NORMSPRACHLICHE ENTWICKLUNG VON INFORMATIONSSYSTEMEN

Erich Ortner
Technische Universität Darmstadt
Entwicklung von Anwendungssystemen
Hochschulstr. 1, 64289 Darmstadt
http://www.winf1.bwl.tu-darmstadt.de/
ortner@bwl.tu-darmstadt.de

1. Einleitung

Der übliche Sprachansatz in der Informationssystementwicklung ist die Entwicklung von Anwendungssoftware mit formalen Sprachen. Die benutzerrelevante Terminologie (Fachwörter) kommt beim Einsatz einer „formalen Sprache“ (z. B. Diagrammsprachen, Programmiersprachen) mit Hilfe einer „Metasprache“ – bei der es sich um die materiale (Benutzer-)Fachsprache handelt – nach dem Konzept einer Interpretationssemantik [Verbalisierung, Ober92] bei der Anwendungsentwicklung hinzu. Dadurch kann die Korrektheit der Entwicklungsergebnisse lediglich im Hinblick auf ihre Struktur (anhand der kalkülisierten oder axiomatisierten formalen Sprache) „automatisch“ (z.B. im Hinblick auf die modellierte Kardinalität von Datenabbildungen) überprüft werden, während die verwendete Fachterminologie aus der Metasprache (Fachsprache) der Benutzer i. d. R. unrekonstruiert und unnormiert übernommen wird. Ein weiteres Problem in der partizipativen Anwendungsentwicklung resultiert aus der großen Zahl verschiedener Modellierungsmethoden (Sprachen, Vorgehensweisen), die bei der Entwicklung unterschiedlicher Typen von Anwendungssystemlösungen eingesetzt werden. Für die Entwicklung von Datenbank-Anwendungen kommen in der Regel andere Modellierungsmethoden als für die Entwicklung von objektorientierten Anwendungssystemlösungen oder Expertensystem-Anwendungen zum Einsatz. Kaum ein Softwareentwickler (-entwicklerin) überblickt heute sämtliche relevanten Entwicklungsmethoden für Anwendungssysteme, und erst recht kann von Anwendern nicht

erwartet werden, daß sie sich zur Rekonstruktion des anwendungsspezifischen Fachwissens mit den Entwicklern diese Methoden aneignen.

2. Materialsprachlichkeit

Bei einem materialsprachlichen Ansatz in der Anwendungssystementwicklung gehört die rekonstruierte Fachterminologie eines Anwendungsgebiets (z. B. der Betriebswirtschaftslehre, dem Maschinenbau oder der Medizin) – in Form eines Lexikons verwaltet – mit zu der eingesetzten Entwicklungssprache (Domain-Specific-Language). Die Korrektheit der Anwendung wird sowohl im Hinblick auf ihre Struktureigenschaften (z. B. Kardinalität) als auch im Hinblick auf die eingesetzte Fachterminologie (korrekter Einsatz von Fachwörtern) mit Hilfe der implementierten materialen Entwicklungssprache (Grammatik und gefülltes Lexikon) überprüft. Eine andere Variante des materialsprachlichen Ansatzes in der Anwendungssystementwicklung ist die Zusammensetzung von Softwarelösungen aus inhaltlich (fachlich) definierten, normierten Komponenten (sogenannten Business Objects oder Anwendungselementen). Dagegen gehört die Entwicklung von Informationssystemen auf der Grundlage domänenspezifischer Referenzmodelle (z. B. eines branchenspezifischen oder unternehmensweiten Datenmodells) nicht in die Kategorie materialsprachlicher Ansätze. Ein Referenzmodell wird hierbei als eine globale materiale Lösung (als Sprachprodukt, Entwicklungsergebnis) und nicht als Teil einer zum Einsatz kommenden materialen Konstruktionssprache (materialsprachlicher Ansatz) verstanden.

3. Methodenneutralität

Um dem Problem der Sprachvielfalt in der Anwendungssystementwicklung zu begegnen, empfiehlt sich eine Zweiteilung des Fachentwurfs in einen methodenneutralen und einen methodenspezifischen Teil. Wir setzen eine materiale Entwicklungssprache – eine sogenannte Normsprache - vor allem im 1. Teil des Fachentwurfs eines Vorgehensmodells für die Anwendungssystementwicklung ein. Dabei geht es zunächst um die Erarbeitung eines Fachentwurfs mit den Anwendern als Fachexperten, ohne spezielle Architekturkonzepte für die Software, wie sie z. B. Datenbank-Anwendungen, die objektorientierte Entwicklung von Anwendungssystemen oder Workflow-Management-Anwendungen aufweisen, berücksichtigen zu müssen. Dadurch werden die Anwender in diesem ersten Teil des Fachentwurfs von Konzepten wie z. B „Relation“, „Transaktion“,

„Workflow“, „Klasse“ oder „Objekt“ bei der Rekonstruktion des entwurfsrelevanten Fachwissens entlastet. Der zweite Teil des Fachentwurfs findet dann – ohne Beteiligung der Benutzer – methodenspezifisch in verschiedenen Diagrammsprachen für die Teilergebnisse (z. B. für Datenschemata, für den Funktionsentwurf für die Kontrollflußmodellierung) einer fachlichen Softwarelösung statt. Die Transformationen der Entwicklungsergebnisse von einer Sprachform in die andere können werkzeugunterstützt erfolgen.

4. Normsprachlichkeit

Fachwörter (Termini) für Gegenstände und Eigenschaften in den Anwendungsbereichen sind für die Entwicklung von Anwendungssoftware nicht „Schall und Rauch“ – wie manche Formalisten in der Softwareentwicklung annehmen -, sondern sie entscheiden darüber, auf welchen Gegenstandsbereich die implementierten Programme in den Anwendungsbereichen anzuwenden sind. Daher sind sie während der Systementwicklung methodisch zu rekonstruieren und aus Sicht der Systeme und ihrer Anwender einheitlich zu normieren. Rechnet man die rekonstruierten Fachwörter zur Syntax (rekonstruierten Grammatik) der eingesetzten anwendungsbezogenen Entwicklungssprache (Domain-Specific-Language) hinzu, so liegt eine normierte (materiale) Fachsprache zur Entwicklung von Informationssystemen für bestimmte Anwendungsgebiete (z. B. kaufmännische Informationssysteme, medizinische Informationssysteme etc.) vor. Von einer materialen Fachsprache (Konstruktionssprache, Normsprache) aus läßt sich die normierte Terminologie flexibler und mit geringerem Aufwand als über Referenzmodelle (globale Anwendungen) auf der Seite der entwickelten Anwendungen verwalten. Größtenteils gelöst ist inzwischen die Frage, wie eine materiale Normsprache für den methodenneutralen Entwurf von Anwendungssystemen – mit einer methodenneutralen Syntax (Grammatik) und einer methodenneutralen Terminologie (Lexikon) – aufgebaut (rekonstruiert) werden kann. Die Arbeiten der konstruktiven Wissenschaftstheorie [KaLo79] bilden hierfür die entscheidende Grundlage. Dabei besteht eine materiale Sprache grundsätzlich aus den Komponenten „Gegenstandseinteilung“ (Wie wird ein Weltausschnitt kategorial (allgemein) eingeteilt?), „Grammatik“ (Nach welchen Regeln (Handlungsvorschriften) werden Wörter (Begriffe) zu Sätzen (Aussagen) verbunden?) und „Lexikon“ (Welche Wörter sind in welchem Kontext zugelassen?). Das Lebenswerk Paul Lorenzens mündet in dieser Hinsicht in den Nachweis [Lore87], daß dazu die komplizierten Verhältnisse „natürlicher Sprachen“ nicht vollständig analysiert (Programm der analytischen Wissenschaftstheorie), sondern

eine annähernd in gleicher Weise wie die natürlichen Sprachen mächtige, jedoch präzise Sprache von Anfang an (ohne Zirkel) methodisch konstruiert werden kann.

5. Fazit

Der Einsatz einer materialen Sprache für die Anwendungssystementwicklung wurde – ausgehend von der konstruktiven Wissenschaftstheorie [KaLo73] - zunächst in [WeOr80] vorgeschlagen. Es folgte im Hinblick auf eine „materialsprachliche“ Entwicklung von Datenstrukturen aus anwendungsspezifischen (materialen) „Datenelementen“ eine Implementierung des Ansatzes [OrRS90] bei einem Software-Unternehmen. Eine Realisierung des materialsprachlichen Ansatzes für die objektorientierte Entwicklung von Informationssystemen wird in allen relevanten Aspekten in [Schi97] detailliert beschrieben. In US-amerikanischen Publikationen wird inzwischen über den praktischen Einsatz rekonstruierter materialer Sprachen in Software-Projekten – z. B. in [McDa96, Business Language] oder in [Robe97, Domain-Specific-Languages] – berichtet. Ähnliche Forderungen und Ansätze zu einer Kontrolle und Verwaltung der Terminologie eines Anwendungsbereichs in einem Lexikon für die Nutzung in der Anwendungsentwicklung finden sich z. B. bei Sowa [Sowa92] und in den Arbeiten [z. B. BuRi92] zum LIKE-Projekt (Linguistic Instruments in Knowledge Engineering) sowie in [BvdR94] zum COLOR-X-Projekt (Conceptual Linguistically-based Object-oriented Representation Language for Information and Communication Systems). Ein formaler Rahmen zur Festlegung und Verwaltung solcher Terminologien bzw. „Ontologien“ mit der Vorgabe einer terminologischen Logik ist beispielsweise in der Wissensrepräsentation in der Entwicklung der KL-ONE-Sprache [BrSc85], [WoSc92] zu finden. Ansätze zu einer „Terminitheorie“ finden sich in den Arbeiten von Kamlah/Lorenzen [KaLo73], Lorenzen [Lore73], Sinowjew [SiWe75] und Wessel [Wess84].

LITERATUR

[BrSc85] **Brachmann, R. J.; Schmolze, J. G.**: An Overview of KL-ONE Knowledge Representation System, in Cognitive Science, 9 (1985) 2, S. 171 – 216.

[BuRi92] **Buitelaar, P.; Riet, van de R. P.**: The Use of a Lexicon to Interpret ER Diagrams, in: Tjoa, A. M. (Hrsg.); Entity Relationship Approach – ER'92, Proceedings, Springer Verlag, Berlin 1992, S. 162 – 176.

[BvdR94] **Burg, J. F. M.; Riet, van de R. P.**: COLOR-X: Object Modelling profits from Linguistics, Technical Report IR – 365, Vrije Universiteit, Amsterdam 1994.

[KaLo73] **Kamlah, W.; Lorenzen, P.**: Logische Propädeutik, B. I. – Wissenschaftsverlag, Mannheim etc. 1973.

[Lore73] **Lorenzen, P.**: Semantisch normierte Orthosprachen, in: Kambartel, F. Mittelstraß, J. (Hrsg.); Zum normativen Fundament der Wissenschaften, Athenäum Verlag, Frankfurt a. M. 1973.

[Lore87] **Lorenzen, P.**: Lehrbuch der konstruktiven Wissenschaftstheorie, B. I. – Wissenschaftsverlag, Mannheim/Wien/Zürich 1987.

[McDa96] **McDavid, D. W.**: Business Language Analysis for Object-oriented Information Systems, in: IBM Systems Journal, 35 (1996) 2, S. 128 – 150.

[Ober92] **Oberschelp, A**: Logik für Philosophen, B. I. – Wissenschaftsverlag, Mannheim etc. 1992

[OrRS90] **Ortner, E.; Rössner, J.; Söllner, B.**: Entwicklung und Verwaltung standardisierter Datenelemente, in: Informatik-Spektrum, 13 (1990) 1, S. 17 – 30.

[Robe97] **Robertson, P.**: Integrating Legacy Systems with Modern Corporate Applications, in: Comm. of the ACM, 40 (1997) 5, S. 39 – 46.

[Schi97] **Schienmann, B.**: Objektorientierter Fachentwurf, B. G. Teubner Verlagsgesellschaft, Stuttgart/Leipzig 1997.

[SiWe75] **Sinowjew, A.; Wessel, H.**: Logische Sprachregeln, Eine Einführung in die Logik, Fink Verlag, München 1975.

[Sowa92] **Sowa, J. F.**: Logical Structures in the Lexicon, in: Knowledge Based Systems, 5 (1993) 3, S. 173 – 182.

[WeOr80] **Wedekind, H.; Ortner, E.**: Systematisches Konstruieren von Datenbankanwendungen, Carl Hanser Verlag, München/Wien 1980.

[Wess84] **Wessel, H.**: Logik, Deutscher Verlag der Wissenschaften, Berlin 1984.

[WoSc92] **Woods, W. A.; Schmolze, J. G.**: The KL-ONE Family, in : Lehmann, M. M. (Hrsg.); Semantic Networks in Artificial Intelligence, Pergamon Press, Oxford 1992, S. 133 – 177.