

X Jan 2

Natürlichsprachlicher Entwurf von Informationssystemen

Erich Ortner
Universität Konstanz, Informationswissenschaft
Lehrstuhl für Informationsmanagement
Postfach 5560
78434 Konstanz
Tel.: 07531-88-2976
Fax: 07531-88-2601
e-mail: ortner@inf-wiss.uni-konstanz.de

In dem Beitrag werden verschiedene linguistische Methoden der Informationssystementwicklung vorgestellt. Dabei wird der Versuch unternommen, sie nach einem einheitlichen Schema (Tabelle 1) zu klassifizieren. Zum Schluß wird auf einen Workshop zum "Natürlichsprachlichen Entwurf von Informationssystemen" hingewiesen, der vom 28.05. - 31.05.1996 in Tutzing stattfinden wird.

1. Einleitung

Linguistische Methoden der Informationssystem-Entwicklung sind dadurch charakterisiert, daß mit ihnen - meist syntaktisch und semantisch reglementierte - natürliche Sprachen (z. B. Englisch, Japanisch, Deutsch) bei der Entwicklung von Informationssystemen methodisch eingesetzt werden. Sie kommen komplementär zu semiformalen Diagrammethoden - z. B. aus dem Requirements Engineering, der Wissensrepräsentation oder aus der Datenbank-Anwendungsentwicklung - in der Anfangsphase einer Systementwicklung sowie bei der Einführung und Nutzung der Systeme zum Einsatz. Ein besonderes Kennzeichen ist dabei, daß ein Entwurfsergebnis mit ihnen nicht

wie beispielsweise bei Diagrammethode aus "Knoten" und "Kanten", sondern aus "Begriffen" (Fach- oder Themenwörtern), die zu "Aussagen" (Sätzen) verbunden werden, entwickelt wird. Die ersten Entwicklungsergebnisse sind Aussagensammlungen, aus denen alle weiteren Ergebnisse einer Informationssystemlösung - z. T. automatisiert - abgeleitet werden können.

Diese Entwicklung setzte Anfang der 80er Jahre ein, weil mit Konzepten wie "rapid prototyping", dem Einsatz "graphischer Entwurfsmethoden" oder einem unter "demokratischen Vorzeichen" praktizierten (benutzer-)partizipativen Entwicklungsstil entscheidende Qualitätsverbesserungen bei der Erstellung von Informationssystemen nicht mehr erzielt werden konnten. Vor allem die fachliche Software-Qualität (z. B. Konsistenz der Begriffsbildung bei Benutzern und den implementierten Systemen) läßt sich mit diesen Ansätzen nicht wesentlich verbessern, und der ingenieurmäßige Arbeitsprozeß - methodisches, auf Ideen und verantwortbaren Entscheidungen basiertes, rationelles Problemlösen - wird bei manchen Ansätzen teilweise entstellt dargestellt.

Neuere Ansätze tendieren deshalb dazu, mehr von dem "natürlichen" Sprach-einsatz der Benutzer durch linguistische Entwurfsmethoden in den Entwicklungs- und Nutzungsprozeß der Anwendungssoftware zu integrieren. Man geht dabei von der Feststellung aus, daß Menschen ihre Arbeitsprozesse (Business Process Engineering, Workflow-Modellierung) mit Fachsprachen organisieren. Desgleichen werden Fachsprachen zur Informationsverarbeitung und Kommunikation bei der Aufgabenerledigung in einer Organisation eingesetzt. Damit ist eine (natürliche) Sprache, wenn sie mit ihrer Grammatik (Satzbildungsregeln) und ihren relevanten Fachbegriffen (Fachtermini) aus den Anwendungsbereichen exakt festgelegt ist, ein "ideales" Konstruktionsinstrument sowohl für Organisationsstrukturen (Aufbau- und Ablauforganisation) als auch für Informationssysteme (z. B. Anwendungssystementwicklung) eines Unternehmens.

Chomsky [11] hat die Bedeutung einer "natürlichen Sprache" für Menschen einmal so charakterisiert: "Der Mensch lernt nicht eigentlich eine Sprache, sondern er lernt in einer Sprache schöpferisch tätig zu sein." Diese Einsicht scheint bei linguistischen Methoden der Informationssystementwicklung ein Leitmotiv ihres Einsatzes zu bilden.

2. Arbeiten zu linguistischen Methoden der Informationssystementwicklung

Die verschiedenen linguistischen Ansätze in Tabelle 1 können zunächst durch die Beantwortung der Frage eingeteilt werden, ob eine existierende (natürliche) Sprache zur Entwicklung der Informationssysteme eingesetzt wird, oder ob eine "quasi-natürliche" Sprache zu diesem Zweck gemeinsam mit Benutzern **neu** aufgebaut wird. Linguistische Entwicklungsmethoden vom ersten Typ werden empirische Ansätze und linguistische Methoden vom zweiten Typ konstruktive Ansätze genannt. Bei den empirischen Ansätzen können empirisch-analytische und empirisch-experimentelle Methoden unterschieden werden.

Empirisch-analytische Methoden stellen im Entwicklungsprozeß natürlichsprachliche Aussagen der Benutzer über Sachverhalte eines Anwendungsgebiets mit künstlichen formalen Mitteln (z. B. mit mathematischen Sprachen, Diagrammsprachen oder einer Logiksprache) "exakt" dar. In empirisch-experimentellen Projekten werden Kommunikationsverhältnisse, wie sie sich z. B. im Bürobereich ergeben, mittels protokollierter Sprechakte simuliert. Diese Ansätze wurden auf Basis von Sprechakttheorien - z.B. in Anlehnung an Austin [5] oder Searle [41] - entwickelt.

Bei den empirisch-analytischen Ansätzen werden die entwickelten Systeme aus der Intension der (Benutzer-)Fachbegriffe "interpretativ" erarbeitet. Bei empirisch-experimentellen Ansätzen kommt man primär über die Extension der (Benutzer-)Fachbegriffe zu einem Systementwurf.

Als Intension bezeichnet man seit Carnap [9] die Bedeutung oder auch die Merkmale bzw. die Definition eines Begriffs. In der Datenmodellierung werden z. B. Attribute, die zu einem Objekttyp (Fachbegriff) zusammengefaßt werden, die Intension eines Objekttyps [45] genannt. Unter der Extension eines Begriffs faßt man alle Objekte, für die ein Begriff zutrifft (die unter den Begriff fallen), zusammen. In der Datenmodellierung werden die Ausprägungen eines Objekttyps - also die Beschreibungen existierender (realer) Objekte (Tupel, Datensätze) - seine Extension [45] genannt.

Konstruktive linguistische Methoden (Tabelle 1) sind nicht empirisch (im Sinne von "aufdecken"), sondern normativ (im Sinne von "herstellen") definiert.

Position Gebiet	empirisch		konstruktiv
	analytisch	experimentell	
Datenbank- Anwendungs- entwicklung	<ul style="list-style-type: none"> - Chen '83 [10] - Colombetti et al. '85 [12] - Buitelaar/ van de Riet '92 [8] - Métais et al. '93 [31] - Tjoa/Berger '93 [43] 	<ul style="list-style-type: none"> - Albrecht et al. '95 [2] 	<ul style="list-style-type: none"> - Wedekind '81 [46] - Ortner '82 [35] - Ortner/Söllner '89 [36] - Vogler '94 [44]
Wissens- repräsen- tation	<ul style="list-style-type: none"> - Weigand '90 [48] - Dignum/van de Riet '91 [13] - Gerstl '92 [17] 		<ul style="list-style-type: none"> - Gunia '94 [18]
Objektorien- tierter Ent- wurf	<ul style="list-style-type: none"> - Abbott '83 [1] - Saeki et al. '89 [39] - Kristen '94 [24] - Mich/Garigliano '95 [32] 		<ul style="list-style-type: none"> - Wedekind '92 [47] - Schienmann '95 [40]
Büroinforma- tionssysteme (Groupware, Workflow- Systeme)		<ul style="list-style-type: none"> - Lehtinen/ Lyyti- nen '86 [27] - Winograd '87/88 [50] - Flores et al. '88 [16] - Auramäki et al. '88 [4] 	
Informations- systement- wicklung (generell)	<ul style="list-style-type: none"> - Wintraecken '89 [51] - Yonezaki '89 [52] - Rolland/ Proix '92 [38] - Lacity/ Janson '94 [25] 	<ul style="list-style-type: none"> - Andersen '91 [3] - Holmquist/ Andersen '91 [22] 	<ul style="list-style-type: none"> - Müller-Merbach '83 [33] - Ortner '94 [37]

Tabelle 1: Arbeiten zu linguistischen Methoden der Informationssystem-
entwicklung

Bei konstruktiven Ansätzen wird eine reglementierte Sprache zur Informationssystementwicklung zusätzlich hergestellt, während bei empirischen Ansätzen eine vorhandene Sprache im Entwicklungsprozeß (mit formalen Mitteln) "aufgedeckt" wird. Bei konstruktiven Ansätzen besteht ein Ziel oft darin, zwecks besserer Entwicklung und Nutzung der Informationssysteme, die faktische Sprachgenese in den Anwendungsbereichen um eine rekonstruierte normative Genese der Benutzerfachsprachen zu ergänzen:

Bezüglich einer Einteilung der Arbeiten zum natürlichsprachlichen Entwurf von Informationssystemen in Tabelle 1 kann man feststellen, daß empirische Ansätze im allgemeinen eine formale Sprache (Grammatik) und konstruktive Ansätze eine materiale Sprache (Grammatik und Fachlexikon) zur Entwicklung von Informationssystemen einsetzen [37].

2.1. Empirisch-analytische Methoden der Informationssystementwicklung

Empirisch-analytische Entwicklungsmethoden basieren i. d. R. auf einem formalen Sprachansatz. Neben einer (formalen) Objektsprache - z. B. einer Diagrammsprache -, die als Entwicklungssprache eingesetzt wird, ist eine weitere (materiale) Metasprache - in Form einer Benutzerfachsprache - erforderlich, in der die "Variablen" in den objektsprachlichen Konstrukten (z. B. Rechtecke, Pfeile, Dreiecke) ggf. gemeinsam mit Benutzern fachsprachlich interpretiert werden. Erst durch Interpretation ergeben sich aus konstruierten Aussageformen oder graphischen Konstrukten "inhaltliche" Aussagen.

Bei konstruktiven Entwicklungsmethoden werden im Vergleich dazu öfter materiale Entwicklungssprachen eingesetzt, die aus einer Grammatik (formaler Teil) **und** aus einem mit rekonstruierten Fachbegriffen "gefüllten" Lexikon (materialer Teil) bestehen. Eine Unterscheidung zwischen Objekt- und Metasprachen ist bei ihrem Einsatz nicht erforderlich.

In der Informatik als Ingenieurdisziplin überwiegen natürlich die formalsprachlichen Ansätze, während in traditionellen Ingenieurdisziplinen - z. B. im Maschinenbau - materiale Sprachen zum Einsatz kommen. Hier werden "darstellende Geometrie" und "technisches Zeichnen" als formaler Teil

(Grammatik) einer Konstruktionsprache und die Teildisziplin "Maschinenelemente" als ihr materialer Teil (Lexikon) angesehen.

Einem formalsprachlichen Ansatz folgend schlägt z. B. Chen [10] zur besseren Verständigung zwischen Entwicklern und Benutzern vor, den Informationsbedarf einer Anwendung zunächst in Englisch zu dokumentieren. Die englischen Beschreibungen werden dann von Datenbankdesignern in Entity-Relationship-Diagramme (oder ähnliche formale Repräsentationssprachen) "umgewandelt". In diesem Beitrag [10] untersucht Chen die (syntaktische) Übereinstimmung zwischen englischen Satzstrukturen und Entity-Relationship-Diagrammen und schlägt elf Regeln für die "Übersetzung" vor. Die Regeln für die Ermittlung (Rekonstruktion) der Fachbegriffe bleiben dabei allerdings im Dunkeln. Eine fachbegrifflich unreflektierte natürlichsprachliche Repräsentation wird in eine fachbegrifflich unreflektierte diagrammsprachliche Repräsentation transformiert. Die grundlegenden Konstrukte des Englischen - Substantiv, Verb, Adjektiv, Adverb, Gerundium, Satz - finden Gegenstücke in der Entity-Relationship-Diagrammtechnik.

Beim Entwurf von Informationssystemen werden Wörter (Fachtermini) mit einer festen Bedeutung benutzt. Dazu zählen Schlüsselwörter und Wörter, welche vom Entwerfer zur Bezeichnung von Entitäten, Beziehungen, Attributen, Werten, Akteuren etc. ausgewählt wurden. Gewöhnlich haben diese Wörter eine bestimmte Bedeutung, und manchmal stimmt ihr Gebrauch in den Anwendungsbereichen mit dieser Bedeutung überein. In einem Beitrag von Buitelaar und van de Riet [8] wird beschrieben, wie ein Lexikon, welches diese Wörter und ihre Bedeutung enthält, zusammen mit einem CASE (Computer Aided Software Engineering) - Tool erfolgreich eingesetzt werden kann.

Ein Werkzeug zum konzeptionellen Entwurf wird in [43] vorgestellt, das Bedarfsspezifikationen in natürlicher Sprache in Konstrukte des erweiterten Entity-Relationship-Approach transformiert. Ein Parsing-Algorithmus in Verbindung mit einer Grammatik und einem Lexikon wird entworfen, um den Anforderungen an das Werkzeug zu genügen. Die Ergebnisse des Parsings werden anhand von Regeln und Heuristiken weiter verarbeitet, womit eine "Beziehung" zwischen linguistischem Wissen und Entwurfsergebnissen aufgebaut wird. Das Werkzeug arbeitet interaktiv, um mehrdeutige, unvollständige und redundante Informationen handhaben zu können.

In seinem Buch "Linguistically Motivated Principles of Knowledge Base Systems" [48] beschreibt Weigand verschiedene Komponenten eines wissensbasierten Informationssystems von einem linguistischen Standpunkt aus. Dabei wird die Theorie der funktionalen Grammatik [14] als ein allgemeiner Rahmen übernommen. In der funktionalen Grammatik werden Strukturen und Eigenschaften der Ausdrücke in Form von Prädikaten repräsentiert. Aus einer wissensbasierten Perspektive sind diese Prädikate informationstragende Behauptungen, und die Prädikatensprache kann als höhere Wissensrepräsentationssprache betrachtet werden. Dies legt eine umfassende Integration von linguistischer und wissensbasierter Forschung nahe.

Im Projekt LILOG (linguistic and logic methods and tools) geht Gerstl [17] davon aus, daß "natürlichsprachliche Semantik" in Systemen zum Textverstehen abhängig ist von der linguistischen Oberflächenstruktur von Sätzen und dem Wissen über den Anwendungsbereich. Während linguistische Theorien dazu tendieren, den erstgenannten Aspekt zu untersuchen, scheinen Entwicklungen in der KI (Künstliche Intelligenz) den letztgenannten zu favorisieren. Die Arbeit in der Knowledge-Engineering-Group des LILOG-Projekts [21] an einem System zum Textverstehen, welches beide Aspekte integriert, gibt den Anlaß zu seinem Beitrag [17]. Das Hauptziel darin ist es, einige Schwierigkeiten bei der Verbindung von Anwendungswissen und Wissen über die Sprache aufzuzeigen und einige Möglichkeiten ihrer Überwindung vorzuschlagen.

Saeki et al. [39] stellen einen Prozeß zur inkrementellen Herleitung einer formalen Spezifikation, ausgehend von einer informalen Spezifikation in natürlicher Sprache, vor. Durch "Entwurfsaktivitäten" wird die Struktur der Softwaremodule, basierend auf einem objektorientierten Modell, von informalen englischen Beschreibungen extrahiert. Jedes Wort in den natürlichsprachlichen Sätzen wird mit einem Softwarekonzept, z. B. Klasse, Attribut oder Methode, verbunden. Besonderes Augenmerk wird Typen von Verbmustern geschenkt, die in Sätzen auftreten, um das System in seinem dynamischen Verhalten zu bestimmen. In "Ausführungsaktivitäten" wird der abgeleitete Modulentwurf verfeinert und präzisiert. Durch den "Entwurfs-/Ausführungs-Aktivitätenzyklus" kann die formale Spezifikation vervollständigt werden.

Die Fähigkeit von Computer-Anwendungen, Informationen zu verarbeiten und mit Benutzern zu kommunizieren sind stark eingeschränkt durch ihre

Unfähigkeit, "freiformatig" geschriebene oder gesprochene Sprache zu analysieren oder zu erzeugen. Während die Techniken, welche sich mit sehr beschränkten "formalen" Sprachen befassen, gut verstanden werden, sind Techniken, welche sich mit geschriebenem Text oder gesprochenem Wort befassen, noch nicht ausgereift. Die Verarbeitung solcher Informationen wird gemeinhin als "Natural Language Processing" (NLP) bezeichnet. Das LOLITA-System (large-scale object-based linguistic interactor and analyser) ist ein prototypisches NLP-System, das geschriebenes Englisch analysieren und erzeugen kann [32]. Einerseits dient LOLITA beim Informationssystementwurf der Vorverarbeitung, um Text zu korrigieren, zu selektieren und zu normalisieren. Andererseits unterstützt LOLITA die "automatische" Konstruktion objektorientierter Modelle anhand der natürlichsprachlichen Bedarfsspezifikation, repräsentiert in Form semantischer Netze [26].

In [52] wird ein Rahmen für ein System vorgestellt, welches ein Programm in einer natürlichen Sprache spezifiziert. Mit dem Ansatz wird eine neue Modulzerlegungsmethode, basierend auf einer lexikalischen Zerlegung, vorgestellt. Wörter in einer natürlichen Sprache korrespondieren mit den meisten "programmiersprachlichen" Notationseinheiten. Die Semantik der Wörter wird zunächst durch Sätze in natürlicher Sprache festgelegt, welche in Ausdrücke einer Modallogik übersetzt werden, so daß eine exakte Semantik bestimmt und eine semantische Untersuchung am Computer möglich wird. Der Übersetzungsmechanismus basiert auf dem Konzept der Montague-Grammatik [28]. Mehrere Beispiele zur Übersetzung für funktionale und dynamische Systemspezifikationen werden gezeigt. Auch wird eine Software-Entwicklungs-Umgebung (SEU) für das vorgestellte System präsentiert.

Ein Ansatz von Rolland und Proix [38] basiert auf der Prämisse, daß Wissensakquisition stark mit der Manipulation von Sprache zusammenhängt. Er stellt einen Versuch dar, die Sammlung von Aussagen zur Spezifikation einer Problemstellung, ihre Interpretation und ihre Modellierung durch den Gebrauch einer linguistischen Methode zu verbessern. Es wird vorgeschlagen, die Problemspezifikation für die Informationssystementwicklung durch Sätze in natürlicher Sprache auszudrücken. Es wird gezeigt, wie mit einem linguistischen Ansatz, der auf dem CASE-Gedanken beruht, automatisch die Modellierung eines Informationssystems durchgeführt werden kann. Der linguistische Ansatz und seine Implementierung als Expertensystem - bekannt als OICSI (französisches Akronym für "intelligentes Werkzeug für die Informationssy-

stementwicklung") - werden im Detail erläutert. Die zentrale These lautet, daß der linguistische Ansatz allgemeingültig ist, und zwar in zweifacher Hinsicht. So kann er an verschiedene Modellierungstechniken angepaßt und zusätzlich für ein weites Problemspektrum genutzt werden.

2.2. Empirisch-experimentelle Methoden der Informationssystementwicklung

Empirisch-experimentelle Methoden der Informationssystementwicklung haben zum Ziel, fachliche Entwicklungsergebnisse in der Praxis zu erproben (zu überprüfen) oder sie überhaupt erst aus der Praxis heraus schrittweise zu erarbeiten. In diesem Sinne werden konstruktive Entwicklungsmethoden - wie der nächste Abschnitt zeigen wird - auch "experimentell" eingesetzt. Jedoch wird bei ihnen nicht das Ziel verfolgt, eine bestehende Sprache zu analysieren, sondern eine neue, zu Benutzerfachsprachen komplementäre Fachsprache, eine sogenannte "Orthosprache" [29], zur Entwicklung und Nutzung von Informationssystemen aufzubauen.

Ein Fachentwurf kann überprüft werden, indem man in den fachbegrifflich festgelegten Grenzen das Geschehen in einem Anwendungsbereich simuliert. Handeln und Sprechen fallen dabei zusammen, sei es durch Beobachter (Protokollierung) und handelnde Personen getrennt oder von handelnden Personen selbst ausgeführt. Zur Überprüfung relevanter Aussagen beim Systementwurf werden Sprechhandlungen in Verbindung mit physischen Ausführungen (z. B. mit einem Bleistift schreiben und dabei das Wort "schreiben" aussprechen) gemeinsam ausgeführt.

Im anderen Falle (nur Sprechhandlungen liegen vor) wird von Entwicklern beim Systementwurf die Bedeutung relevanter Fachtermini in Gesprächen mit Benutzern über Handlungen und ihre Gegenstände in Anwendungsbereichen ermittelt. Hier ist die "naturwüchsige" Kontrolle der Aussagen (Sprechhandlungen) durch das begleitende konkrete Handeln nicht mehr möglich, und es ist nicht ausgeschlossen, daß sich Mißverständnisse ergeben.

In Anlehnung an den Sprachwissenschaftler Bühler [7] wird der erste Ansatz "experimentell-empraktisch" (empraktisch = mit der Praxis) und der zweite Ansatz "experimentell-epipraktisch" (epipraktisch = nach der Praxis) genannt.

Erhebungsverfahren wie die Simulation von Arbeitssituationen in Anwendungsbereichen - mit sprachlich begleitender Protokollierung der Sachverhalte durch ausführende Personen oder Beobachter - eignen sich zur empirischer Analyse eines Fachentwurfs. Dagegen kann z. B. durch Interviews, Fragebögen oder "Beispieldiskussionen" (auf rein sprachlicher Ebene) ein Fachentwurf epipraktisch überprüft werden.

Die vollständige und richtige Angabe der Semantik ist dabei insbesondere für große Anwendungen eine der schwierigsten Aufgaben im Datenbankentwurf. Im Rahmen des Projektes "Rapid Application Database Development" (RADD) werden Erfahrungen bei der informalen Modellierung von Strukturen einer Datenbank gesammelt. Mit dem in [2] vorgestellten Werkzeug wird die Semantik durch eine Analyse natürlichsprachlicher Beschreibungen, die Auswertung von Beispieldaten und Nachfragen in Form einer "Beispieldiskussion" (epipraktisch) gewonnen. Aus natürlichsprachlicher Information können Kandidaten geltender "Constraints" (Integritätsbedingungen) für den Datenbankentwurf abgeleitet werden. Durch Beispiele kann die intendierte Bedeutung der ermittelten Constraints geklärt und gemeinsam mit dem Experten validiert werden. Der Dialog läßt sich in einer einfachen und auch für (formal) ungeübte Experten verständlichen (natürlichsprachlichen) Form bis zur vollständigen Erfassung der Semantik fortführen.

In [27] werden Schwächen in einigen neueren Modellen zur Spezifikation von Büroinformationssystemen diskutiert. Ausgehend von der Überlegung, daß ein Informationssystem insbesondere eine Mittlerrolle in der organisationsinternen Kommunikation spielt, wird ein formales Modell eines Informationssystems aufgebaut. Dazu werden die Sprechakttheorie [5], [41] und neuere Ergebnisse der Organisationstheorie [49] herangezogen. Diese Theorien werden an Informationssysteme dahingehend angepaßt, daß sie eine formale Interpretation derselben liefern. Nicht die technische Spezifikation wird ermittelt, sondern die formalen Strukturen sozialer Interaktion mittels eines Informationssystems werden als Ausgangsbasis für ihren Entwurf erarbeitet.

Die Entwicklung eines (Büro-)Informationssystems geschieht gewöhnlich aus einer ganz bestimmten Perspektive, die die Entwurfsfragen, welche gestellt werden, sowie die Lösungen, welche gesucht werden, von vornherein beeinflusst. In [50] führt Winograd eine Perspektive ein, welche auf der Interpretation der Sprache als Handlung beruht. Die Konsequenzen für den Systement-

wurf werden untersucht. Es wird das Kommunikationstool "The Coordinator" beschrieben, das aus einer Sprache-/Handlungsperspektive heraus entworfen wurde. Dabei wird angedeutet, wie einige Aspekte koordinierter Arbeit mit dem Tool analysiert werden können. Die Sprache-/Handlungsperspektive wird anhand eines Beispiels illustriert, welches auf Studien über die Arbeit von Stationsschwestern im Krankenhaus beruht, und sie wird mit anderen aktuellen, verbreiteten Perspektiven verglichen.

In [4] stellen Auramäki et al. Methoden und Prinzipien dar, die dazu beitragen, Büros als Systeme kommunikativer Handlungen [19] zu untersuchen. Im Rahmen einer kommunikativen Handlung erzeugen Handelnde Bindungen durch symbolische Mittel. Der SAMPO-Ansatz (speech-act based office modeling approach), welcher die Büroaktivitäten als eine Folge von Sprechakten untersucht, die (symbolisch repräsentierte) Bindungen erzeugen, aufrecht erhalten, verändern, offenlegen und beenden, wird vorgestellt. Die wesentlichen Stufen und Methoden bei der Spezifikation eines Büroinformationssystems werden herausgehoben und ihre Anwendung anhand eines einfachen Beispiels illustriert. Abschließend werden Vor- und Nachteile des Ansatzes genannt und einige Schwerpunkte zukünftiger Forschung empfohlen.

Nach Andersen [3] stellen semiotische Ansätze die eigene Interpretation der Benutzer bezüglich ihrer Arbeit in den Vordergrund und nicht diejenigen von Methoden und Werkzeugen, welche vorgeben, eine objektive Beschreibung des Arbeitsprozesses zu bieten. Methode, Ergebnisse und Grenzen eines dieser Ansätze werden untersucht. Eine Schlüsselannahme des vorgeschlagenen Ansatzes besteht darin, daß Informationssysteme sinnvollerweise als Medium der sozialen Interaktion und nicht als Modelle der Realität betrachtet werden sollten. Anhand des Studiums der Sprache, und wie sie in Bezug zu Arbeitssituationen steht, lassen sich dabei Entwurfsideen ableiten, die aus der Wahrnehmung sich wiederholender Muster, Konzepte und Metaphern in der Art, wie Benutzer über ihre Arbeit sprechen, resultieren. Die Ergebnisse dieser Analyse vertiefen das Verständnis der Entwerfer von dem Anwendungsbereich, für den sie entwerfen.

2.3. Konstruktive Methoden der Informationssystementwicklung

Konstruktive Methoden der Informationssystementwicklung haben den **Aufbau** einer - zu der vorhandenen Fachsprache eines Anwendungsgebiets kom-

plementären - neuen quasi-natürlichen, reglementierten Fachsprache zum Ziel. Sie ist mit Beteiligung der Informationssystem-Benutzer so zu rekonstruieren, daß sich mit ihr alle entwicklungsrelevanten Sachverhalte eines Anwendungsgebiets klar, eindeutig und wörtlich (mit einer normierten Terminologie) darstellen lassen. Daher basieren konstruktive Entwicklungsmethoden i. d. R. auf einem materialen Sprachansatz. Die rekonstruierte und zur Entwicklung von Informationssystemen eingesetzte (materiale) Sprache besteht aus einer Grammatik (Satzbildungsregeln) und einem aus normierten Fachwörtern aufgebauten Fachlexikon (normierte Terminologie).

Eine komplette Methodologie zur Entwicklung von Datenbank-Anwendungen nach diesem, in Anlehnung an die konstruktive Wissenschaftstheorie [30] als "sprachkritisch" bezeichneten Ansatz wurde zuerst in [46] vorgestellt. Im Zentrum der Methodologie steht ein auf drei Abstraktionsebenen (fachliche Ebene, konstruktive Ebene, Schema-Ebene) organisierter Entwicklungsprozeß. Ausgangspunkt der Entwicklung ist ein Mangel in der Informationsversorgung eines Anwendungsbereichs, der auf einer fachlichen Ebene zu einer Aufgabenstellung führt. Durch Aufgabenabstraktion gelangt man zu einer konstruktiven Ebene, auf der die tragenden Fachbegriffe der Anwender wegen der Automatisierungsabsicht präzise "orthosprachlich" (ins rechte Lot gebracht) als Objekttypen festgelegt werden. Ein weiterer Abstraktionsschritt führt zur Schemaebene mit strengen Formalerfordernissen für Relationen (Tabellen) und Operatoren (Datenmanipulation). Aus dem neutralen Konzeptionellen Schema (Basistabellen) können Externe Schemata (Sichten auf Basistabellen) anwendungsbezogen gewonnen werden. Sie sind die Grundlage für den Programmentwurf und die Rückkehr zur konstruktiven Ebene. Ein weiterer Konkretionsschritt führt zur Lösung der Aufgabe auf der fachlichen Ebene, womit der methodologische "Kreis" geschlossen wird.

In [36] stellen die Autoren ein in einem Software-Unternehmen auf obigem Ansatz [46], [35] basierendes implementiertes Verfahren zur Datenmodellierung (Objekttypenmethode) vor, bei dem der methodische Entwicklungsprozeß mit der Sammlung relevanter (natürlichsprachlicher) Aussagen über Sachzusammenhänge in den Anwendungsbereichen beginnt. Es folgt die Klärung und Rekonstruktion wichtiger Fachbegriffe, wobei Regeln zur Behandlung von Synonymen, Homonymen, Äquipollenzen, Vagheit und falscher Bezeichner (Begriffswörter) angegeben werden. Aus der sich auf diese Weise ergebenden Aussagensammlung können durch Analyse der Satzstrukturen und unter Be-

rücksichtigung der rekonstruierten Fachbegriffe - z. T. werkzeugunterstützt - zunächst das Grobdatenmodell (Objekttypen und Beziehungen) und anschließend die Ergebnisse der Feindatenmodellierung (Attribute, funktionale Abhängigkeiten, Integritätsbedingungen) abgeleitet werden. Sämtliche Resultate der Entwicklungsarbeit - Aussagen, Objekttypen und Beziehungen, Attribute, Integritätsbedingungen - werden mit einem "Data Dictionary" (Repository, Metainformationssystem oder Entwicklungsdatenbank), das auch Beziehungen zu Ergebnissen späterer Entwicklungsphasen (logischer und physischer Entwurf) herstellt, strukturiert verwaltet. Erfahrungen mit dem Ansatz in dem Software-Unternehmen werden anhand von Kennzahlen dokumentiert.

Für die Objekttypenmethode [36] zur Datenmodellierung wird in [44] ein automatischer Übersetzer (OTMAR: Object-Type Model Translator) zur Übertragung reglementierter natürlichsprachlicher Aussagen in graphische Repräsentationen der Objekttypenmethode vorgestellt. OTMAR wurde mit den UNIX-Standardwerkzeugen LEX und YACC [20] realisiert. Dabei wird, gewissermaßen als "Filter", um kontrolliert von natürlichsprachlichen Aussagen zu syntaktisch (Aussageformen, Satzbaupläne) und semantisch (Fachtermini) normierten Aussagen überzugehen, eine **am Objekttypenansatz orientierte spezifische "Orthosprache"** als Referenzsprache definiert. Erst die in der "Orthosprache" reglementierten "natürlichsprachlichen" Aussagen werden mit OTMAR in graphische Repräsentationen der Objekttypenmethode automatisch überführt. Mit dem Werkzeug sind Transformationsregeln sowie Konsistenzbedingungen für die Übersetzung der Aussagen von einer Quellsprache (Orthosprache) in eine Zielsprache (Diagrammsprache) implementiert.

Die Entwicklung von Anwendungssystemen erfordert eine systematische Vorgehensweise bereits in den frühen Projektierungsphasen. Eine grundlegende Schwäche der meisten Modellierungsmethoden für den Fachentwurf ist die mangelnde benutzernahe systematische Konstruktion der Entwicklungsergebnisse aus dem Fachwissen der Anwendungsbereiche heraus. In [40] wird ein Ansatz vorgestellt, der die objektorientierte Spezifikation eines Fachkonzepts auf der Grundlage einer Rekonstruktion der Fachbegriffe eines Anwendungsbereichs in Zusammenarbeit mit den Anwendern ermöglicht. Unterteilt in die Aufgabenfelder Spezifikation, Repräsentation und Integration werden Lösungskonzepte für diesen "terminologie-basierten Ansatz" entwickelt und in

ein Vorgehensmodell für den Fachentwurf mit den Phasen "Rekonstruktion" und "Spezifikation" integriert.

Schließlich wird in [37] die Entwicklung einer am Orthosprachenkonzept der konstruktiven Wissenschaftstheorie [29] orientierten methodenneutralen Konstruktionssprache für Informationssysteme (und Organisationsstrukturen) vorgeschlagen. Um sie von formalsprachlichen Ansätzen der Informationssystementwicklung zu unterscheiden, wird die Konstruktionssprache auch "materiales Entwicklungssystem" (in Abgrenzung zu "formalen Entwicklungssystemen") genannt. Sie besteht aus einer Grammatik, in der die zulässigen Aussageformen (Satzbaupläne) zur Bildung von Aussagen über ein Anwendungsgebiet festgelegt sind, und einem Lexikon, das die normierten Fachwörter aus den Anwendungsbereichen konsistent verwaltet. Grammatik und Lexikon werden in dem Beitrag [37] ansatzweise definiert. An der Ausführung dieses Programms wird in einem Projekt KASPER (Konstanzer Sprachkritik-Programm für das Software-Engineering) gegenwärtig gearbeitet.

Die Aufzählung und Beschreibung der verschiedenen Methoden ist nicht vollständig. Eine Bibliographie zu linguistischen Methoden der Informationssystementwicklung ist in Vorbereitung. Auch die Grenzen zwischen den Arbeiten in Tabelle 1 sind für einige Methoden zu eng und für andere vielleicht zu weit gezogen worden. Ziel der Darstellung war es jedoch, von verschiedenen Grundpositionen ausgehend einige Entwicklungslinien für linguistische Methoden der Informationssystementwicklung anhand vorliegender Arbeiten aufzuzeigen.

3. Workshop zum natürlichsprachlichen Entwurf von Informationssystemen

Ein methodisch-natürlichsprachlicher Beginn der Entwicklung von Informationssystemen wird in der Software-Industrie [36] sowie bei Anwendern in manchen Unternehmen [6], [23] bereits praktiziert. Daneben gibt es - wie Tabelle 1 zeigt - zahlreiche Projekte, die eine Entwicklung von Methoden und Modellen sowie die Bereitstellung von Werkzeugen zur Unterstützung des natürlichsprachlichen Entwurfs von Informationssystemen zum Ziele haben. Die GI-Fachgruppe 2.5.2 EMISA (Entwicklungsmethoden für Informationssysteme)

steme und deren Anwendung) wird einen **Pfingstworkshop** zu diesem Thema in der Ev. Akademie Tutzing vom 28. - 30. Mai 1996 abhalten.

Mit dem Workshop ist geplant, eine Diskussion innerhalb und zwischen Teildisziplinen wie Requirements Engineering, Information Engineering, Datenbank-Anwendungsentwicklung, Wissensrepräsentation, (Computer-)Linguistik und Wissenschaftstheorie anzuregen. Als Themenkomplexe, die zur Orientierung für Beitragseinreichungen dienen sollen, werden

- Grundlagen methodisch-natürlichsprachlicher Systementwicklung,
- Entwicklung einzelner Methoden und ihre Integration in Vorgehensmodelle,
- Werkzeugentwicklungen und Werkzeugeinsatz,
- Projekte bei Anwendern sowie in Software-Unternehmen

vorgeschlagen. **Beiträge** im Umfang von 15 DIN A4 Seiten (1,5-zeilig) sind in 4-facher Ausfertigung bis zum **19.11.1995** an den Autor dieses Beitrags zu senden. Die Beiträge werden nach Beurteilung durch ein Programmkomitee in einem Tagungsband publiziert und den Teilnehmern vor Workshop-Beginn vorliegen.

Literatur

- [1] **Abbott, Russel J.:** Program Design by Informal English Descriptions. In: Communications of the ACM, Bd. 26, Nr. 11, November 1983, S. 882-894.
- [2] **Albrecht, M.; Buchholz, E.; Düsterhöft, A.; Thalheim, B.:** Ein Werkzeug zur Gewinnung semantischer Constraints aus natürlichsprachlichen Eingaben und Beispieldaten. In: Datenbanksysteme in Büro, Technik und Wissenschaft, GI-Fachtagung, Dresden, 22.-24. März 1995, hrsg. von Lausen, G., Berlin et al.: Springer-Verlag, 1995, S. 441-460.
- [3] **Andersen, P.B.:** A semiotic approach to construction and assessment of computer systems. In: Information Systems Research: Contemporary Approaches and Emergent Traditions, hrsg. von Nissen, H.-E., Klein, H.K. und Hirschheim, R., Elsevier Publishers B.V. (North-Holland), 1991, S. 465-514.

- [4] **Auramäki, E.; Lehtinen, E.; Lyytinen, K.:** A Speech-Act-Based Office Modeling Approach. In: ACM Transactions on Information Systems, Bd. 6, Nr. 2, April 1988, S. 126-152.
- [5] **Austin, J.L.:** How to do things with words, Harvard University Press, Cambridge Mass. 1962.
- [6] **Bechtel, M. et al.:** Datenmodellierung bei der Bühler AG, Uzwil (Schweiz) - Pensionskasse, Universität Konstanz, Informationswissenschaft, Bericht 3/92, Februar 1992.
- [7] **Bühler, K.:** Sprachtheorie, Die Darstellungsfunktion der Sprache, Verlag Ullstein GmbH, Frankfurt 1978.
- [8] **Buitelaar, P.; van de Riet, R.:** The Use of a Lexicon to Interpret ER Diagrams: a LIKE project. In: Entity-Relationship Approach - ER '92, Proceedings, Karlsruhe 7. - 9. Oktober 1992, hrsg. von Pernul, G. und Tjoa, A.M., Berlin et. al: Springer-Verlag, 1992, S. 162-177.
- [9] **Carnap, R.:** Bedeutung und Notwendigkeit - Eine Studie zur Semantik und modalen Logik, Library of Exact Philosophy Bd. 6, Springer Verlag, Wien/New York 1972.
- [10] **Chen, P.:** English Sentence Structure and Entity-Relationship Diagrams. In: Information Sciences, Bd. 29 (1983), Nr. 1, S. 127-149.
- [11] **Chomsky, N.:** Aspekte der Syntax-Theorie, Suhrkamp Verlag Frankfurt am Main/Akademie Verlag, Berlin 1969.
- [12] **Colombetti, M.; Guida, G.; Somalvico, M.:** NLDA: A Natural Language Reasoning System for the Analysis of Data Base Requirements. In: Methodology and Tools for Data Base Design, hrsg. von Ceri, St., Amsterdam et al.: North-Holland, 2. Auflage, 1985, S. 163-179.
- [13] **Dignum, F.; van de Riet, R.P.:** Knowledge base modelling based on linguistics and founded in logic. In: Data & Knowledge Engineering 7 (1991), S. 1-34.
- [14] **Dik, S.C.:** Theory of Functional Grammar, Dordrecht 1979.
- [15] **Fillmore, C.:** The case for case. In: Universals in linguistic theory, hrsg. von Bach, E. und Harms, R.T., New York et al.: Holt, Rinhart and Winston Inc., 1968, S. 1-90.
- [16] **Flores, Fernando; Graves, G.; Hartfield, B.; Winograd, T.:** Computer Systems and the Design of Organizational Interaction. In: ACM Transaction on Office Information Systems, Bd. 6, Nr. 2., April 1988, S. 153-172.

- [17] **Gerstl, P.:** Linking linguistic and non-linguistic information. In: Data & Knowledge Engineering, Bd. 8, Nr. 3, Juli 1992, S. 205-222.
- [18] **Gunia, H.:** Sprachkritische Entwicklung von Expertensystemen, Dissertation, Technische Fakultät der Universität Erlangen-Nürnberg, 1994.
- [19] **Habermas, J.:** Theorie des kommunikativen Handelns, Bd. 1, Frankfurt/M: Suhrkamp, 1981.
- [20] **Herold, H.:** Unix and seine Werkzeuge, Addison Wesley, Bonn 1992.
- [21] **Herzog, O.; Rollinger, C.-R. (Hsg.):** Text Understanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence, Final Report on the IBM Germany LILOG-Project, Springer-Verlag, Berlin et al. 1991.
- [22] **Holmquist, B.; Andersen, P.B.:** Language, Perspectives and Design. In: Design at Work, hrsg. von Greenbaum, J. und Kyng, M., Hillsdale, NJ, 1991, S. 91-119.
- [23] **Irion, A.M.:** Regelwerk und Qualitätscheckliste zur Bildung von Fachbegriffen bei der Entwicklung und Administration einer normierten Unternehmensfachsprache, Diplomarbeit, Universität Konstanz, Informationswissenschaft, Februar 1995.
- [24] **Kristen, G.:** Object-orientation: the KISS method: from information architecture to information system, Wokingham/England et al.: Addison-Wesley, 1994.
- [25] **Lacity, M. C.; Janson, M.A.:** Understanding Qualitative Data: A Framework of Text Analysis Methods. In: Journal of Management Information Systems, Fall 1994, Bd. 11, Nr. 2, S. 137-155.
- [26] **Lehmann, F. (Ed.):** Semantik Networks in Artificial Intelligence, Pergamon Press Ltd., New York 1992.
- [27] **Lehtinen, E.; Lyytinen, K.:** Action based model of information system. In: Information Systems, Bd. 11 (1986), Nr. 4, S. 299-317.
- [28] **Löbner, S.:** Einführung in die Montague-Grammatik, Kronberg 1976.
- [29] **Lorenzen, P.:** Semantisch normierte Orthosprachen. In: Zum normativen Fundament der Wissenschaft, hrsg. von Kambartel, F. und Mittelstraß, J., Frankfurt/M.: Athenäum, 1973, S. 231-249.
- [30] **Lorenzen, P.:** Lehrbuch der konstruktiven Wissenschaftstheorie, Mannheim et al.: B.I.-Wissenschaftsverlag, 1987.

- [31] **Métais, E.; Meunier, J.-N.; Levreau, G.:** Database Schema Design: A Perspective from Natural Language Techniques to Validation and View Integration. In: Entity-Relationship Approach - ER '93, Proceedings, Arlington 15.-17. Dezember 1993, hrsg. von Elmasri, R.A., Kouramajian, V. und Thalheim, B. Berlin et. al: Springer-Verlag, 1993, S. 190-205.
- [32] **Mich, L.; Garigliano, R.:** Object Oriented Modelling using the Natural Language Processing System LOLITA, in: ISCO 3, International Conference on Information System Concepts: Towards a Consolidation of View, Universität Marburg, 28.-30. März 1995, Marburg, 1995, S. A12-1 - A12-2.
- [33] **Müller-Merbach, H.:** Model Design Based on the Systems Approach. In: Journal of the operational Research Society, 34(1983)8, S. 739-751.
- [34] **Müller-Merbach, H.:** Design of Planning Models Supported by Relational Databases. In: Journal of the Operational Research Society, 42(1991)6, S. 522-523.
- [35] **Ortner, E.:** Aspekte einer Konstruktionsprache für den Datenbankentwurf, Dissertation, Fachbereich 1, Technische Hochschule Darmstadt, Mai 1982, als Buch: Darmstadt: S. Toeche-Mittler, 1983.
- [36] **Ortner, E.; Söllner, B.:** Semantische Datenmodellierung nach der Objekttypenmethode. In: Informatik-Spektrum, Bd. 8 (1989), Nr. 1, S. 31-42.
- [37] **Ortner, E.:** MELCHIOS: Methodenneutrale Konstruktionsprache für Informationssysteme, Universität Konstanz, Informationswissenschaft, Bericht 60-94, Konstanz, September 1994.
- [38] **Rolland, C.; Proix, C.:** A Natural Language Approach to Conceptual Modeling, in: Conceptual Modeling, Databases and CASE: an integrated view of information system development, New York et al.: Wiley, 1992, S. 447-464.
- [39] **Saeki, M.; Horai, H.; Enomoto, H.:** Software Development Process from Natural Language Specification. In: Proceedings of the 11th International Conference on Software Engineering, Pittsburg, Pennsylvania (USA): IEEE Computer Society Press, 1989, S. 64-73.
- [40] **Schienmann, B.:** Fachentwurf mit TAOS: ein Terminologiebasierter Ansatz für die Objektorientierte Spezifikation, Universität Konstanz, Informationswissenschaft, Bericht 67-95, März 1995.
- [41] **Searle, J.R.:** Speech Acts, Cambridge: Cambridge University Press, 1969.

- [42] **Sowa, J. F.:** Conceptual Structures: Information Processing in Mind and Machine, Reading/Massachusetts et al.: Addison-Wesley, 1984.
- [43] **Tjoa, A Min; Berger, L.:** Transformation of Requirement Specifications Expressed in Natural Language into an EER Model. In: Entity-Relationship Approach - ER '93, Proceedings, Arlington 15.-17. Dezember 1993, hrsg. von Elmasri, R.A. Kouramajian, V. und Thalheim, B., Berlin et. al: Springer-Verlag, 1993, S. 206-217.
- [44] **Vogler, M.:** OTMAR - Ein automatischer Übersetzer zur Konstruktion von Objekttypen und Beziehungen aus Aussagen, Diplomarbeit, Universität Konstanz, Informationswissenschaft, November 1994.
- [45] **Wedekind, H.; Ortner, E.:** Systematisches Konstruieren von Datenbankanwendungen - Zur Methodologie der Angewandten Informatik, Carl Hanser Verlag, München 1980.
- [46] **Wedekind, H.:** Datenbanksysteme I: eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung, 2., völlig neu bearbeitete Auflage, Mannheim et al.: B.I.-Wissenschaftsverlag, 1981.
- [47] **Wedekind, H.:** Objektorientierte Schemaentwicklung: ein kategorialer Ansatz für Datenbanken und Programmierung, Mannheim et al.: BI-Wissenschaftsverlag, 1992 (Reihe Informatik; 85).
- [48] **Weigand, H.:** Linguistically Motivated Principles of Knowledge Base Systems, Dordrecht et al.: Foris Publications, 1990.
- [49] **Williamson, O.E.:** Transaction-cost economics: the governance of contractual relations. In: J. Law Econ., 22(1979)2, S. 233-261.
- [50] **Winograd, T.:** A Language/Action Perspective on the Design of Cooperative Work. In: Human-Computer Interaction, Bd. 3 (1987-1988), S. 3-30.
- [51] **Wintraecken, J.J.V.R.:** The NIAM Information Analysis Method: Theory and Practice, Dordrecht et al.: Kluwer Academic Publishers, 1989.
- [52] **Yonezaki, N.:** Natural Language for Requirement Specification. In: Japanese perspectives in software engineering, hrsg. von Matsumoto und Ohne, Singapore et al.: Addison-Wesley, 1989, S. 41-71.