Zusammenhänge deutlich erkennbar. Z.B. ist es für die Benennung von Rollen hilfreich, Geschäftsaktivitäten im Prozess zu kennen und zu klassifizieren. Hier sei nochmals auf unser iteratives Vorgehen verwiesen. Die Reihenfolge der Fragen ist kein Dogma.

¹¹ Hier haben wir eine enge Verbindung zum Faktenmodell. Das Ergebnis muss im Faktenmodell präsent sein.

4 Wie textuelle Analyse wirklich funktioniert

In den Kapiteln 2 und 3 haben wir so getan, als ob wir mit der textuellen Analyse sofort Modellelemente identifizieren können. Das ist in der Praxis nicht so.

Das prinzipielle Vorgehen ist vielmehr:

1. **Identifizieren** von Kandidaten für Modellelemente
2. **Bewerten** der gefundenen Kandidaten
3. **Erzeugen** der Modellelemente für die relevanten Kandidaten entsprechend der von uns definierten Konventionen für die Modellierung

Was wir in Kapitel 2 und 3 dargestellt haben, deckt den Punkt 1 ab. Wir haben Kandidaten für die gewünschten Inhalte identifiziert. Wir haben dazu gezielt Fragen gestellt und erhaltene Informationen systematisch analysiert.

Für den zweiten Schritt gelten die am Anfang gemachten Aussagen ebenfalls: Jeder Modellierer nimmt diesen Schritt vor. Meist im Kopf. Selten werden die Bewertungskriterien festgelegt, selten wird das Ergebnis dokumentiert. Sicher spielt auch hier wieder die Werkzeugfrage eine Rolle. Nur wenige Werkzeuge gestatten die Dokumentation des Modellierungsprozesses in einer einfachen Art und Weise.

4.1 Bewerten der identifizierten Kandidaten

Für die Bewertung der gefundenen Kandidaten werden quantitative und qualitative Kriterien genutzt.

Die quantitative Bewertung nutzt statistische Informationen. Wie oft wurde das Konzept in unseren Informationssammlungen genannt? Wird das Konzept oft genannt, scheint es relevanter zu sein, als andere, nicht so häufig genannte Konzepte.

Eine rein quantitative Bewertung würde aber nicht ausreichen. Ein Konzept wird möglicherweise nur selten genannt, weil sich nur wenige Geschäftsanwender damit auskennen. Oder der Workshop oder Fragebogen war schlecht gestaltet und hat Suggestiv-Fragen gestellt. Die Antworten wurden so in eine bestimmte Richtung gelenkt. Wir müssen daher auch eine qualitative Bewertung vornehmen.

Wir müssen dafür den Kontext des jeweiligen Konzeptes analysieren. Ein Konzept wurde z.B. nur wenig genannt. Es steht aber in Beziehung zu einem wichtigen, häufig genannten Konzept. Es kann daher für unser Projekt sehr wohl relevant sein.

Auch bei der Formulierung unserer Fragen haben wir bereits eine Bewertung vorweggenommen. Wir haben z.B. nach den wichtigsten Konzepten gefragt. Dadurch haben wir andere, dem Befragten nicht wichtig erscheinende Konzepte ausgeschlossen. Möglicherweise sind diese aber durchaus für unser Projekt wichtig.

Hier hat unser iteratives Vorgehen Vorteile. In 3.2.2 haben wir bereits gesehen, dass aus einer Verfeinerung des Prozessmodells eine Verfeinerung des Faktenmodells resultiert.

Bei der Dekomposition unseres Makroprozesses werden wir wiederum unseren Fragekatalog anwenden. Da wir aber jetzt auf einer feineren Detail-Stufe sind, wird der Fragekatalog ergänzt und präzisiert. Die Fragestellungen werden bezogen auf die Konzepte der jeweiligen Detailstufe umformuliert. Wir sehen hier auch vor, dass der Geschäftsanwender uns weitere Informationen, die ihm wichtig erscheinen, mitteilen kann. Diese sind für die qualitative Beurteilung der Elementkandidaten sehr wichtig.

4.2 Sprache und textuelle Analyse

Ein Problem in der textuellen Analyse ist die Verwendung unterschiedlicher Bezeichnungen (siehe 3.2.2). Um diesem Problem beizukommen, werden wir die erhaltenen Informationen vereinheitlichen. Zuerst ersetzen wir synonyme Begriffe durch Begriffe, die wir für unser Projekt definiert haben. Wir formulieren Text um, so dass Statistik und Bewertung der gefundenen Element-Kandidaten einfacher wird. Dazu gehört auch die Vereinheitlichung der genutzten Verben. Diese sollen den Fakttypen entsprechen. Ein weiteres Problem ist dabei die Beugung der Verben, die in den meisten Werkzeugen nur schlecht berücksichtigt ist.

Im Allgemeinen ist Deutsch für die textuelle Analyse etwas sperriger als Englisch, da der Satzbau mehr Varianten aufweisen kann. Wie bei der Normalisierung von Geschäftsregeln wird der Text daher aufbereitet. Das muss jedoch so geschehen, dass später die Verständlichkeit durch den Fachanwender und die Rückverfolgbarkeit zur Ursprungsinformation noch gegeben ist.

5 Werkzeugunterstützung

Ich habe bereits betont, dass die Werkzeugunterstützung für die textuelle Analyse den Einsatz erheblich bestimmt.

Einzelne Modellierungswerkzeuge, z.B. Visual Paradigm, unterstützen die textuelle Analyse als Arbeitstechnik. Dabei wird nicht das gesamte hier skizzierte Spektrum abgedeckt. Vorteil einer solchen Werkzeugunterstützung ist die spätere Nachvollziehbarkeit. Es ist erkennbar, aus welchen Informationen Modellelemente abgeleitet wurden. Zum Teil ist das Vorgehen bei der Analyse der vorliegenden Informationen dokumentierbar.

Für die Erfassung der Informationen durch den Fachanwender wird häufig Word als Textprozessor genutzt. Hier bieten sich Add-Ins an, die dazu dienen Text in Modellinformation zu transformieren. EssWork von Ivar Jacobson International geht einen solchen Weg.

Eine neue Klasse von Anwendungen wird durch USoft definiert. USoft bezeichnet seine Produkte als „Language Based BPM“-Anwendungen. USoft fokussiert sich dabei auf sprachintensive Modelle, wie Vokabular, Anforderungen, Geschäftsregeln und Geschäftsprozesse. Die im vorliegenden Dokument beschriebenen Arbeitstechniken werden in der Anwendung unterstützt oder automatisiert. Textuelle Eingaben werden analysiert und in Modellelemente für das Vokabular, die Regelbasis oder Anforderungen transformiert. USoft geht aber noch einen Schritt weiter und nutzt das Vokabular, um z.B. BPMN-Modelle in eine sprachliche Darstellung zurück zu übersetzen. Auf diese Weise sind grafische Darstellung und sprachliche Präsentation immer synchronisiert.

Eine Bewertung von Werkzeugen hinsichtlich der Unterstützung der textuellen Analyse wird hier nicht betrachtet.

6 Zusammenfassung

„Textuelle Analyse“ ist eine Arbeitstechnik, die bei der systematischen Entwicklung von Modellen eingesetzt wird. Die Anwendung solcher Arbeitstechniken macht Modellieren zu einer ingenieurtechnischen Disziplin.

Textuelle Analyse geht davon aus, dass Modellinhalte aus Sprache abgeleitet werden und in Sprache zurück übersetzt werden. Der Hauptzweck von Geschäftsmodellen ist die Kommunikation von Geschäftsanwendern untereinander und mit anderen Projektbeteiligten.

Eine Betrachtung der Wortarten wie bei Abbott kann nur bestimmte, einfache Modellelemente identifizieren. Um gefundene Inhalte zu bewerten, Zusammenhänge zu erkennen und komplexe Sachverhalte abzuleiten, müssen die Satzglieder im Zusammenhang betrachtet werden. Ansätze wie die Zielfragetechnik oder RuleSpeak[®] tragen dem Rechnung.

Die Werkzeugunterstützung für die textuelle Analyse ist meist eingeschränkt und auf einzelne Modellelemente beschränkt.

Literatur

- Ross, Ronald G., Principles of the Business Rule Approach, Addison-Wesley, 2003
- Ross, Ronald G., Business Rule Concepts, Second Edition, Business Rule Solutions, LLC, 2005
- Zachman, John F., John Zachman, The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing, Zachman International, 2006, Electronic Book
- Ross, R., Pitschke, J., RuleSpeak® Satzformen, Business Rules in natürlich sprachlichem Deutsch spezifizieren, Business Rules Solutions, LLC und BCS, Dr. Jürgen Pitschke, Version 1.1, 2009
- Ross, R., Pitschke, J., RuleSpeak® Guidelines- Grundlagen Geschäftsregeln in natürlich sprachlichem Deutsch darstellen – was man tun sollte und was nicht, Business Rules Solutions, LLC und BCS, Dr. Jürgen Pitschke, Version 1.1, 2009
- Pitschke, J., RuleSpeak® - Kommentare zu den Basisdokumenten, BCS, Dr. Jürgen Pitschke, Version 1.1, 2009
- Abbott, Russel J., An Integrated Approach to Software Development, John Wiley & Sons, 1986
- Du , Richard T., Mentoring Object Technology Projects, Prentice Hall, 2002

Anhang A: Beispiel Service Desk

Ungeordnete Informationssammlung (Auszug)

Anfragen erreichen den Help Desk entweder über E-Mail, Telefon oder über das Endbenutzerportal. In jedem Fall werden grundsätzliche Informationen zur Anfrage und zum anfragenden Benutzer erfasst. Darüber können die Nutzer Serviceanfragen (Service Request) an die Servicemitarbeiter stellen. Die Serviceanfragen können Fragen zur Bedienung, Störungsmeldungen (Incident) oder Änderungswünsche (Change Request) sein.

Eingang der SR:

Email

- Inhalt und Betreff der Anfrage wird automatisch in das Beschreibungsfeld und den Titel übernommen
 - anfragende Endbenutzer anhand seiner E-Mail-Adresse identifiziert
 - Endbenutzer sind alle, die über den LDAP-Server 152.728.12.1 importiert werden. Zusätzlich werden noch Kunden der externen Unternehmen "RauBau" und "Sergo" per Hand bzw. per CSV-Liste importiert. Notwendige Benutzerdaten sind der Vor- und Nachname, das Passwort (mind. 6 Stellen) sowie E-Mail-Adresse, Passwort und Unternehmenszugehörigkeit. Nach Möglichkeit werden auch noch der Ort, Telefonnummer und Mobiltelefonnummer erfasst.
- ...

Support:

Support wird grundsätzlich für Software und Assets geleistet. Bei Assets, also den Geräten der IT-Infrastruktur werden allerdings nur solche vom technischen Support betrachtet, die in der Assetdatenbank erfasst sind. Dies schließt insbesondere Scanner und Plotter aus. Server, die als Webserver dienen werden mit hoher Priorität behandelt.

Der User Helpdesk ist in drei Ebenen (Level) organisiert:

- 1st Level: Annahme, kurzfristige Lösung und Techniker
- 2nd Level: interne Servicekräfte und Spezialisten
- 3rd Level: externe Servicekräfte und Spezialisten

1st level

- Wenn die Hauptkategorie der Anfrage "Hardware" ist, soll die Anfrage grundsätzlich an den First Level Support zugewiesen werden.
- Software Anfragen werden unabhängig von der Subkategorie stets zuerst vom First Level Support bearbeitet.
- Daneben werden die IT-Systeme über ein Monitoringsystem überwacht. Im Alarmfall, d.h. bei Überschreitung vorgegebener Schwellwerte, wird eine Information an den Servicemitarbeiter ausgelöst. Dieser nimmt diese Information als Störungsmeldung auf und leitet sie an den zuständigen Administrator weiter.
- Der 1st Level wird durch zwei Funktionen abgedeckt: den Servicemitarbeiter und den Techniker. Der Servicemitarbeiter steht dem Nutzer als direkter Ansprechpartner während der Servicezeit 8 – 18 Uhr zur Verfügung. Er identifiziert den Anrufenden, nimmt die Daten des Nutzers und die Anfrage in das Helpdesk-System auf. Emails an den Helpdesk werden automatisch als Anfrage erfasst. Der Servicemitarbeiter versucht die Anfrage mit den ihm zur Verfügung stehenden Mitteln zu lösen. Dazu gehören:
 - Eigenes Wissen und Erfahrung
 - Problemdatenbank des Helpdesk-Systems
 - Remote Steuerung vom System Management Server (SMS)

Der Techniker steht als „Feuerwehr“ beim Einsatz vor Ort zur Verfügung und übernimmt darüber hinaus folgende Aufgaben:

- Fehleranalyse (Hard- und Software)

- Installation
 - Technische Instandsetzung
 - Durchführen von Umzügen (Verkabeln, Patchen)
 - Arbeitsaufgaben, Installation nach Standardvorgaben
- Aufgrund der garantierten Erreichbarkeit von 8 – 18 Uhr für den Nutzer wird die Arbeit im Helpdesk in Schichten aufgeteilt. Die erste Schicht dauert von 8 – 13 Uhr, die zweite von 13 – 18 Uhr. Anwesend pro Schicht sind mindestens jeweils ein Servicemitarbeiter und ein Techniker.
- Der Benutzer ruft die zentrale Rufnummer an und schildert seine Anfrage dem Servicemitarbeiter des 1st Level. Dieser nimmt die Anfrage im System auf. Alternativ kann er die Anfrage per Email senden oder das Monitoringsystem eine Störung an den 1st Level melden. Der Servicemitarbeiter priorisiert und kategorisiert die Anfrage aufgrund der in den SLAs vereinbarten Lösungszeiten und nach Angaben des Benutzers.
- Er versucht, innerhalb von maximal 15 Minuten die Anfrage mit den ihm zur Verfügung stehenden Mitteln zu lösen. Gelingt die Lösung nicht innerhalb der vorgegebenen Zeit, wird die Anfrage je nach Art an den Techniker oder an den jeweils zuständigen Mitarbeiter des 2nd Level weitergeleitet.
- Der Techniker kommt vor Ort zum Einsatz und versucht das Problem dort zu lösen. Gelingt dies nicht, wird abgewogen, ob hier Garantie- oder Wartungsverträge greifen und der entsprechende Prozess durch den Techniker bzw. Servicemitarbeiter in die Wege geleitet. Liegen keine Verträge vor, so wird das Problem an den zuständigen Mitarbeiter des 2nd Level gegeben.
- ...

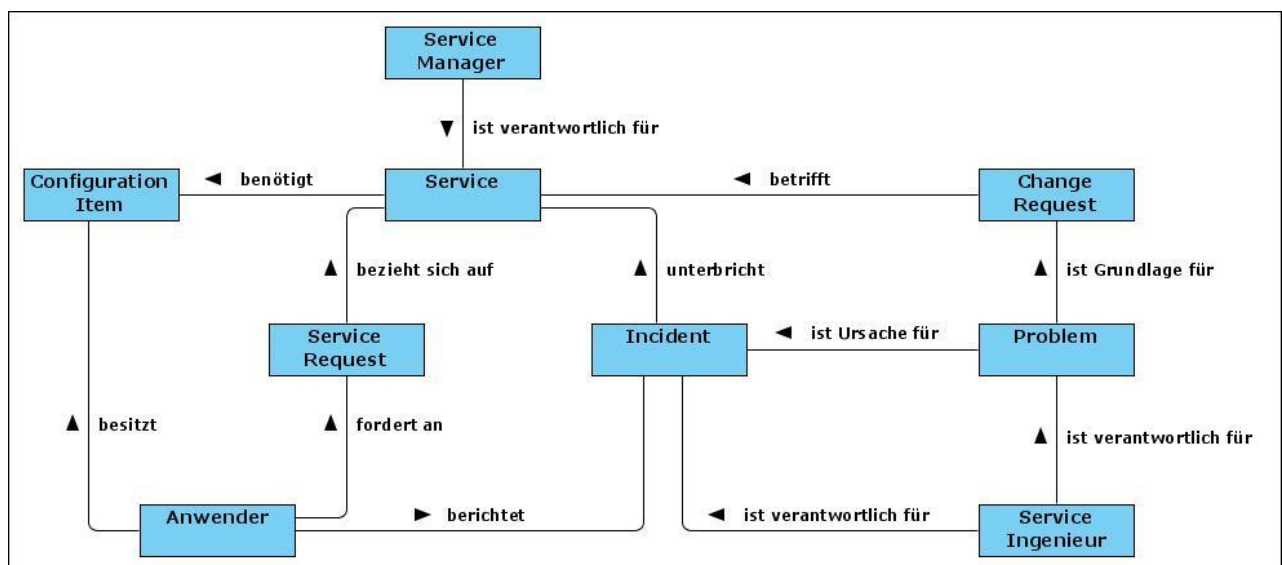
Supervisor

- Wenn ein Supervisor beobachtet oder von den Administratoren berichtet wird, dass mehrere Serviceanfragen mit wahrscheinlich gleicher Ursache vorliegen, kann er ein Problem eröffnen. Alle Anfragen, die vermutlich zusammenhängen werden mit dem neuen Problem verknüpft. Es wird je nach Sachlage entweder die Vorlage für Hardware-, Software- oder ein Service-Problem verwendet. Der Supervisor setzt verantwortliche Administratoren für den Gesamtprozess sowie für die einzelnen Prozessschritte ein und gibt den Prozess initial frei.
- Wenn das Problem zwar identifiziert wurde, aber eine Änderung an der IT-Infrastruktur oder an den angebotenen Dienstleistungen notwendig ist, muss eine Veränderungsanfrage (Change Request) vom Supervisor eröffnet werden. Diese Anfrage wird entweder aus der Vorlage "IT-Infrastruktur" oder "IT-Services" ausgewählt. Auch hier weist wieder der Supervisor für den Gesamtprozess und die einzelnen Schritte verantwortliche Administratoren zu. Bei den Phasen "Approve" und "Release" kann nicht freigewählt werden, sondern es sind immer der "Leiter-IT" für die Genehmigung der Planung und der "Verantwortliche Configuration Management" für die Aktualisierung der CMDB im "Release" Schritt einzusetzen. Nach der erfolgten Änderung der angebotenen Dienstleistungen bzw. der IT-Infrastruktur wird gegebenenfalls an die Verantwortlichen des zugrundeliegenden Problems Rückmeldung erstattet.

Begriffssammlung (Makroebene, Auszug)

- Anwender
Person, die für die Nutzung eines Service registriert ist
Synonyme: User, Service Nutzer
- Service
Leistung, die einen Mehrwert für den Anwender bietet
Synonym: Dienst
- Service Request
Anfrage eines Anwenders bezogen auf einen Service
- Incident
ungeplante Unterbrechung eines Service
Synonyme: Störung
- Problem
Ursache für einen oder mehrere Incidents
- Change Request
Anforderung zur Änderung eines Service
Synonym: Änderungsanforderung
- Configuration Item
Komponente, die für die Bereitstellung eines Service benötigt wird
- Service Ingenieur
Mitarbeiter im Service Desk
- Service Manager
Verantwortlicher für einen Service

Faktenmodell (Makroebene)



Prozessmodell (Makroprozess Incident Management)

