

Literaturverzeichnis

- ANSI (2001):** ATIS Telecom Glossary 2000, www.atis.org/tg2k/, Abgerufen am 28.06.06, 2001.
- ANSI/X3/SPARC (1975):** Study Group on Database Management – Systems-Interim-Report. In: Bulletin of ACM SIGMOD, Bd. 7, Nr. 2, 1975.
- Ackermann, J. et al. (2002):** Vereinheitlichte Spezifikation von Fachkomponenten. In: Memorandum des Arbeitskreises 5.10.3 der Gesellschaft für Informatik e.V. „Komponentenorientierte betriebliche Anwendungssysteme, Augsburg, 2002.
- Alexander, C. et al. (1977):** A Pattern Language: Towns – Buildings – Construction, 26. Aufl., New York, 1977.
- Alexander, C. (1979):** Center for Environmental Structure: The timeless Way of Building, New York, 1979.
- Alexander, C. (1999):** The Origins of Pattern Theory, the Future of the Theory, and the Generation of a Living World. In: IEEE Software, 1999, S. 71-82.
- Andreou, A. S. et al. (2002):** Mobile Commerce Applications and Services: A Design and Development Approach. In: Proceedings of the First International Conference on Mobile Business (M-Business 2002), Athen, 2002, S. 1-9.
- Arildson, D.; Dedo, D. (2002):** Pocket PC Systems Management - White Paper Microsoft, <http://www.microsoft.com/technet/archive/itsolutions/mobile/evaluate/pctpcmgmt.mspx?mfr=true>, Abgerufen am 08.12.06, 2002.
- Austin, D.; Barbir, A.; Garg, S. (2002):** Web Services Architecture Requirements, <http://www.w3.org/TR/wsa-reqs>, Abgerufen am 28.06.06, 2002.
- Austin, J. C. (1962):** How to Do Things with Words, Cambridge, 1962.
- Aßmann, U.; Neumann, R. (2003):** Quo vadis Komponentensysteme? In: HMD - Praxis der Wirtschaftsinformatik, Nr.231, 2003, S. 19-27.
- Babiak, U. (1999):** Effektive Suche im Internet: Suchstrategien, Methoden, Quellen, 3. akt. u. erw. Aufl., Beijing [u. a.], 1999.
- Balzert, H. (1989):** Die Entwicklung von Software-Systemen: Prinzipien, Methoden, Sprachen, Werkzeuge, unveränd. Nachdr., Mannheim [u. a.], 1989.
- Bass, L.; Clements, P.; Kazman, R. (2003):** SEI Series in Software Engineering, Software Architecture in Practice, 2. Aufl., Reading [u. a.], 2003.

- Beitz, A.; Bearman, M.; Vogel, A. (1995):** Service Location in an Open Distributed Environment. In: Proceedings of 2nd Workshop on Services in Distributed and Networked Environments, Whistler, 1995, S. 28-34.
- Bell, G.; Gray, J. N. (1997):** The Revolution Yet to Happen. In: Denning, P. J.; Metcalfe, R. M. (Hrsg.): Beyond Calculation – The next fifty Years of Computing, New York, 1997, S. 5-32.
- Berbner, R. et al. (2005):** Management of Service-oriented Architecture (SoA)-based Application Systems. In: LNI-Proceedings of Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'05), Bd. 2, Nr. 1, 2005, S. 208-211.
- Berkeley University of California (o. J.):** SETI@home, www.setiathome.ssl.berkeley.edu/, Abgerufen am 18.04.06, o. J.
- Bienert, P. (1998):** Information und Kommunikation: Technik und Anwendung in Wirtschaft und Medien, Berlin [u. a.], 1998.
- Biggerstaff, T. J.; Perlis, A. J. (1989):** Software Reusability, Bd. 1, New York, 1989.
- Bluetooth SIG (1999):** Service Discovery Protocol (SDP), http://www.pday.com.cn/technology/bluetooth_documents/sdp.pdf, Abgerufen am 12.04.06, 1999.
- Bluetooth SIG (2003):** Specification of the Bluetooth System, www.bluetooth.org/foundry/adopters/document/Bluetooth_Core_Specification_v1.2, Abgerufen am 05.06.06, 2003.
- Bodendorf, F. (2006):** Daten- und Wissensmanagement, 2. akt. u. erw. Aufl., Berlin [u. a.], 2006.
- Bodoff, S. et al. (2002):** The J2EE Tutorial, Indianapolis, 2002.
- Boehm, B. (1988):** A Spiral Model of Development and Enhancement. In: IEEE Computer 21, Nr. 5, 1988, S. 61-72.
- Boutilier, C. et al. (2003):** Cooperative Negotiation in Autonomic Systems using Incremental Utility Elicitation. In: Uncertainty in Artificial Intelligence, 2003, S. 89-97.
- Brabänder, E.; Klückmann, J. (2006):** Geschäftsprozessmanagement als Grundlage für SOA. In: OBJEKTspektrum, Nr. 5, 2006, S. 32-40.
- Brandt, C.; Lonthoff, J. (2004):** Vorschlag für eine Architektur zur Personalisierung und Individualisierung mobiler Dienste. In: Höpfner, H.; Saake, G. (Hrsg.): Workshop Grundlagen und Anwendungen mobiler Informationstechnologie, Preprint Nr. 4, Magdeburg, 2004, S. 33-40.
- Broy, M. et al. (1998):** What characterizes a (software) component? In: Software – Concepts and Tools, Bd. 19, Nr. 1, 1998, S. 49-56.

- Buxmann, P. (1996):** Standardisierung betrieblicher Informationssysteme, Dissertation Universität Frankfurt, Fakultät Betriebswirtschaftslehre, insbesondere Wirtschaftsinformatik, Wiesbaden, 1996.
- Böhm, O. (2006):** Aspektorientierte Programmierung mit AspectJ 5: Einsteigen in AspectJ und AOP, Heidelberg, 2006.
- Capra, L.; Emmerich, W.; Mascolo, C. (2002):** Middleware for Mobile Computing. In: LNCS-Proceedings Advanced Lectures on Networking: Networking 2002 Tutorials, Bd. 2497, Berlin [u. a.], 2002, S. 20-58.
- Cerwenka, P. et al. (2000):** Kompendium der Verkehrssystemplanung, Wien, 2000.
- Chittaro, L. (2003):** Human-Computer Interaction with Mobile Devices and Services, LNCS-Proceedings of 5th International Symposium on Mobile HCI 2003, Bd. 2795, Berlin [u. a.], 2003.
- Clark, J. (1993):** Human Resource Management and Technical Change, London [u. a.], 1993.
- Coad, P.; Nicola, J. (1993):** OOObject-Oriented Programming, Englewood Cliffs, 1993.
- ComponentSource (o. J.):** The Open Market, <http://www.componentsource.com/Services/TheOpenMarket.asp>, Abgerufen am 11.04.06, o. J.
- Councill, W. T.; Heinemann, G. T. (2001):** Definition of a Software Component and its Elements. In: Heinemann, G. T.; Councill, W. T. (Hrsg.): Component-based Software Engineering, Upper Saddle River, 2001, S. 6-19.
- Crnkovic, I.; Larsson, M. (2000):** Component Configuration Management. In: Bosh, J. et al. (Hrsg.): Proceedings of the 5th Int. Workshop on Component-Oriented Programming (WCOP 2000). Research Report No 15/00, Blekinge Institute of Technology, Sophia Antipolis, 2000, S. 87-90.
- Davenport, T. H.; Beck, J. C. (2001):** The Attention Economy: Understanding the New Currency of Business, Boston, 2001.
- Deemer, K. (1980):** A Life Cycle Model of New Product Profitability. In: Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small Systems, Palo Alto, 1980, S. 150-155.
- Deitel, H. M.; Deitel, P. J. (1997):** Java how to program, Upper Saddle River, 1997.
- Dix, A. J. (2004):** Human-Computer Interaction, 3. bearb. Aufl., London [u. a.], 2004.
- Dornbusch, P.; Huber, M. (2003):** Generierung von Ortsinformationen durch User-Communities. In: Uhr, W.; Esswein, W.; Schoop, E. (Hrsg.): Wirtschaftsinformatik 2003 – Medien, Märkte, Mobilität, Bd. 1, Heidelberg [u. a.], 2003, S. 175-186.

- Dostal, W. et al. (2005):** Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis, Heidelberg [u. a.], 2005.
- Drucker, P. (2005):** Was ist Management?: Das Beste aus 50 Jahren, 3. Aufl., Berlin, 2005.
- ECMA (2005):** Standard ECMA-335 Common Language Infrastructure (CLI), 3rd edition, <http://www.ecma-international.org/publications/standards/Ecma-335.htm>, Abgerufen am 11.04.06, 2005.
- ECMA (2006):** Standard ECMA-335 Common Language Infrastructure (CLI), 4th edition, <http://www.ecma-international.org/publications/standards/Ecma-335.htm>, Abgerufen am 20.12.06, 2006.
- Eckert, C. (2004):** IT-Sicherheit: Konzept – Verfahren – Protokolle, 3. überarb. u. erw. Aufl., München [u. a.], 2004.
- Eicker, S.; Nietsch, M. (1999):** Standards zum objektorientierten Paradigma. In: Wirtschaftsinformatik, Bd. 4, Nr. 5, 1999, S. 358-370.
- Eicker, S.; Schwichtenberg, H.; Büning, W. (2001):** Ein Framework für Shop-Systeme auf Basis der Java 2 Enterprise Edition. In: Buhl, H. U.; Huther, A.; Reitwiesner, B. (Hrsg.): Information Age Economy, Heidelberg, 2001, S. 637-650.
- Eicker, S.; Schwichtenberg, H. (2003):** Universal Component Trading – Trading in heterogenen Komponentenumgebungen. In: Uhr, W.; Esswein, W.; Schoop, E. (Hrsg.): Wirtschaftsinformatik 2003 – Medien, Märkte, Mobilität, Bd. 1, Heidelberg [u. a.], 2003, S. 303-323.
- Eller, B. (2005):** Klassifikation von Anwendungsgebieten im privaten und mobilen Bereich, Seminararbeit TU Darmstadt, Fachgebiet Wirtschaftsinformatik I, Darmstadt, 2005.
- Elzenheimer, M.; Lonthoff, J.; Ortner, E. (2006):** Requirements and Tools for Reasonable End User Development. In: Hochberger, C.; Liskowsky, R. (Hrsg.): INFORMATIK 2006 – Informatik für Menschen, LNI-Proceedings Bd. P-93 Nr. 1 zur 36. Jahrestagung der GI e.V., Bonn, 2006, S. 580-583.
- Engel, A.; Koschel, A.; Tritsch, R. (2002):** J2EE kompakt: Enterprise Java – Konzepte und Umfeld, Heidelberg [u. a.], 2002.
- Ferstl, O. K.; Sinz, E. J. (2001):** Grundlagen der Wirtschaftsinformatik, Bd. 1, 4. Aufl., München [u. a.], 2001.
- Fettke, P.; Loos, P.; Viehweger, B. (2003):** Komponentenmarktplätze – Bestandsaufnahme und Typologie. In: Turowski, K. (Hrsg.): Proceedings des 5. Workshops Komponentenorientierte betriebliche Anwendungssysteme (WKBA 5), Augsburg, 2003, S. 1-15.

- Fettke, P.; Loos, P.; von der Tann, M. (2002):** Entwicklung eines Repositoriums für Fachkomponenten auf Grundlage des Vorschlages zur Vereinheitlichung der Spezifikation von Fachkomponenten – Analyse von Problemen und Diskussion von Lösungsalternativen. In: Turowski, K. (Hrsg.): Modellierung und Spezifikation von Fachkomponenten, Proceedings der Multikonferenz Wirtschaftsinformatik 2002 (MKWI02), Nürnberg, 2002, S. 87-117.
- Fink, A.; Schneiderei, G.; Voß, S. (2001):** Grundlagen der Wirtschaftsinformatik, Heidelberg, 2001.
- Flake, S. (2003):** Temporal OCL Extensions for Specification of Real-Time Constraints. In: Workshop Specification and Validation of UML models for Real Time and Embedded Systems (SVERTS'03) at UML 2003, San Francisco, 2003.
- Fletcher, P.; Waterhouse, M. (2002):** Web Services Business Strategies and Architectures, Indianapolis, 2002.
- Frank, U. (2006):** Towards a Pluralistic Conception of Research Methods in Information Systems Research, ICB-Research Report No. 7, Universität Duisburg-Essen, 2006.
- Friedell, E. (1960):** Kulturgeschichte der Neuzeit: die Krisis der europäischen Seele von der Schwarzen Pest bis zum Ersten Weltkrieg, Bd. 1 – Einleitung, Renaissance und Reformation, ungek. Ausg. in 1 Bd., 33. - 53 Tsd. Aufl., München, 1960.
- Fritsch, D.; Klinec, D.; Volz, S. (2000):** Nexus – Positioning and Data Management Concepts for Location Aware Applications. In: Proceedings of the 2nd International Symposium on Telegeoprocessing, Nice-Sophia-Antipolis, 2000, S. 171-184.
- Galileo (2003):** The Galilei Project – Galileo Design Consolidation, europa.eu.int/comm/dgs/energy_transport/galileo/doc/galilei_brochure.pdf, Abgerufen am 05.11.06, 2003.
- Gamma, E. (2005):** Design Patterns: Elements of reusable object-oriented Software, 32. Aufl., Boston [u. a.], 2005.
- Garlan, D.; Shaw, M. (1993):** An Introduction to Software Architecture. In: Ambriola, V.; Tortora, G. (Hrsg.): Advances in Software Engineering and Knowledge Engineering, Singapore, 1993, S. 1-39.
- Gates, B. (1999):** Remarks by Bill Gates Fusion '99, <http://www.microsoft.com/bill-gates/speeches/07-23fusion.asp>, Abgerufen am 07.04.06, 1999.
- Gelernter, D. (1985):** Generative Communication in Linda. In: ACM Transactions on Programming Languages and Systems, Bd. 7, Nr. 1, 1985, S. 80-112.

- Geraghty, R. et al. (1999):** COM-CORBA Interoperability, Upper Saddle River, 1999.
- Gillenson, M. L. (1982):** The State of Practice of Data Administration. In: Communications of the ACM, Bd. 25, Nr. 10, 1982, S. 699-706.
- Goedicke, M.; Neumann, G.; Zdun, U. (2001):** Design and Implementation Constructs for the Development of Flexible, Component-Oriented Software Architectures. In: Butler, G. (Hrsg.): LNCS-Proceedings of second international Symposium on Generative and Component based Software Engineering (GCSE 2000), Erfurt, Bd. 2177, Berlin [u. a.], 2001, S. 114-128.
- Graeve, C. (2001):** M-Commerce – Mobilität, Machbarkeit und Manie. In: HMD – Praxis der Wirtschaftsinformatik, Nr. 220, 2001, S. 5-14.
- Griffel, F. (1998):** Componentware: Konzepte und Techniken eines Softwareparadigmas, Heidelberg, 1998.
- Grohmann, W. (2002):** ASP Application Service Providing: Software auf Mietbasis: Kosten sparen. Vorteile nutzen, Köln, 2002.
- Grollius, T. et al. (2005):** An approach for the Management of Service-oriented Architecture (SoA) based Application Systems. In: Desel, J.; Frank, U. (Hrsg.): LNI-Proceedings of the Workshop Enterprise Modelling and Information Systems Architectures (EMISA'05), Klagenfurt, Bd. P-75, Bonn, 2005, S. 208-221.
- Grollius, T.; Lonthoff, J.; Ortner, E. (2006a):** Externes Anwendungsmanagement für ubiquitäre Anwendungssysteme. In: Lehner, F.; Nösekabel, H.; Kleinschmidt, P. (Hrsg.): Tagungsband zur Multikonferenz Wirtschaftsinformatik 2006 (MKWI 2006), Bd. 2, Berlin, 2006, S. 363-373.
- Grollius, T.; Lonthoff, J.; Ortner, E. (2006b):** An Approach to Manage Flexible Component-Based Application Systems. In: Isaias, P.; McPherson, M.; Bannister, F. (Hrsg.): Proceedings of IADIS International Conference e-Society 2006, Bd. 2, Dublin, 2006, S. 219-223.
- Grollius, T. (2004):** Benutzungsplanung und -steuerung komponentenbasierter Anwendungssysteme, Diplomarbeit TU Darmstadt Fachgebiet Wirtschaftsinformatik I, Darmstadt, 2004.
- Gruhn, V.; Thiel, A. (2000):** Komponentenmodelle: DCOM, JavaBeans, Enterprise-JavaBeans, CORBA, München [u. a.], 2000.
- Hanhart, D.; Legner, C.; Österle, H. (2005):** Anwendungsszenarien des Mobile und Ubiquitous Computing in der Instandhaltung. In: Hampe, F. et al. (Hrsg.): Mobile Business – Processes, Platforms, Payments – LNI-Proceedings zur 5. Konferenz Mobile Commerce Technologien und Anwendungen (MCTA 2005), Bd. P-59, Bonn, 2005, S. 45-58.

- Hansen, H. R.; Neumann, G. (2001):** Wirtschaftsinformatik I, 8. Aufl., Stuttgart, 2001.
- Hansen, H. R.; Neumann, G. (2005):** Wirtschaftsinformatik 1: Grundlagen und Anwendungen, 9. Aufl., Stuttgart, 2005.
- Hanser, P. (1995):** Aufbruch in den Cyberspace. In: asw –absatzwirtschaft, Bd. 38, Nr. 8, 1995, S. 34-39.
- Hansmann, U. et al. (2001):** Pervasive Computing Handbook, Berlin [u. a.], 2001.
- Harmon, P. (2000):** What Types of Components are Companies using, <http://www.cutter.com/research/2000/crb001219.html>, Abgerufen am 11.04.06, 2000.
- Hasselbring, W. (2006):** Software-Architektur. In: Informatik Spektrum, Bd. 29, Nr. 1, 2006, S. 48-52.
- Hasselmeyer, P.; Kehr, R.; Voß, M. (2000):** Trade-offs in a Secure Jini Service Architecture. In: Linnhoff-Popien, C.; Hegering, H.-G. (Hrsg.): LNCS-Proceedings of 3rd IFIP/GI International Conference on Trends towards a Universal Service Market (USM 2000), Bd. 1890, Berlin [u. a.], 2000, S. 190-201.
- Hau, M.; Mertens, P. (2002):** Computergestützte Auswahl komponentenbasierter Anwendungssysteme. In: Informatik-Spektrum, Bd. 25, Nr. 5, 2002, S. 331 - 340.
- Hautzinger, H.; Tassaux-Becker, B.; Pfeiffer, M. (1996):** Gesetzmäßigkeiten des Mobilitätsverhaltens – Verkehrsmobilität in Deutschland zu Beginn der 90er Jahre. In: Bundesanstalt für Straßenwesen (Hrsg.): Berichte, Bd. 4, Heft M57, Bergisch-Gladbach, 1996.
- Hayes-Roth, B. et al. (1995):** A Domain-Specific Software Architecture for Adaptive Intelligent Systems. In: IEEE Transactions on Software Engineering, Bd. 21, Nr. 4, 1995, S. 288-301.
- Heinemann, E. (2006):** Sprachlogische Aspekte rekonstruierten Denkens, Redens und Handelns: Aufbau einer Wissenschaftstheorie der Wirtschaftsinformatik, Dissertation TU Darmstadt, Fachgebiet Wirtschaftsinformatik I, Wiesbaden, 2006.
- Heinrich, L. J. (2002):** Wirtschaftsinformatik, Informationsmanagement: Planung, Überwachung und Steuerung der Informationsinfrastruktur, 7. Aufl., München [u. a.], 2002.
- Hellmuth, T. W. (1997):** Terminologiemanagement in der Informationsverarbeitung: Aspekte einer effizienten Kommunikation in der computerunterstützten Informationsverarbeitung, Dissertation Universität Konstanz, Berlin, 1997.

- Henderson-Sellers, B.; Edwards, J. M. (1994):** The working Object: Object-oriented Software Engineering – Methods and Management – Book two of Object-oriented Knowledge, Sydney [u. a.], 1994.
- Hilty, L.; et al. (2003):** Das Vorsorgeprinzip in der Informationsgesellschaft, Auswirkungen des Pervasive Computing auf Gesundheit und Umwelt: Studie des Zentrums für Technologie-Abschätzung, Nr. 46, Bern, 2003.
- Hitz, M. et al. (2005):** UML@Work: objektorientierte Modellierung mit UML2, 3. akt. u. überarb. Aufl., Heidelberg, 2005.
- Holzapfel, H. (2001):** Drang nach Überallität. In: fairkehr, Nr.6, 2001.
- Häckelmann, H.; Petzold, H.-J.; Strahinger, S. (2000):** Kommunikationssysteme: Technik und Anwendungen, Berlin [u. a.], 2000.
- Härder, T.; Rahm, E. (1999):** Datenbanksysteme: Konzepte und Techniken der Implementierung, Berlin [u. a.], 1999.
- Höpfner, H.; Türker, C.; König-Ries, B. (2005):** Mobile Datenbanken und Informationssysteme: Konzepte und Techniken, Heidelberg, 2005.
- Höb, O. (2005):** Ein System für das Wiederverwendungs-Management von Software-Komponenten, Dissertation Universität Stuttgart, Fakultät Maschinenbau, Stuttgart, 2005.
- Hümmer, W. (2004):** Vertragsverhandlungen um konfigurierbare Produkte im elektronischen Handel, Dissertation Friedrich-Alexander-Universität Erlangen-Nürnberg, Fakultät Informatik 6, Erlangen, 2004.
- IBM (o. J.):** Autonomic Computing, <http://www.research.ibm.com/autonomic/overview/>, Abgerufen am 02.10.06, o. J.
- IEEE (1990a):** IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture, New York, 1990.
- IEEE (1990b):** IEEE standard glossary of software engineering terminology, <http://www.ieee.ie11/2238/4148/00159342.pdf?tp=&number=159342&isnumber=4148>, Abgerufen am 05.11.06, 1990.
- IEEE (2000):** IEEE Recommended Practice for Architectural Description of Software-Intensive Systems: IEEE Standard 1471-2000, 2000.
- IEEE (2006a):** IEEE 802.15 WPAN Task Group 1 (TG1), <http://www.ieee802.org/15/pub/TG1.html>, Abgerufen am 08.12.06, 2006.
- IEEE (2006b):** The IEEE 802.16 Working Group on Broadband Wireless Access Standards, <http://grouper.ieee.org/groups/802/16/>, Abgerufen am 13.12.06, 2006.

- IETF; Dierks, T.; Rescorla, E. (2006):** The Transport Layer Security (TLS) Protocol Version 1.1, <http://www.ietf.org/rfc/rfc4346.txt>, Abgerufen am 22.11.06, 2006.
- IETF; Postel, J. B.; Reynolds, J. (1985):** File Transfer Protocol, <http://www.ietf.org/rfc/rfc959>, Abgerufen am 22.11.06, 1985.
- IETF; Postel, J. B. (1982):** Simple Mail Transfer Protocol, <http://www.ietf.org/rfc/rfc0821.txt>, Abgerufen am 22.11.06, 1982.
- IETF; Srinivasan, R. (1995):** RPC: Remote Procedure Call Protocol Specification Version 2, <http://tools.ietf.org/html/rfc1831>, Abgerufen am 22.11.06, 1995.
- ISO (2005):** Common Language Infrastructure, ISO-Dokument Nr. ISO/IEC 23271:2003, Abgerufen am 11.04.06, 2005.
- ISO (1996):** Extended BNF - ISO/IEC 14977:1996, <http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=26153>, Abgerufen am 22.11.06, 1996.
- ISO (1998):** ISO 9241 Ergonomics of human-system interaction - Part 10: Dialogue principles, <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38009&ICS1=13&ICS2=180&ICS3=>, Abgerufen am 21.12.05, 1998.
- ISO (2006):** International Organization for Standardization, <http://www.iso.org>, Abgerufen am 22.11.06, 2006.
- Infrared Data Association (2006):** IrDA, www.irda.org, Abgerufen am 11.12.06, 2006.
- Inmarsat (o. J.):** Total Communications Network, www.inmarsat.org, o. J.
- Jablonski, S. (1997):** Workflow-Management: Entwicklung von Anwendungen und Systemen, Heidelberg, 1997.
- Jablonski, S. (2004):** Guide to web application and platform architectures, Berlin [u. a.], 2004.
- José, D.; Davis, N. (1999):** Scalable and Flexible Location-Based Services for Ubiquitous Information Access. In: LNCS-Proceedings of First International Symposium on Handheld and Ubiquitous Computing (HUC'99), Karlsruhe, Bd. 1707, 1999, S. 52-66.
- Kamlah, W.; Lorenzen, P. (1996):** Logische Propädeutik: Vorschule des vernünftigen Redens, 3. Aufl., Stuttgart, 1996.
- Kehr, R. (2000):** Spontane Vernetzung. In: Informatik Spektrum, Bd. 23, Nr. 3, 2000, S. 161-172.

- Kephart, J. O.; Chess, D. M. (2003):** The Vision of Autonomic Computing. In: IEEE Computer Magazine, Bd. 36, Nr. 1, 2003, S. 41-50.
- Kim, G. (2005):** Designing Virtual Reality Systems: The structured Approach, London [u. a.], 2005.
- Kleinrock, L. (1996):** Nomadicity: anytime, anywhere in a disconnected World. In: Mobile Network Applications, Bd. 1, Nr. 4, 1996, S. 351-357.
- Kollmann, T. (1998):** The Information Triple Jump as the Measure of Success in Electronic Commerce. In: EM - Electronic Markets, Bd. 8, Nr. 4, 1998, S. 44-49.
- Kollmann, T. (2001):** Virtuelle Marktplätze: Grundlagen – Management – Fallstudien, München, 2001.
- Kosiol, E. (1972):** Die Unternehmung als wirtschaftliches Aktionszentrum: Einführung in die Betriebswirtschaftslehre, Reinbeck bei Hamburg, 1972.
- Koster, K. (2002):** Ein Streifzug durch die Welt der mobilen Endgeräte. In: Hartmann, D. (Hrsg.): Business Computing – Geschäftsprozesse mit Mobile Computing: konkrete Projekterfahrung, technische Umsetzung, kalkulierbarer Erfolg des Mobile Business, Braunschweig [u. a.], 2002, S. 127-145.
- Krafzig, D.; Banke, K.; Slama, D. (2005):** Enterprise SOA: Service-Oriented Architecture Best Practices, Upper Saddle River, 2005.
- Krcmar, H. (2000):** Informationsmanagement, 2. Aufl., Berlin [u. a.], 2000.
- Kruchten, P. (2004):** The Rational Unified Process, 2. Aufl., Reading [u. a.], 2004.
- Kruse, W. (2003):** Lebenslanges Lernen in Deutschland – Finanzierung und Innovation: Kompetenzentwicklung, Bildungsnetze, Unterstützungsstrukturen. In: BMBF (Hrsg.): Bericht des BMBF für die OECD zu "Good Practice der Finanzierung Lebenslangen Lernens" im Rahmen des Projektes Co-financing lifelong learning, 2003.
- Kuhn, J. (2003):** Kommerzielle Nutzung mobiler Anwendungen, Dissertation Universität Regensburg, Wirtschaftswissenschaftliche Fakultät, Regensburg, 2003.
- Kuhn, T. S. (1967):** Die Struktur wissenschaftlicher Revolutionen, Frankfurt a. M., 1967.
- Ledoux, T. (1999):** OpenCorba: a Reflective Open Broker. In: LNCS-Proceedings of the Second International Conference Reflection'99, Saint-Malo, Bd. 1616, 1999, S. 197-214.
- Lehmann, F. R. (1998):** Aktuelles Schlagwort: Normsprache. In: Informatik Spektrum, Bd. 21, Nr. 6, 1998, S. 366-367.

- Lehner, F. (1989):** Nutzung und Wartung von Software. Das Anwendungssystem-Management, München, 1989.
- Lehner, F. (2003):** Mobile und drahtlose Informationssysteme: Technologien, Anwendungen, Märkte, Berlin [u. a.], 2003.
- Lehner, F. (1966):** Wechselbeziehungen zwischen Städtebau und Nahverkehr. In: Schriftenreihe für Verkehr und Technik, Nr. 29, 1966.
- Lehner, W. (2002):** Subskriptionssysteme: Marktplatz für omniprésente Informationen, Stuttgart [u. a.], 2002.
- Leinweber, G.; Andresen, T. (1992):** Software- und Anwendungsmanagement: Strategien und Entscheidungshilfen für die Unternehmenspraxis, München [u. a.], 1992.
- Lipp, L. (2004):** Interaktion zwischen Mensch und Computer im Ubiquitous Computing: Alternative Ein- und Ausgabemöglichkeiten für allgegenwärtige Informationstechnologien, Münster, 2004.
- Longshaw, A. (2001):** Choosing Between COM+, EJB, and CCM. In: Heinemann, G. T.; Councill, W. T. (Hrsg.): Component-based Software Engineering, Upper Saddle River, 2001, S. 621-640.
- Lonthoff, J.; Leiber, T. (2005):** Mobile Location Based Gaming als Wegbereiter für Location Based Services. In: von Knop, J.; Haverkamp, W.; Jessen, E. (Hrsg.): "Heute schon das Morgen sehen", LNI-Proceedings zur 19. DFN-Arbeitstagung über Kommunikationsnetze, Düsseldorf, Bd. P-73, 2005, S. 343-360.
- Lonthoff, J.; Ortner, E., Schmidt-Eisenlohr, K. (2006a):** Mobile Anwendungssysteme im privaten und beruflichen Bereich. In: Hochberger, C.; Liskowsky, R. (Hrsg.): INFORMATIK 2006 – Informatik für Menschen, LNI-Proceedings zur 36. Jahrestagung der GI e.V., Bonn, Bd. P-93, Nr. 1, 2006, S. 187-188.
- Lonthoff, J.; Ortner, E. (2006b):** Mobile Location-Based Gaming as Enabler for Location-Based Services (LBS). In: Isaías, P.; McPherson, M.; Bannister, F. (Hrsg.): Proceedings of IADIS International Conference e-Society 2006, Bd. 1, Dublin, Ireland, 2006, S. 485-492.
- Lonthoff, J.; Ortner, E. (2007):** Klassifikations- und Lösungsansätze für Web Services im mobilen Umfeld. In: LNI-Proceedings der 2. Konferenz Mobilität und Mobile Informationssysteme (MMS 2007), Aachen, Bd. P-104, 2007, S. 73-84.

- Lonthoff, J.; Wolf, M., Hoffmann, M. (2006):** Mobile Couponing – Coupon- und Rabattsysteme für Location Based Services. In: Hochberger, C.; Liskowsky, R. (Hrsg.): INFORMATIK 2006 – Informatik für Menschen, LNI-Proceedings zur 36. Jahrestagung der GI e.V., Bonn, Bd. P-93, Nr. 1, 2006, S. 248-254.
- Lonthoff, J. (2004):** Projekt mobiCOMP. In: von Knop, J.; Haverkamp, W.; Jessen, E. (Hrsg.): E-Science und Grid – Ad-hoc-Netze – Medienintegration, LNI-Proceedings zur 18. DFN-Arbeitstagung über Kommunikationsnetze, Bonn, Bd. P-55, 2004, S. 451-465.
- Lonthoff, J. (2007):** Enterprise Mobility. In: T-Systems (Hrsg.): White Paper – Enterprise Mobility: Durch Mobilität zum Erfolg, Frankfurt, 2007.
- Lorenz, K. (1992):** Einführung in die philosophische Anthropologie, Darmstadt, 1992.
- Lorenzen, P. (1973):** Semantisch normierte Orthosprachen. In: Kambartel, F.; Mittelstraß, J. (Hrsg.): Zum normativen Fundament der Wissenschaft, Frankfurt: Athenäum, 1973, S. 231-249.
- Lorenzen, P. (1985):** Grundbegriffe technischer und politischer Kultur, Frankfurt, 1985.
- Malkewitz, R. (2004):** Ambient Intelligence. Stand und Trends, <http://www.zgdv.de/zgdv/zgdv20/Vortraege/STAR-Aml-abstract>, Abgerufen am 30.04.06, 2004.
- Mascolo, C. et al. (2002):** XMIDDLE: A Data-Sharing Middleware for Mobile Computing. In: International Journal on Personal and Wireless Communications, Bd. 1, Nr. 21, 2002, S. 77-103.
- Mattern, F.; Fleisch, E. (2005):** Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis, Berlin [u. a.], 2005.
- Mattern, F. (1999):** Infrastruktur für den elektronischen Markt. In: Praxis der Informationsverarbeitung und Kommunikation, Bd. 22, Nr. 3, 1999, S. 185.
- Mattern, F. (2003):** Total vernetzt: Szenarien einer informatisierten Welt, Berlin [u. a.], 2003.
- McGregor, J. D.; Sykes, D. A. (1992):** Object-Oriented Software Development: Engineering Software for Reuse, New York, 1992.
- Meeks, M. (2001):** Bonobo and Free Software GNOME Components. In: Heinemann, G. T.; Councill, W. T. (Hrsg.): Component-based Software Engineering, Upper Saddle River, 2001, S. 607-619.

- Meffert, H. (2000):** Marketing: Grundlagen marktorientierter Unternehmensführung – Konzepte, Instrumente, Praxisbeispiele, 9. überarb. u. erw. Aufl., Wiesbaden, 2000.
- Mertens, P. (2001):** Lexikon der Wirtschaftsinformatik, 4. vollst. neu bearb. u. erw. Aufl., Berlin [u. a.], 2001.
- Mertens, P. (2005a):** Grundzüge der Wirtschaftsinformatik, 9. Aufl., Berlin [u. a.], 2005.
- Mertens, P. (2005b):** Wirtschaftsinformatik – Aktuelle Probleme eines interdisziplinären Faches, <http://ahr.informatik.uni-kl.de/pubs/Dagstuhl/Mertens.dbag-2005.pdf>, Abgerufen am 01.07.05, 2005.
- Merz, M.; Tu, T.; Lamersdorf, W. (1999):** Electronic Commerce. In: Informatik Spektrum, Bd. 22, Nr. 5, 1999, S. 328-343.
- Merz, M. (2002):** E-commerce und e-business: Marktmodelle, Anwendungen und Technologien, 2. Aufl., Heidelberg, 2002.
- Microsoft (o. J.):** COM: Component Object Model Technologies, <http://www.microsoft.com/com/default.aspx>, Abgerufen am 11.04.06, o. J.
- Microsoft (2006a):** Microsoft Windows Mobile, www.microsoft.com/windows/mobile/, Abgerufen am 08.12.06, 2006.
- Microsoft (2006b):** .NET Compact Framework, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dv_evtuv/html/etconNETCompactFramework.asp, Abgerufen am 08.12.06, 2006.
- Microsoft (2006c):** Managing Mobile Devices in the Enterprise, <http://www.microsoft.com/technet/itsolutions/mobile/evaluate/mblmange.aspx>, Abgerufen am 08.12.06, 2006.
- Microsoft (2006d):** Systems Management Server, <http://www.microsoft.com/smsserver/default.aspx>, Abgerufen am 08.12.06, 2006.
- Microsoft (2006e):** Mobile Information Server, <http://www.microsoft.com/technet/prodtechnol/mis>, Abgerufen am 08.12.06, 2006.
- mik21 (2004):** mik21 Projektvorhaben. Migrationskompetenz als Schlüsselfaktor der Ökonomie des 21. Jahrhunderts, <http://www.mik21.uni-kassel.de/download/mik21-Projektvorhaben.pdf>, Abgerufen am 28.04.06, 2004.
- Mili, H.; Mili, F.; Mili, A. (1995):** Reusing Software – Issues and Research Directions. In: IEEE Transactions on Software Engineering, Bd. 21, Nr. 6, 1995, S. 529-561.
- Mili, H. et al. (2002):** Reuse-Based Software Engineering – Techniques, Organization, and Controls, New York, 2002.

- Mittelstraß, J. (2004):** Enzyklopädie Philosophie und Wissenschaftstheorie, Bd. 2: H-O, Stuttgart [u. a.], 2004.
- Modahl, M. et al. (2006):** UbiqStack: a Taxonomy for a Ubiquitous Computing Software Stack. In: Personal and Ubiquitous Computing, Bd. 10, Nr. 1, 2006, S. 21-27.
- Mutschler, B.; Specht, G. (2004):** Mobile Datenbanksysteme: Architektur, Implementierung, Konzepte, Berlin [u. a.], 2004.
- Müller-Wilken, S. (2002):** Mobile Geräte in verteilten Anwendungsumgebungen: Ein Integrationsansatz zwischen Abstraktion und Migration, Dissertation Universität Hamburg, 2002.
- Nielsen, J. (2003):** Usability engineering, Amsterdam [u. a.], 2003.
- Niemann, F. (2006):** SOA zwischen Mythos und Realität. In: Computerwoche Nr. 5, 2006, S. 6.
- Noam, E. M. (1997):** Systematic Bottlenecks in the Information Society. In: European Communication Council (ECC) - Report 1997, Berlin, 1997, S. 35-44.
- OASIS (2006):** OASIS UDDI Specification TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec, Abgerufen am 22.11.06, 2006.
- OMA (1998):** WMLScript Specification, <http://www.wapforum.org/what/technical/wmlss-30-apr-98.pdf>, Abgerufen am 23.11.06, 1998.
- OMA (2001a):** Wireless Markup Language Version 2.0, <http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/wap-238-wml-20010911-a.pdf>, Abgerufen am 22.11.06, 2001.
- OMA (2001b):** WAP Architecture, <http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/wap-210-waparch-20010712-a.pdf>, Abgerufen am 22.11.06, 2001.
- OMA (2001c):** Wireless Transaction Protocol, <http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/wap-224-wtp-20010710-a.pdf>, Abgerufen am 22.11.06, 2001.
- OMA (2001d):** Wireless Transport Layer Security, <http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/wap-261-wtls-20010406-a.pdf>, Abgerufen am 22.11.06, 2001.
- OMG (1996):** Common Facilities RFP-5 (Meta-Object Facility), OMG TC Document cf/96-02-01, Revision 2, 1996.
- OMG (2004):** Common Object Request Broker Architecture: Core Specification Version 3.0.3, <http://www.omg.org/cgi-bin/doc?formal/04-03-01>, Abgerufen am 22.11.06, 2004.

- OMG (2006):** CORBA Component Model Specification, v4.0, <http://www.omg.org/cgi-bin/doc?formal/06-04-01>, Abgerufen am 11.04.06, 2006.
- OMG (2002):** OMG IDL Syntax and Semantics, <http://www.omg.org/cgi-bin/apps/doc?formal/02-06-39.pdf>, Abgerufen am 04.05.06, 2002.
- OMG (2003):** Response to the UML 2.0 OCL RfP (ad/2000-09-03), <http://www.omg.org/cgi-bin/doc?ad/03-01-07.pdf>, Abgerufen am 06.05.06, 2003.
- OMG (2006):** The Object Management Group (OMG), <http://www.omg.org/>, Abgerufen am 22.11.06, 2006.
- Oestereich, B. (2005):** Analyse und Design mit UML 2: objektorientierte Softwareentwicklung, 7. akt. Aufl., München [u. a.], 2005.
- Opennet-Initiative (2006):** Opennet, wiki.opennet-initiative.de, Abgerufen am 13.12.06, 2006.
- Ortner, E.; Overhage, S. (2003a):** E-NOgS: Ein komponentenorientiertes Middleware-Framework für E-Commerce-Anwendungen. In: Thema Forschung, Nr. 1, 2003, S. 22-27.
- Ortner, E.; Overhage, S. (2003b):** CompoNex: Ein elektronischer Marktplatz für den Handel mit Software-Komponenten über das Internet. In: Thema Forschung, Nr. 1, 2003, S. 28-32.
- Ortner, E.; Söllner, B. (1989):** Semantische Datenmodellierung nach der Objekttypenmethode. In: Informatik Spektrum, Bd. 12, 1989, S. 31-42.
- Ortner, E.; Söllner, B. (1987):** Management betrieblicher Informationssysteme mit Data Dictionary-Unterstützung. In: Strohl-Goebel, H. (Hrsg.): Deutscher Dokumentartag 1987 – Von der Information zum Wissen, vom Wissen zur Information: Traditionelle und moderne Informationssysteme für Wissenschaft und Praxis, Weinheim, 1987, S. 127-150.
- Ortner, E.; Wedekind, H. (2003):** Die Zukunft des Bürgers im Internet. In: Arbeitsbericht Nr. 2, Darmstadt, 2003.
- Ortner, E. (1983):** Rekonstruktion von Begriffen und Begriffsbeziehungen beim Datenbankentwurf. In: Dahlberg, I.; Schader, M. R. (Hrsg.): Studien zur Klassifikation – Automatisierung in der Klassifikation, Bd. 13, Frankfurt, 1983, S. 76-88.
- Ortner, E. (1984a):** Modellierung von Datenbankanwendungen – Datenbankentwurf auf der Ebene von Benutzerfachsprachen. In: HMD – Praxis der Wirtschaftsinformatik, Nr. 118, 1984, S. 37-49.

- Ortner, E. (1984b):** Datenbankentwurf auf der Ebene der Benutzer - dargestellt am Beispiel von Buchungs- und Abrechnungssystemen. In: Schiemenz, B.; Wagner, A. (Hrsg.): Angewandte Wirtschafts- und Sozialkybernetik: neue Ansätze in Praxis und Wissenschaft, Bd. 51, Berlin, 1984, S. 167-181.
- Ortner, E. (1991a):** Informationsmanagement: Wie es entstand, was es ist und wohin es sich entwickelt. In: Informatik Spektrum, Bd. 14, 1991, S. 315-327.
- Ortner, E. (1991b):** Ein Referenzmodell für den Einsatz von Dictionary-/ Repository-Systemen in Unternehmen. In: Wirtschaftsinformatik, Bd. 33, Nr. 5, 1991, S. 420-430.
- Ortner, E. (1993):** Software-Engineering als Sprachkritik: die sprachkritische Methode des fachlichen Software-Entwurfs, Konstanz, 1993.
- Ortner, E. (1999):** Repository Systems. Teil 1: Mehrstufigkeit und Entwicklungsumgebung. In: Informatik-Spektrum, Bd. 22, Nr. 4, 1999, S. 235 - 251.
- Ortner, E. (2002):** Sprachingenieurwesen: Empfehlung zur inhaltlichen Weiterentwicklung der (Wirtschafts-)Informatik. In: Informatik Spektrum, Bd. 25, Nr. 1, 2002, S. 39-51.
- Ortner, E. (2003):** Vom Wort zum Bauelement. In: Thema Forschung, Nr. 1, 2003.
- Ortner, E. (2004):** Anthropozentrik und Sprachbasierung in der (Wirtschafts-)informatik. In: Hammwöhner, R.; Rittberger, M.; Semar, W. (Hrsg.): Wissen in Aktion – Der Primat der Pragmatik als Motto der Konstanzer Informationswissenschaft, Bd. 41, Konstanz, 2004, S. 141-152.
- Ortner, E. (2005a):** Sprachbasierte Informatik: Wie man mit Wörtern die Cyber-Welt bewegt, Leipzig, 2005.
- Ortner, E. (2005b):** Forschungsprogramm 2005, Darmstadt, 2005.
- Ortner, E. (2006):** Entwicklung von Anwendungssystemen II, Vorlesungsfolien, Darmstadt, 2006.
- Overhage, S. (2002):** Die Spezifikation – kritischer Erfolgsfaktor der Komponentenorientierung. In: Turowski, K. (Hrsg.): Proceedings des 4. Workshops Komponentenorientierte betriebliche Anwendungssysteme (WKBA 4), Augsburg, 2002, S. 1-18.
- Overhage, S. (2006):** Vereinheitlichte Spezifikation von Komponenten: Grundlagen, UnSCom Spezifikationsrahmen und Anwendung, Dissertation Universität Augsburg, Fakultät Wirtschaftsinformatik und Systems Engineering, Augsburg, 2006.
- Pagé P. (1992):** Strategien in der Software-Produktion. In: Leinweber, G. (Hrsg.): Software- und Anwendungsmanagement, München, 1992, S. 43-104.

- Pandya, R. (2000):** Mobile and personal Communication Systems and Services, Piscataway, 2000.
- Parnas, D. L. (1972):** On the Criteria to be used in Decomposing Systems into Modules. In: Communications of the ACM, Bd. 15, Nr. 12, 1972, S. 1053-1058.
- Patrick, P. (2005):** Impact of SOA on Enterprise Information Architectures. In: Proceedings of the 2005 ACM SIGMOD international Conference on Management of Data, New York, 2005, S. 844-848.
- Perry, D. E.; Wolf, A. L. (1992):** Foundations for the Study of Software Architecture. In: ACM Sigsoft Software Engineering Notes, Bd. 17, Nr. 4, 1992, S. 40-52.
- Picco, G. P.; Murphy, A. L.; Roman, G.-C. (1999):** LIME: Linda meets mobility. In: IEEE Computer Society Press (Hrsg.): Proceedings of the 21st international conference on software engineering (ICSE'99), Los Angeles, 1999, S. 368-377.
- Preuß, S. (2000):** NetObjects – Dynamische Proxy-Architektur für Jini. In: Proceedings of Net.ObjectDays 2000, Erfurt, 2000, S. 146-155.
- Prieto-Diaz, R. (1993):** Status Report: Software Reusability. In: IEEE Software, Bd. 10, Nr. 3, 1993, S. 61-66.
- Ramakrishnan, S.; McGregor, J. (1999):** Extending OCL to Support Temporal Operators. In: Proceedings of 21st International Conference on Software Engineering (ICSE99), Workshop on Testing Distributed Component-Based Systems, L.A., 1999.
- Ravichandran, T.; Rothenberger, M. (2003):** Software Reuse Strategies and Component Markets. In: Communications of the ACM, Bd. 46, Nr. 8, 2003, S. 109-114.
- Reiche, D. (2003):** Roche-Lexikon Medizin, 5. Aufl., München [u. a.], 2003.
- Repp, N. (2003):** mobileISM – Entwicklung eines Frameworks zur benutzerunterstützten Verwaltung von Diensten auf mobilen Endgeräten. In: Wirtschaftsinformatik, Bd. 45, Nr. 6, 2003, S. 658-661.
- Reussner, R.; Hasselbring, W. (2006):** Handbuch der Software-Architektur, Heidelberg, 2006.
- Ridder, L. (2002):** Mereologie: ein Beitrag zur Ontologie und Erkenntnistheorie. In: Philosophische Abhandlungen, Bd. 83, Frankfurt, 2002.
- Rischpater, R. (2000):** WAP und WML: Wireless Web – Das neue Internet, Bonn, 2000.

- Rorty, R. M. (1967):** The linguistic Turn: Essays in Philosophical Method, Chicago [u. a.], 1967.
- Rotach, M.; Hoppler, F.; Mötteli, M (1986):** Erarbeitung eines Forschungsprogrammes Mobilität. Forschungsauftrag ETH Zürich, Institut für Verkehrsplanung, Transporttechnik, Straßen- und Eisenbahnbau, Zürich, 1986.
- Roth, J. (2002):** Mobile Computing: Grundlagen, Technik, Konzepte, Heidelberg, 2002.
- Roth, J. (2005):** Mobile Computing: Grundlagen, Technik, Konzepte, 2. akt. Aufl., Heidelberg, 2005.
- Rothermel, K. (2005):** SFB 627: Nexus Umgebungsmodelle für Mobile Kontextbezogene Systeme, <http://www.nexus.uni-stuttgart.de/>, Abgerufen am 22.11.06, 2005.
- Royce, W. (1970):** Managing the Development of Large Software Systems: Concepts and Techniques. In: Proceedings of IEEE WESCON, Washington, 1970, S. 1-9.
- Röttger-G., S (2002):** Lokalisierungsmethoden. In: Gora, W.; Röttger-Gerigk, S. (Hrsg.): Handbuch Mobile-Commerce: technische Grundlagen, Marktchancen und Einsatzmöglichkeiten, Berlin [u. a.], 2002, S. 419-426.
- Rürup, B. (1998):** Wohlfahrtsstaatliche Politik in der globalisierten Informationsgesellschaft. In: Schriften der Friedrich-Ebert-Stiftung, Bonn, 1998.
- Sametinger, J. (1997):** Software Engineering with reusable Components, Berlin [u. a.], 1997.
- Sang-Bum Park, A. (2004):** A Service-Based Agent System Supporting Mobile Computing, Dissertation RWTH Aachen, Aachen, 2004.
- Sarkar, M. B.; Butler, B.; Steinfield, C. (1995):** Intermediaries and Cybermediaries: A Continuing Role for Mediating Players in the Electronic Marketplace. In: Journal of Computer-Mediated Communication, Bd. 1, Nr. 3, 1995.
- Satyanarayanan, M. et al. (1990):** Coda: a highly available file system for a distributed workstation environment. In: IEEE Transactions on Computers, Bd. 39, Nr. 4, 1990, S. 447-459.
- Scheidl, S. (2006):** Workshop End-User Development. In: Hochberger, C.; Liskowsky, R. (Hrsg.): INFORMATIK 2006 – Informatik für Menschen, Bd. 1, Bonn, 2006, S. 563.
- Schiemann, H. (1992):** Glossar. In: Leinweber, G. (Hrsg.): Software- und Anwendungsmanagement, München, 1992, S. 253-265.

- Schienmann, B. (1997):** Objektorientierter Fachentwurf, Stuttgart, 1997.
- Schiller, J. (2000):** Mobilkommunikation: Techniken für das allgegenwärtige Internet, London [u. a.], 2000.
- Schiller, J. (2003):** Mobilkommunikation, 2. überarb. Aufl., München [u. a.], 2003.
- Schmid, B. (1993):** Elektronische Märkte. In: Wirtschaftsinformatik, Bd. 35, Nr. 5, 1993, S. 465-480.
- Schneider, H. J. (1992):** Phantasie und Kalkül: über die Polarität von Handlung und Struktur in der Sprache, Frankfurt, 1992.
- Schulze, W. (1999):** Ein Workflow-Management-Dienst für ein verteiltes Objektverwaltungssystem, Dissertation TU Dresden, Fakultät Informatik, Dresden, 1999.
- Schwanke, R. W.; Altucher, R. Z.; Platoff, M. A. (1989):** Discovering, Visualizing, and Controlling Software Structure. In: Proceedings of 5th International Workshop on Software Specification and Design IWSSD, Pittsburgh, 1989, S. 147-150.
- Schwichtenberg, H. (2003):** Universal Component Trading, Dissertation Universität Duisburg-Essen, Fakultät Wirtschaftswissenschaften, Essen, 2003.
- Senko, M. E. et al. (1972):** Specifications in a Data Independent Accessing Model. In: Proceedings of 1972 ACM-SIGFIDET workshop on Data description, access and control, New York, 1972, S. 363-382.
- Shannon, C. E.; Weaver, W. (1949):** The Mathematical Theory of Communication, Urbana, 1949.
- Shaw, M. (1989):** Larger Scale Systems Require Higher-Level Abstractions. In: Proceedings of 5th International Workshop on Software Specification and Design IWSSD, Pittsburgh, 1989, S. 143-146.
- Siedersleben, J. (2003):** Softwaretechnik, 2. überarb. u. akt. Aufl., München [u. a.], 2003.
- SEI (2006):** How Do You Define Software Architecture? Software Engineering Institute (SEI), Carnegie Mellon University, www.sei.cmu.edu/architecture/definitions.html, Abgerufen am 27.02.06, 2006.
- Stahlknecht, P.; Hasenkamp, U. (2005):** Einführung in die Wirtschaftsinformatik, 11. Aufl., Berlin [u. a.], 2005.
- Stevens, P.; Pooley, R. J. (2000):** Using UML: Software Engineering with Objects and Components, akt. Aufl., Harlow [u. a.], 2000.

- Stowasser, A. C.; Petschenig, J. M.; Skutsch, M. F. (1994):** Lateinisch-deutsches Schulwörterbuch, Wien, 1994.
- Sun (o. J.):** Java Platform, Enterprise Edition (Java EE), <http://java.sun.com/javae/index.jsp>, Abgerufen am 11.04.06, o. J.
- Sun (o. J.):** Java 2 Platform, Micro Edition, <http://java.sun.com/j2me/j2me-ds.pdf>, Abgerufen am 22.11.06, o. J.
- Sun (2006a):** Jini Network Technology, sun.com/jini/, Abgerufen am 22.11.06, 2006.
- Sun (2006b):** Remote Method Invocation Home, <http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>, Abgerufen am 22.11.06, 2006.
- Sun (2006c):** The Java ME Platform, <http://java.sun.com/javame/index.jsp>, Abgerufen am 22.11.06, 2006.
- Sun (2006d):** Mobile Information Device Profile (MIDP), <http://java.sun.com/products/midp/>, Abgerufen am 22.11.06, 2006.
- Sun (2006e):** Connected Limited Device Configuration (CLDC), <http://java.sun.com/products/cldc/>, Abgerufen am 22.11.06, 2006.
- Szyperski, C.; Pfister, C. (1997):** Workshop on Component-Oriented Programming Summary. In: Mühlhäuser, M. (Hrsg.): Special issues in Object Oriented Programming – Workshop Reader of the 10th European Conference on Object Oriented Programming ECOOP '96, Linz, 1997, S. 127-130.
- Szyperski, C.; Gruntz, D. W.; Murer, S. (2002):** Component software: beyond object-oriented programming, 2. Aufl., London [u. a.], 2002.
- Szyperski, C. (1998):** Component software: beyond object-oriented programming, Harlow [u. a.], 1998.
- Tandler, P. (2004):** Synchronous Collaboration in Ubiquitous Computing Environments, Dissertation TU Darmstadt, Fakultät Informatik, Darmstadt, 2004.
- Tanenbaum, A. S. (1998):** Computernetzwerke, 3. Aufl., München [u. a.], 1998.
- Tanenbaum, A. S. (2003):** Computernetzwerke, 4. überarb. Aufl., München [u. a.], 2003.
- Tarski, A. (1979):** Der Wahrheitsbegriff in den formalisierten Sprachen. In: Berka, K.; Kreiger, L. (Hrsg.): Logik Texte, Berlin, 1979.
- Taubner, D. (2005):** Software-Industrialisierung. In: Informatik Spektrum, Bd. 28, Nr. 4, 2005, S. 292-296.

- Terry, D. et al. (1995):** Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. In: Proceedings of the 15th ACM Symposium on Operating Systems Principles, 1995, S. 172-183.
- Tielert, R. (2004):** Forschungsschwerpunkt Ambient Intelligence. Szenario: Assisted Living, TU Kaiserslautern, http://www.eit.uni-kl.de/Aml/de/inhalte/assisted_living/_pics/Scenario_assisted_living.pdf, Abgerufen am 30.04.06, 2004.
- Traas, V.; van Hillegersberg, J. (2000):** The software component market on the internet current status and conditions for growth. In: SIGSOFT Software Engineering Notes, Bd. 25, Nr. 1, 2000, S. 114-117.
- Tsichritzis, D. C.; Lug, A. (1978):** The ANSI/X3/Sparc DBMS Framework - Report of the Study Group on Database Management Systems. In: Information systems, Bd. 3, Nr. 3, 1978, S. 173-191.
- Turowski, K.; Conrad, S. (2001):** Temporal OCL: Meeting Specifications Demands for Business Components. In: Siau, K.; Halpin, T. (Hrsg.): Unified Modeling Language: Systems Analysis, Design and Development Issues, Hershey, 2001, S. 151-165.
- Turowski, K.; Pousttchi, K. (2004):** Mobile Commerce: Grundlagen und Techniken, Berlin [u. a.], 2004.
- Ullmer, B.; Ishii, H. (2000):** Emerging Frameworks for Tangible unser Interfaces. In: IBM Systems Journal, Bd. 39, Nr. 3&4, 2000, 915-931.
- Ulrich, P.; Fluri, E. (1995):** Management: eine konzentrierte Einführung, 7. verb. Aufl., Bern [u. a.], 1995.
- van Renesse, R.; Birman, K. P.; Vogels, W. (2003):** Astrolabe: A robust and scalable Technology for distributed System Monitoring, Management, and Data Mining. In: CM Transactions on Computer Systems, Bd. 21, Nr. 2, 2003, S. 164-206.
- Versteegen, G.; Salomon, K.; Heinold, R. (2001):** Change-Management bei Softwareprojekten, Berlin [u. a.], 2001.
- Versteegen, G. (2003):** Konfigurationsmanagement, Berlin [u. a.], 2003.
- Voas, J. (1998):** COTS Software: The Economical Choice? In: IEEE Software, Bd. 15, Nr. 2, 1998, S. 16-19.
- Vogler, M. (1994):** OTMAR: Ein automatischer Übersetzer zur Konstruktion von Objekttypen und Beziehungen aus Aussagen, Diplomarbeit Universität Konstanz, Fakultät Informationsmanagement, Konstanz, 1994.

- Vogt, W. (2002):** Auch Mobilität hat ihre Grenzen. In: Der Bürger im Staat – Mobilität, Nr. 3, 2002.
- Voigtmann, P.; Zeller, T. (2003):** Beiträge zur Integrationsproblematik im Kontext von Electronic Business und Elektronischen Marktplätzen. In: Uhr, W.; Esswein, W.; Schoop, E. (Hrsg.): Wirtschaftsinformatik 2003 – Medien, Märkte, Mobilität, Bd. 1, Heidelberg [u. a.], 2003, S. 215-237.
- W3C (1999):** XML Path Language (XPath) Version 1.0, <http://www.w3.org/TR/xpath>, Abgerufen am 22.11.06, 1999.
- W3C (2001):** Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>, Abgerufen am 22.11.06, 2001.
- W3C (2004):** Simple Object Access Protocol (SOAP) Specifications, <http://www.w3.org/TR/soap/>, Abgerufen am 22.11.06, 2004.
- W3C (2005):** HTTP - Hypertext Transfer Protocol, <http://www.w3.org/Protocols/>, Abgerufen am 08.12.06, 2005.
- W3C (2006a):** Extensible Markup Language (XML), <http://www.w3.org/XML/>, Abgerufen am 22.11.06, 2006.
- W3C (2006b):** HyperText Markup Language (HTML) Home Page, <http://www.w3.org/MarkUp/>, Abgerufen am 22.11.06, 2006.
- W3C (2006c):** World Wide Web Consortium, <http://www.w3.org/>, Abgerufen am 22.11.06, 2006.
- W3C (2006d):** XML Schema, <http://www.w3.org/XML/Schema>, Abgerufen am 22.11.06, 2006.
- W3C (2006e):** The Extensible Stylesheet Language Family (XSL), <http://www.w3.org/Style/XSL/>, Abgerufen am 22.11.06, 2006.
- WAP Forum (o. J.):** WAP 2.0 Technical White Paper, <http://www.openmobile-alliance.org/tech/affiliates/wap/wapindex.html>, Abgerufen am 08.10.06, o. J.
- Waldo, J. (1998):** Javaspace specification 1.0, <http://www-csag.ucsd.edu/teaching/cse225s04/Reading%20List/microsystems98jvaspaces.pdf>, Abgerufen am 08.10.06, 1998.
- Wang, N.; Schmidt, D. C.; O’Ryan, C. (2001):** Overview of the CORBA Component Model. In: Heinemann, G. T.; Councill, W. T. (Hrsg.): Component-based Software Engineering, Upper Saddle River, 2001, S. 557-571.
- Wedekind, H. et al. (1998):** Modellierung, Simulation, Visualisierung: Zu aktuellen Aufgaben der Informatik. In: Informatik-Spektrum, Bd. 21, Nr. 5, 1998, S. 265-272.

- Wedekind, H.; Ortner, E.; Inhetveen, R. (2004):** Informatik als Grundbildung. In: Informatik Spektrum, Bd. 27, Nr. 4, 2004, S. 337-342.
- Wedekind, H.; Ortner, E. (1980):** Systematisches Konstruieren von Datenbank-anwendungen: zur Methodologie der angewandten Informatik, München [u. a.], 1980.
- Wedekind, H. (1973):** Systemanalyse: die Entwicklung von Anwendungssystemen für Datenverarbeitungsanlagen, München, 1973.
- Wedekind, H. (1999):** Von der Informatik zum Computational Science and Engineering. In: Walther, A. (Hrsg.): Pionier des wissenschaftlichen Rechnens, TUD Schriftenreihe Wissenschaft und Technik, Bd. 75, Darmstadt, 1999.
- Wedekind, H. (2005):** Die Arbeitsteilung ist es! In: Informatik Spektrum, Bd. 28, Nr. 5, 2005, S. 440-442.
- Weinreich, R.; Sametinger, J. (2001):** Component Models and Component Services – Concepts and Principles. In: Heinemann, G. T.; Councill, W. T. (Hrsg.): Component-based Software Engineering, Upper Saddle River, 2001, S. 33-48.
- Weiser, M. (1991):** The Computer of the 21st Century. In: Scientific American, Bd. 91, Nr. 9, 1991, S. 94-100.
- Weiser, M. (1993):** Some Computer Science Issues in Ubiquitous Computing. In: Communications of the ACM, Bd. 36, Nr. 6, 1993, S. 75-84.
- Westhoff, J. (2000):** IDW-Informationdienst Wissenschaft, Pressegespräch anlässlich des Kongresses Medizin und Mobilität <http://idw-online.de/pages/de/news?print=1&id=24482>, Abgerufen am 19.03.06, 2000.
- Westphal, R. (2002):** .Net kompakt, Heidelberg [u. a.], 2002.
- Wiecker, M. (2002):** Breitbandige, kabellose Übertragungstechnologien. In: Gora, W.; Röttger-Gerigk, S. (Hrsg.): Handbuch Mobile-Commerce: technische Grundlagen, Marktchancen und Einsatzmöglichkeiten, Berlin [u. a.], 2002, S. 427-440.
- Wiedenmaier, S. J. (2004):** Unterstützung manueller Montage durch Augmented-reality-Technologien, Aachen, 2004.
- Wild, J. (1971):** Management-Konzeption und Unternehmungsverfassung. In: Schmidt, R.-B. (Hrsg.): Probleme der Unternehmungsverfassung, Tübingen, 1971, S. 57-95.
- Wild, M.; Herges, S. (2000):** Total Cost of Ownership (TCO) – Ein Überblick. In: Arbeitspapiere WI, Nr. 1, Mainz, 2000.

- Wilkes, M. V. (1970):** Time-sharing Computer Systems, Leipzig, 1970.
- Wille, R.; Ganter, B. (1996):** Formale Begriffsanalyse: mathematische Grundlagen, Berlin [u. a.], 1996.
- Williams, J. D. (1996):** Managing Iteration in OO Projects. In: Computer, Bd. 29, Nr. 9, 1996, S. 39-43.
- Wilmsen, E. N. (1972):** Social Exchange and Interaction, Ann Arbor, 1972.
- Wollnik, M. (1988):** Ein Referenzmodell des Informationsmanagements. In: Information Management, Bd. 3, Nr. 3, 1988, S. 34-43.
- Wyckoff, P. et al. (1998):** T spaces. In: IBM (Hrsg.): IBM Systems Journal, Bd. 37, Nr. 3, 1998, S. 454-474.
- Zeidler, A.; Gruteser, M. (1999):** Bezaubernde Geräte. In: iX – Magazin für professionelle Informationstechnik, Nr. 4, 1999, S. 144.
- Zimmerman, J. B. (1999):** Mobile Computing: Characteristics, Business Benefits, and the Mobile Framework, Bowie State, 1999.
- Zimmermann, H.-D. et al. (2006):** Electronic Markets – Special Issue Electronic Negotiation. In: Electronic Markets - The International Journal, Bd. 16, Nr. 3, 2006.
- Zobel, J. (2001):** Mobile-Business und M-Commerce: die Märkte der Zukunft erobern, München [u. a.], 2001.
- Zwahr, A. (2000):** Brockhaus, Sonderausgabe 2000, Leipzig [u. a.], 2000.
- Zwicky, F. (1966):** Entdecken, erfinden, forschen im morphologischen Weltbild, München [u. a.], 1966.

A Anhang – Systementwurf von mobiCOMP

Der grobe Aufbau des mobiCOMP-Systems wurde im Abschnitt 5.1 mit der Beschreibung der mobiCOMP-Architektur gezeigt. In Abbildung 58 wurden die einzelnen Marktplatzkomponenten veranschaulicht. Zwischen den Marktplatzkomponenten bestehen folgende Schnittstellen im System:

- Darstellungssystem (GUI) mit dem Vorgangsteuerungssystem
- Vorgangsteuerungssystem mit dem Kommunikationssystem
- Kommunikationssystem mit dem Dateitransfersystem
- Kommunikationssystem mit dem DB-Wrapper
- DB-Wrapper mit dem Metainformationssystem

Im Folgenden wird das methodenspezifische Systemkonzept wiedergegeben, das die Grundlage für die Implementierung von mobiCOMP bildet.

A.1 Das Darstellungssystem

A.1.1 Interne Struktur des Darstellungssystems

Das Darstellungssystem setzt sich im Wesentlichen aus Formularen, Programm-Skripten und die Schnittstelle zum VSS zusammen.

Aktionen des Benutzers werden an das VSS gesendet. Dieses verarbeitet die Aktion und sendet eine Formularidentifikationsnummer zurück. Daraufhin wird vom DSS das anzuzeigende Formular gerätespezifisch generiert. Zur Generierung des Formulars werden bestimmte Metadaten benötigt, welche die Struktur der anzuzeigenden Formularelemente beschreiben. Diese Daten werden über die VSS-Schnittstelle abgefragt und von einem zentralen Skript verarbeitet. Dieses Skript verteilt die Daten an formularspezifische Skripte, welche daraus den WML bzw. HTML Code für die Darstellung generieren. Nach Eingabe der Daten sendet der Benutzer das Formular ab. Vor der Weiterleitung der Eingaben an das VSS werden die Formulare auf syntaktischer Ebene auf Zulässigkeit überprüft. Dazu gehört die Überprüfung, ob

- alle Muss-Felder ausgefüllt wurden,
- Textfelder und Textbereiche die vorgeschriebene Länge einhalten,
- Datumsangaben im vorgesehenen Bereich liegen,
- E-Mail Adressen eine gültige Struktur haben und
- ob Zahlen im vorgeschrieben Bereich liegen.

Sollte mindestens eine der Eingaben ungültig sein, wird das Formular mit einem entsprechenden Hinweis erneut angezeigt. Sobald alle Eingaben korrekt vorliegen, werden diese an das VSS weitergeleitet.

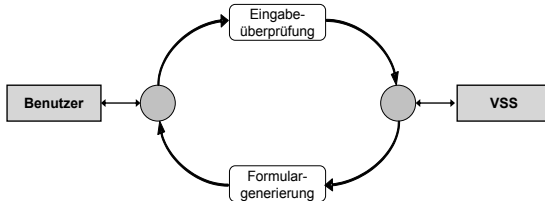


Abbildung 60: Interaktion⁶²² Benutzer – Vorgangsteuerungssystem

Formulardefinitionen

Formular	Willkommenseite	
Beschreibung	Willkommenseite für den mobiCOMP Marktplatz	
Felder	keine	

Formular	Authentifikation	
Beschreibung	Formular zur Eingabe der Authentifikationsdaten des Benutzers	
Felder	Benutzername	Textfeld
	Passwort	Textfeld

Formular	Komponente suchen	
Beschreibung	Ermöglicht die Suche nach Komponenten basierend auf verschiedenen Kriterien	
Felder	Komponentenname	Textfeld
	Preis	Textfeld
	Preis-Bedingung	Liste
	Währung	Liste
	Betriebssystem	Liste
	Hersteller	Textfeld

⁶²² Zur Notation von Interaktionsdiagrammen vgl. Oestereich 2005, S. 326.

Formular	Komponente stöbern	
Beschreibung	Zeigt die für die Komponenten vorgesehenen Kategorien zum Stöbern nach Komponenten an	
Felder	<i>Verweise für Kategorien</i>	

Formular	Komponentendetails anzeigen	
Beschreibung	Zeigt erweiterte Details zu einer Komponente an	
Felder	Komponentenname	Textfeld
	Preis	Textfeld
	Währung	Textfeld
	Beschreibung	Textfeld

Formular	Komponente kaufen	
Beschreibung	Fragt den Benutzer nach einer Bestätigung vor dem Kaufen einer Komponente	
Felder	<i>Bestätigungsseite</i>	

Formular	Komponente hinzufügen	
Beschreibung	Erlaubt dem Benutzer das Hinzufügen einer Komponente auf dem Marktplatz	
Felder	Komponentenname	Textfeld
	Beschreibung	Textfeld
	Preis	Textfeld
	Betriebssystem	Liste
	Währung	Liste

Formular	Komponente ändern	
Beschreibung	Erlaubt einem Benutzer die Daten zu einer Komponente zu ändern	
Felder	<i>siehe Komponente hinzufügen</i>	

Formular	Registrierung	
Beschreibung	Formular zur Eingabe der Registrierungsseite eines neuen Benutzers und zum Ändern der Registrierungsdaten eines vorhandenen Benutzers	
Felder	Vorname	Textfeld
	Nachname	Textfeld
	Telefonnummer	Textfeld
	Faxnummer	Textfeld
	Postleitzahl	Textfeld
	Wohnort	Textfeld
	Kreditkartennummer	Textfeld
	Kreditkartengültigkeit	Textfeld
	Kreditkartengesellschaft	Liste
	E-Mail-Adresse	Textfeld

Formular	Statistik anzeigen	
Beschreibung	Zeigt detaillierte Statistiken zu einer Komponente an	
Felder	Komponentenname	Textfeld
	Anzahl Zugriffe auf Detailseite	Textfeld
	Anzahl Käufe	Textfeld
	Anzahl Downloads	Textfeld
	Erstellungsdatum	Textfeld
	Letzte Änderung	Textfeld
	Anzahl Änderungen insgesamt	Textfeld

Formular	Verabschiedung	
Beschreibung	Verabschiedungsseite für den mobiCOMP Marktplatz	
Felder	<i>keine</i>	

Metadaten

Im Folgenden werden die Datentypen für die Metadaten spezifiziert.

FormData	
FormObject] formObject	Die Formularfelder werden durch das 2D-Array gruppiert, wobei die erste Dimension des Arrays die Gruppe und die zweite Dimension Elemente in dieser Gruppe bezeichnet.

FormObject	
String title	Titel des Feldes
String data	aktuelle Daten
Boolean mandatory	Pflichtfeld?

TextField::FormObject	
TextFieldType type	Definiert Typ des Textfeldes

TFTNumeric::TextFieldType	
int min	kleinste gültige Zahl
int max	größte gültige Zahl

TFTAlphanumeric::TextFieldType	
int maxLength	maximale Anzahl an akzeptierten Zeichen
TFTEmail::TextFieldType	

TFDate::TextFieldType	
date from	kleinstes akzeptiertes Datum
date to	größtes akzeptiertes Datum

TextArea::FormObject	
int maxLength	Anzahl akzeptierter Zeichen

List::FormObject	
Boolean multiple	Mehrfachauswahl möglich
ListEntry listEntry	Liste mit Einträgen

ListEntry	
int ID	Identifikation für Listenauswahl
String label	Darstellung (Text) der Listenauswahl

Klassendiagramme der Metadaten

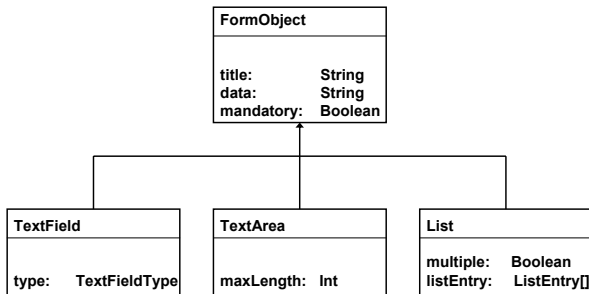


Abbildung 61: Klassendiagramm⁶²³ Formular Objekt

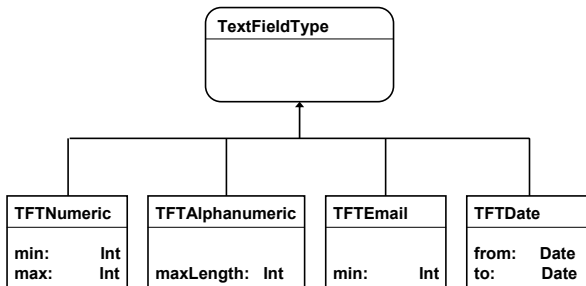


Abbildung 62: Klassendiagramm Textfeldtyp

⁶²³ Zur Notation von Klassendiagrammen vgl. Hitz et al. 2005, S. 52ff.

A.2 Die Objekte des mobiCOMP Marktplatzsystems

User Data

Attribute	Typ	Beschreibung
Company	String	Firmenname
Prename	String	Vorname
Name	String	Name
Phone	Int	Telefonnummer
Fax	Int	Optional; Faxnummer
Plz	Int	Postleitzahl
City	String	Stadt
creditCardNumber	Int	Optional; Kreditkartennummer
bankName	String	Optional; Bankname
Blz	Int	Optional; Bankleitzahl
Account	Int	Optional; Kontonummer
Payment	Choice	Auswahl der Zahlungsart
Password	String	Passwort
sessionID	Session	Siehe Objekt Session
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • Registration • ChangeUserData

Authentication

Attribute	Typ	Beschreibung
userID	String	Siehe Objekt Session
Password	String	Passwort
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • Login • RemoveUser

Session

Attribute	Typ	Beschreibung
sessionID	String	Die eindeutige Identifikation der Session
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • CreateSession • UpdateSession
UserID	String	Die eindeutige Identifikation des Benutzers

WebPage

Attribute	Typ	Beschreibung
webPageID	String	Eine eindeutige Identifikation einer Webseite
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • ShowWebPage

Component

Attribute	Typ	Beschreibung
componentName	String	Dem Benutzer angezeigter Komponentename
Price	Double	Preis der Komponente
OS	String	Benötigtes Betriebssystem
Description	String	Beschreibung der Komponente
smallDescription	String	Kurze Beschreibung
requiredFeatures	String	Funktionale Voraussetzungen
Version	String	Version
Currency	Int	Währung
producerName	String	Herstellernamen der Komponente
componentID	String	Eindeutige Identifikation der Komponente
sessionID	Session	Siehe Objekt Session
Component	Binary	Optional; Die Komponente als Binary
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • AddComponent • ChangeComponent

DownloadComponent

Attribute	Typ	Beschreibung
sessionID	Session	Siehe Objekt Session
componentID	String	Eindeutige Identifikation der Komponente
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • DownloadComponent
Component	Binary	Die Komponente als Binary

Category

Attribute	Typ	Beschreibung
categoryID	String	Die eindeutige Identifikationsnummer der Kategorie
categoryName	String	Dem Benutzer angezeigter Kategorienamen
Description	String	Beschreibung dieser Kategorie; vom Benutzer zu lesen
subCategories	Array SubCategory	Die Beschreibung der Unterkategorien
components	Array Component	Optional
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • BrowseComponent
sessionID	Session	Siehe Objekt Session

SubCategory

Attribute	Typ	Beschreibung
categoryName	String	Der dem Benutzer angezeigte Name einer Kategorie
categoryID	String	Die eindeutige Identifikationsnummer der Kategorie

SearchComponent

Attribute	Typ	Beschreibung
componentName	String	Nach diesen Namensfragmenten soll gesucht werden
Price	Double	Preis der Komponente; steht in Verbindung mit priceLimit
OS	String	Gewünschtes Betriebssystem der Komponente
description	String	Suche in der Beschreibung der Komponente
priceLimit	String	Wie soll der Preis sein: less, greater, equal
producerName	String	Sucher nach Herstellernamen
componentID	String	Wenn man die ID kennt wird nach ihr gesucht
sessionID	Session	Siehe Session
actionID	String	Kann folgende ActionID haben: <ul style="list-style-type: none"> • SearchComponent

A.3 Schnittstelle Darstellungssystem ↔ Vorgangsteuerungssystem

Für die Schnittstelle DSS/VSS werden die Methoden performAction und getForm bereitgestellt.

A.3.1 Methode performAction

Spezifikation	performAction (ActionData data, ActionID actionID, UserID userID)
Parameter	<i>data</i> : Daten, welche die Aktion betreffen <i>actionID</i> : Eindeutiger Bezeichner für durchzuführende Aktion <i>userID</i> : ID des Benutzers, für den die Aktion durchgeführt werden soll
Rückgabe	<i>FormID</i> , zu dem anschließend verzweigt wird; hierdurch kann auch zu einer Seite zur Anzeige von Fehlermeldungen verzweigt werden

Über die Methode performAction leitet die GUI die Aktionen des Anwenders an das VSS weiter. Durch den mitgelieferten Parameter ActionData können optional an einen Aufruf Eingabedaten angehängt werden. Bei den Eingabedaten handelt es sich um Daten, die der Anwender dem System über die grafische Oberfläche (GUI) zur Weiterverarbeitung zur Verfügung stellt. Diese Daten werden an die nachfolgende Ebene weitergegeben.

Neben der ActionData erhält das VSS über die Parameter ActionID und UserID weitere Informationen. Während die UserID dazu dient, die Aktionen des Systems eindeutig dem Anwender zuzuordnen, ermöglicht die ActionID die Einordnung der ausgeführten Aktion. Diese Einordnung erfolgt über eine Konvention.

Anhand der mitgelieferten Parameter ist es dem VSS nun möglich den Anwender zu identifizieren, seinen aktuellen Aufenthalt auf dem Marktplatz (i. S. von Dialog) zu

bestimmen und seine ausgeführten Aktionen zu registrieren. Aus diesen drei Parametern kann der nächste Schritt im zugrundeliegenden Vorgang eingeleitet werden.

Die FormID ist einer inhaltlosen Seite zugeordnet, deren Inhalt über die Methode getForm ermittelt werden soll (siehe Methodenbeschreibung getForm in Abschnitt 0).

Parameterspezifikation

Im Folgenden werden mögliche Aktionen, die im Zusammenhang mit der Methode performAction eine Rolle spielen, näher beschrieben. Die Parameterspezifikation ist abhängig von ActionID.

ActionID	Login	Typ
ActionData	userName	String
	password	String
UserID	sessionID	String
Beschreibung	Benutzer einloggen	

ActionID	Logout	Typ
ActionData	Null	
UserID	sessionID	String
Beschreibung	Benutzer ausloggen	

ActionID	RemoveUser	Typ
ActionData	Null	
UserID	sessionID	String
Beschreibung	Benutzer entfernen	

ActionID	Registration	Typ
ActionData	prename	String
	lastname	String
	phone	String
	fax	String
	postCode	String
	place	String
	creditcardNr	String
UserID	sessionID	String
Beschreibung	Registrierung eines Benutzers im System	

ActionID	AddComponent	Typ
ActionData	componentName	String
	requiredOS	Int
	... (Daten entsprechend DB)	
UserID	sessionID	String
Beschreibung	Eine neue Komponente hinzufügen	

ActionID	SearchComponent	Typ
ActionData	componentName price priceCond currency requiredOS producerName	String Double {less, greater, equal} Int Int String
UserID	sessionID	String
Beschreibung	Stichwortsuche nach mehreren Kriterien	

ActionID	ChangeComponent	Typ
ActionData	componentName requiredOS ... (Daten entsprechend DB)	String Int
UserID	sessionID	String
Beschreibung	Komponentendaten aktualisieren	

ActionID	BrowseComponent	Typ
ActionData	category accessPath (Pfad auf dem „gestöbert“ wird, d. h. spezifiziert auf welchem Weg Benutzer gerade stöbert) Bspw.: AccessPath=(2, 3, 5) ? auf 1. Ebene, 2. Eintrag gewählt auf 2. Ebene, 3. Eintrag gewählt auf 3. Ebene, 5. Eintrag gewählt Nutzer sieht alle Einträge von Ebene 4	Int Int
UserID	sessionID	String
Beschreibung	Ebenensuche mit Katalogansicht	

ActionID	ShowComponent	Typ
ActionData	componentID	Int
UserID	sessionID	String
Beschreibung	Details zu einer Komponente anzeigen	

ActionID	ShowStatistics	Typ
ActionData	componentID Formulardaten für Statistik beinhalten: views (Anzahl Zugriffe auf Detail Ansicht der Komponente) buys (Anzahl Käufe) downloads creationDate lastChange changesTotal (Anzahl Änderungen an Komponente)	Int Int Int Int Date Date Int
UserID	sessionID	String
Beschreibung	Verkaufsstatistik anzeigen	

ActionID	BuyComponent	Typ
ActionData	componentID	Int
UserID	sessionID	String
Beschreibung	Komponente kaufen	

ActionID	DownloadComponent	Typ
ActionData	componentID	Int
UserID	sessionID	String
Beschreibung	Komponente herunterladen	

A.3.2 Methode getForm

Spezifikation	getForm (FormID formID)
Parameter	FormID: Identifikator für Formular
Rückgabe	FormData

Die Methode getForm liefert Metadaten und evtl. auch den Inhalt zur Darstellung eines Formulars. Handelt es sich um eine Datenneueingabe durch den Benutzer, wie bspw. bei der Registrierung, so liefert getForm lediglich die Metadaten. Bei einer Datenänderung hingegen, wie bspw. bei einer Registrierungsdatenänderung, liefert die Methode sowohl Metadaten als auch deren Inhalt. An die Metadaten und deren Inhalt gelangt getForm über eine Schnittstelle zum KDS (Kommunikations- und Datentransfersystem).

A.4 Schnittstelle Vorgangsteuerungssystem ↔ Kommunikationssystem

Für die Schnittstelle VSS/KSS wird die Methoden performAction bereitgestellt.

A.4.1 Methode performAction

Attribut	Ausprägung
Methodenname	performAction
Parameter	Objekt: Die Methode beinhaltet nur einen Parameter, der von verschiedenen Typen sein kann. Jedes übergebene Objekt muss die Attribute SessionID und ActionID beinhalten. Anhand dieser Attribute wird der aufzurufende Workflow erkannt.
Rückgabe	Objekt: Die Rückgabe ist ein Objekt der gleichen Struktur oder eine Fehlermeldung.

Das Vorgangsteuerungssystem leitet Anfragen und Aufgaben über die Methode performAction an das Kommunikationssystem weiter. Das Kommunikationssystem gliedert diese Aufgaben und verteilt die Teilaufgaben an weitere Systeme. Es wird nur eine Methode als Schnittstelle angeboten, anhand der ActionID kann festgestellt werden welcher Workflow gerade angestoßen wird. In den übergebenen Objekten sind

Metadaten enthalten. Die Metadaten beschreiben wie die Daten im Webinterface angezeigt werden sollen.

ActionID	Beschreibung
Login	Einloggen eines Benutzers. UserID und Passwort werden gebraucht.
Logout	Der Benutzer hat sich ausgeloggt.
Registration	Der Registrations Workflow wurde angestoßen.
SearchComponent	Eine Suche nach Komponenten soll durchgeführt werden. Es handelt sich um eine Suche nach bestimmten Parametern.
BrowseComponent	Es wird im Katalog nach Komponenten gesucht.
AddComponent	Eine Komponente soll hinzugefügt werden.
ChangeComponent	Eine Komponente soll geändert werden.
BuyComponent	Die ausgewählte Komponente soll gekauft werden.
RemoveUser	Der Benutzer möchte seine Registrierung vom Marktplatz löschen.
DownloadComponent	Ein Download von Komponenten wird angestoßen.
ChangeUserData	Der Benutzer möchte seine Daten ändern.
CreateSession	Wenn sich ein Benutzer am Marktplatz anmeldet, muss eine neue Session erstellt werden.
UpdateSession	Die aktuelle Session muss erneuert werden; z. B. weil sich der Benutzername geändert hat.
ShowStatistics	Die Statistiken sollen angezeigt werden.

Objekte

Die mit den ActionIDs im Zusammenhang stehenden Objekte (siehe Abschnitt A.2) sind: User Data, Authentication, Component, DownloadComponent, Category, Session, SubCategory, SearchComponent und WebPage

A.5 Schnittstelle Kommunikationssystem ↔ Dateitransfersystem

Für die Schnittstelle KSS/DTS wird die Methode performAction bereitgestellt.

A.5.1 Methode performAction

Spezifikation	Methode performAction
Parameter	DownloadComponent object
Rückgabe	Fehlermeldung oder DownloadComponent Objekt

Über die Methode performAction werden sämtliche Funktionen des Dateitransfersystems aufgerufen. Die übergebenen Objekte haben sessionId und actionID als Attribute gemeinsam. Anhand dieser Attribute kann festgestellt werden welche Aktion ausgeführt werden soll.

ActionID	Beschreibung
show	Ein Download soll gestartet werden.
delete	Ein Download ist erfolgreich abgeschlossen.
abort	Ein Download wurde abgebrochen.

Objekte

Das mit den ActionIDs im Zusammenhang stehende Objekt (siehe Abschnitt A.2) ist: DownloadComponent.

A.6 Schnittstelle Kommunikationssystem ↔ Datenbank-Wrapper

Für die Schnittstelle KSS/DBWrapper wird die Methode action bereitgestellt:

A.6.1 Methode action

Diese Methode ist die einzige Methode, die der DB-Wrapper bereitstellt. Verschiedene Funktionalitäten werden über unterschiedliche Parameter und ActionIDs initialisiert. Dementsprechend ist diese Methode überladen. Das Rückgabeobjekt der Methode richtet sich nach der ActionID.

Spezifikation	Methode action
Parameter	Object object
Rückgabe	Object object

Die Parameter können vom Typ

- a. Component
 - b. Category
 - c. Authentication
 - d. DownloadComponent
 - e. UserData
 - f. WebPage
 - g. FormularLayout
 - h. Session
- sein.

Rückgabe: Jedes Objekt enthält eine Variable actionID, je nach Wert der Variable kann das Rückgabeobjekt vom folgenden Typ sein:

- a. Component
- b. Category

- c. Authentication
- d. DownloadComponent
- e. UserData
- f. WebPage
- g. FormularLayout
- h. Void

Das konkrete Rückgabeobjekt kann aus folgender Tabelle entnommen werden.

ActionID	Beschreibung
change	Es soll ein Datensatz geändert werden.
insert	Es soll ein neuer Datensatz eingefügt werden.
delete	Es soll ein Datensatz gelöscht werden.
abort	Es soll ein Vorgang abgebrochen werden.
show	Es sollen Daten angezeigt werden.

Objekte

Die mit den ActionIDs im Zusammenhang stehenden Objekte (siehe Abschnitt A.2) sind: User Data, Authentication, Component, DownloadComponent, Category, Session, SubCategory, SearchComponent und WebPage.

Im Folgenden sollen mögliche Aktionen, die im Zusammenhang mit der Methode action eine Rolle spielen, näher beschrieben werden. Die Parameterspezifikation ist abhängig von ActionID.

Alle ActionIDs sind Strings

Objects	Action Ids	Rückgabeobjekt	Kommentar
Component	insert	Void	
	show	Component	Suche nach Komponenten; je nach gefüllten Attributen
	delete	Void	
	change	Void	
UserData	insert	Void	Registrierung
	show	UserData	
	delete	Void	
	change	Void	
DownloadComponent	show	DownloadComponent	show=download durchführen
	delete	Void	nach erfolgreichem Download
	abort	Void	
Category	show	Category	
Authentication	show	Authentication	Übergeben der User Ids

A.7 Dokumentation der Schemata und Metaschemata

A.7.1 DBKat – Komponentenkatalog-Datenbank

Die Komponentenbeschreibung erfolgt mit Hilfe des Spezifikationsrahmens der GI⁶²⁴ für Fachkomponenten (siehe Abschnitt 3.5.2.5). Daher umfasst DBKat alle notwendigen Relationen zur Umsetzung des Spezifikationsrahmens. Diese Spezifikation basiert auf sieben Beschreibungsebenen, die im Folgenden einzeln behandelt werden.

A.7.1.1 Vermarktungsebene

Die Vermarktungsebene beinhaltet im wesentlichen vier Kategorien von Daten: grundlegende Informationen über abzulegende Komponenten wie Komponententname und –version, Informationen über Bezugsmodalitäten (bspw. Komponentenpreis, Kontaktdaten und Vertragsbedingungen), klassifizierende Daten, welche die hierarchische Kategorisierung nach Wirtschaftszweig und Anwendungsdomäne ermöglichen sowie Anforderungen an die zugrunde liegende Systemarchitektur (untergliederbar in Hardware, Software und Basissysteme).

Relationenschemata

„Component_Type“		Art der Komponente (Fach-/Systemkomponente, Framework)				Integritätsbedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	ID_Type	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	FatherType	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Type.ID_Type	<i>optional</i>
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen:						
➤ es sollte genau ein Tupel ohne Eintrag in „FatherType“ existieren						

Anmerkungen:

- verwendete Fachbegriffe: ComponentType, Fachkomponente, Systemkomponente, Framework, Komponenten-Anwendungs-Framework, Komponenten-System-Framework
- „FatherType“ ermöglicht eine hierarchische Strukturierung der Komponententypen.

⁶²⁴ Vgl. Ackermann et al. 2002.

„Component“		Hauptdatenstruktur einer abzulegenden Software-Komponente				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Component	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Version	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
4	Price	Dec(10,2)	<input type="checkbox"/>	<input type="checkbox"/>		<i>Price ≥ 0</i>
5	Currency	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Currency.ID_Currency	<i>optional</i>
6	Company	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ContactPersonFor.Company, Company.ID_Company	<i>optional</i>
7	Person	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ContactPersonFor.Person, Person.ID_Person	<i>optional</i>
8	BankAccount	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	BankAccount.ID_BankAccount	<i>optional</i>
9	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
10	ContractualConditions	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
11	Size	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>Size ≥ 0</i>
12	ComponentType	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Type.ID_Type	
13	UsedTechnology	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Technology.ID_Technology	<i>optional</i>
14	UsedFramework	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Framework.ID_Framework	<i>optional</i>
15	UsedDBMS	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_DBMS.ID_DBMS	<i>optional</i>
16	RequiredCPU	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
17	RequiredMM	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional, MM ≥ 0</i>
18	RequiredHDD	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>HDD ≥ 0</i>
19	RequiredOS	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_OS.ID_OS	

Integritätsbedingungen:
 $\neg(\text{Price} = 0) \vee ((\text{Currency ist gültiger Verweis}) \wedge (\text{BankAccount ist gültiger Verweis}))$

Anmerkungen:

- verwendete Fachbegriffe: Component, ProducingCompany, ContactPerson, ComponentType, Technology, Framework, DBMS, OS
- verwendete Abkürzungen: DBMS, CPU, MM, HDD, OS
- „UsedFramework“ ist optional, da Frameworks auch Komponenten darstellen, selbst aber in der Regel auf kein anderes Framework zurückgreifen.

„Component_Storage“		<i>Physische Speicherung der Komponente</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	Data	BLOB	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component
- Um flexible Erweiterbarkeit und Gestaltungsmöglichkeiten zu eröffnen, werden die physischen Komponentendaten getrennt von den Beschreibungsinformationen gehalten.

„Component_Currency“		<i>Währungen, in denen Komponentenpreise angegeben sind</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Currency	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Country	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: (keine)						

Anmerkungen: (keine)

„Component_Technology“		<i>Technologien, auf denen die Komponenten basieren</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Technology	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Producer	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Version	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Technology
- „Producer“ korrelieren nicht mit „Companies“ aus DBUser; sie werden in DBKat nicht weiter geführt.

„Component_Framework“		Frameworks, auf denen die Komponenten basieren				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Framework	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Producer	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Version	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Framework, Komponenten-Anwendungs-Framework, Komponenten-System-Framework
- „Producer“ korrelieren nicht mit „Companies“ aus DBUser; sie werden in DBKat nicht weiter geführt.

„Component_ScopeOfSupply“		Lieferumfang der Komponenten				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_SupplyNumber	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Component	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
3	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component
- Beispiele für Lieferumfang (neben der eigentlichen Komponente) sind Handbücher, Workshop-Gutscheine, Beratungsdienstleistungen, Einführungsvideos etc.

„Component_DBMS“		<i>Datenbank-Systeme, die von den Komponenten benötigt werden</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
					Verweis	
1	ID_DBMS	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Producer	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Version	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: <i>(keine)</i>						

Anmerkungen:

- verwendete Fachbegriffe: DBMS
- verwendete Abkürzungen: DBMS
- „Producer“ korrelieren nicht mit „Companies“ aus DBUser; sie werden in DBKat nicht weiter geführt.

„Component_Domain“		<i>Anwendungsdomänen der Komponenten (m:n - Materialisierung)</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
					Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	Domain	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_DomainStructure.ID_Domain	
Integritätsbedingungen: <i>(keine)</i>						

Anmerkungen:

- verwendete Fachbegriffe: Component, Domain
- Eine Domäne kann viele zugeordnete Komponenten besitzen; eine Komponente kann jedoch auch mehreren Domänen zugeordnet sein.

„Component_OS“		<i>Betriebssysteme, die von den Komponenten benötigt werden</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_OS	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Producer	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Version	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: <i>(keine)</i>						

Anmerkungen:

- verwendete Fachbegriffe: OS
- verwendete Abkürzungen: OS
- „Producer“ korrelieren nicht mit „Companies“ aus DBUser; sie werden in DBKat nicht weiter geführt.

„Component_DomainStructure“		<i>Anwendungsdomänen der Komponenten als Baumstruktur</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Domain	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	FatherDomain	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_DomainStructure.ID_Domain	<i>optional</i>
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen:						
➤ <i>es sollte genau ein Tupel ohne Eintrag in „FatherDomain“ existieren</i>						

Anmerkungen:

- verwendete Fachbegriffe: Domain
- „FatherDomain“ ermöglicht eine hierarchische Strukturierung der Domänen.

„Component_IndustrialSector“		<i>Wirtschaftszweige der Komponenten (m:n - Materialisierung)</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	IndustrialSector	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_IndustrialSector.ID_IndustrialSector	
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, IndustrialSector
- Ein Wirtschaftszweig kann viele zugeordnete Komponenten besitzen; eine Komponente kann jedoch auch mehreren Wirtschaftszweigen zugeordnet sein.

„Component_IndustrialSectorStructure“		<i>Wirtschaftszweige der Komponenten als Baumstruktur</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_IndustrialSector	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	FatherIndustrialSector	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_IndustrialSector.ID_IndustrialSector	<i>optional</i>
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen:						
➤ es sollte genau ein Tupel ohne Eintrag in „FatherIndustrialSector“ existieren						

Anmerkungen:

- verwendete Fachbegriffe: IndustrialSector
- „FatherIndustrialSector“ ermöglicht eine hierarchische Strukturierung der Wirtschaftszweige.

Es existieren keine interrelationalen Integritätsbedingungen in der Vermarktungsebene.

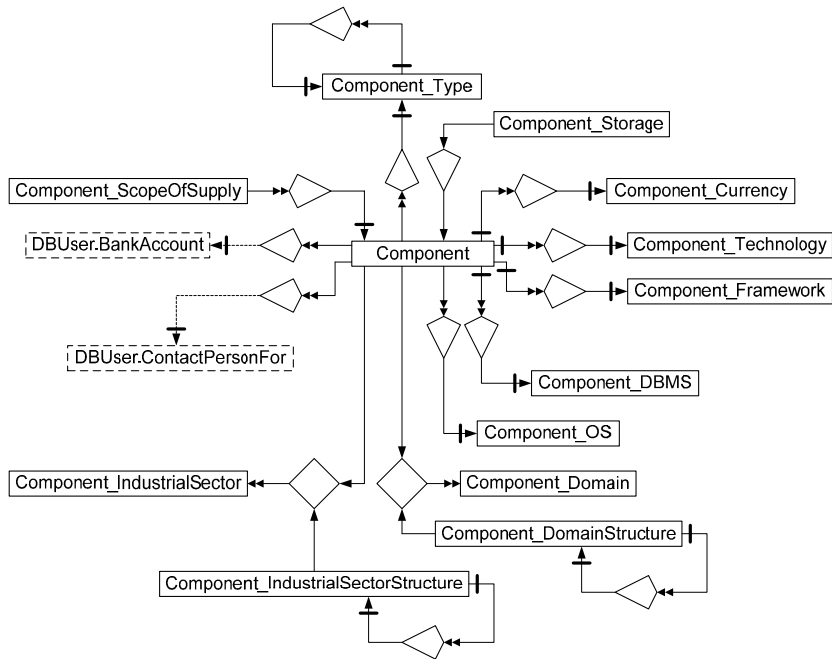


Abbildung 63: Objektypendiagramm⁶²⁵ DBKat – Vermarktungsebene

A.7.1.2 Aufgabenebene

Die Aufgabenebene erfasst normsprachlich die Funktionalität der Komponente, wobei von den einzelnen, an der Schnittstelle bereitgestellten Prozeduren, abstrahiert wird. Aufgaben können hierbei hierarchisch (sowohl abstraktiv als auch kompositiv) beliebig zerlegt und von entsprechenden Prozeduren der Schnittstelle referenziert werden. Hierbei wird von DBNorm Gebrauch gemacht, welche die Normsprachenfunktionalität zur Verfügung stellt.

⁶²⁵ Zur Notation vgl. Mertens 2001, S. 342f., Stichwort „Objektypenmethode“.

Relationenschemata

„Component_Tasks“		<i>Normsprachlich beschriebene Aufgaben der Komponenten</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	ID_SentenceOfComponent	Int	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	SentenceStructure	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SentenceStructure.ID_SentenceStructure	
4	FatherTask	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Tasks.ID_SentenceOfComponent	
Integritätsbedingungen:						
<ul style="list-style-type: none"> ➤ ID_SentenceOfComponent muss innerhalb jeder Komponente eindeutig sein ➤ es sollte genau ein Tupel ohne Eintrag in „FatherTask“ existieren 						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Tasks, SentenceStructure
- „FatherTask“ ermöglicht eine hierarchische Strukturierung der Aufgaben.

„Component_Task_Terms“		<i>Materialisierte m:n - Zuordnung von Fachbegriffen zu den zur Aufgabenbeschreibung der Komponenten verwendeten Satzbauplänen</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Tasks.Component	
2	SentenceOfComponent	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Tasks.ID_SentenceOfComponent	
3	PlaceholderPosition	Int	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4	PlaceholderTerm	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Tasks, SentenceStructure, Placeholder, Term

Interrelationale Integritätsbedingungen

- „PlaceholderPosition“ in „Task_Terms“ muss einen Wert zwischen „1“ und „NoOfPlaceholders“ des entsprechenden Eintrags in „SentenceStructure“ besitzen:
- $1 \leq \text{Task_Terms}[\text{comp},\text{sent}].\text{PlaceholderPosition} \leq \text{SentenceStructure}[\text{Task_Terms}[\text{comp},\text{sent}].\text{SentenceStructure}].\text{NoOfPlaceholders}$

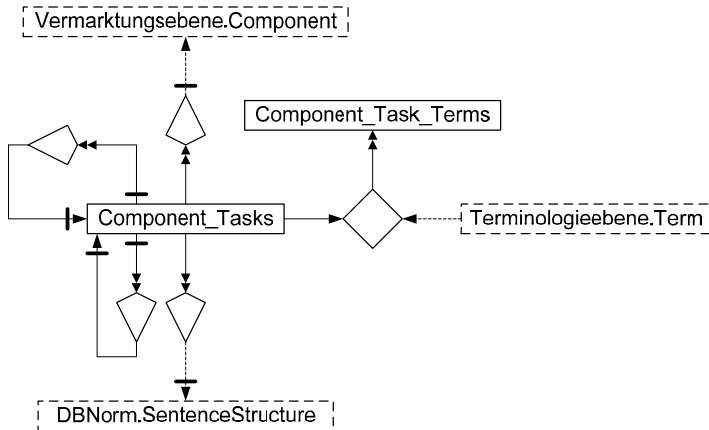


Abbildung 64: Objektypendiagramm DBKat – Aufgabenebene

A.7.1.3 Terminologieebene

Die Terminologieebene des GI-Spezifikationsrahmens ist Teil der Normsprache und daher in DBNorm enthalten.

A.7.1.4 Qualitätsebene

Die Qualitätsebene erlaubt die hierarchische Speicherung von Qualitätsklassen mit zugeordneten Qualitätseigenschaften, welche wahlweise für eine gesamte Komponente oder für einzelne Komponentenfunktionen gelten kann. In beiden Fällen werden Qualitätsausprägungen gespeichert, welche sich zwecks Vergleichbarkeit explizit auf im System vorgegebene Referenzumgebungen beziehen.

Relationenschemata

„QualityClass“		Qualitätsklassen der Komponenten				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_QualityClass	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Father	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	QualityClass.ID_QualityClass	
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	ShortDescription	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
5	FullDescription	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: QualityClass
- „Father“ ermöglicht eine hierarchische Strukturierung der Qualitätsklassen.

„QualityProperty“		Qualitätsmerkmale der Qualitätsklassen				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_QualityProperty	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	QualityClass	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	QualityClass.ID_QualityClass	
3	DynamicProperty	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
4	PropertyForComponent	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
5	ShortDescription	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
6	FullDescription	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
7	UnitOfMeasurement	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: QualityClass, QualityProperty, DynamicProperty

„QualityOfComponent“			<i>Ausprägung eines komponentenbezogenen Qualitätsmerkmals als materialisierte ternäre Beziehung</i>			
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_Function.Component	
2	QualityProperty	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	QualityProperty.ID_QualityProperty	
3	TestEnvironment	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TestEnvironment.ID_TestEnvironment	
4	Value	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, QualityProperty, TestEnvironment

„QualityOfFunction“			<i>Ausprägung eines funktionsbezogenen Qualitätsmerkmals als materialisierte ternäre Beziehung</i>			
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_Function.Component	
2	NameOfFunction	VarChar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_Function.Name	
3	QualityProperty	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	QualityProperty.ID_QualityProperty	
4	TestEnvironment	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TestEnvironment.ID_TestEnvironment	
5	Value	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Interface_Function, QualityProperty, TestEnvironment

„TestEnvironment“		<i>Testumgebung für Qualitätsmerkmale</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_TestEnvironment	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: TestEnvironment

Interrelationale Integritätsbedingungen

- $QualityProperty[i].PropertyForComponent \rightarrow \neg \exists j. QualityOfFunction[j].QualityProperty = i$
- $\neg QualityProperty[i].PropertyForComponent \rightarrow \neg \exists j. QualityOfComponent[j].QualityProperty = i$

Objektypendiagramm siehe Abbildung 65.

A.7.1.5 Abstimmungsebene

Die Abstimmungsebene hat zum Ziel, die Reihenfolgebeziehungen zwischen den Funktionen der Komponenten in formaler Notation (TemporalOCL, siehe Abschnitt 3.5.2.9) zu spezifizieren.

Relationenschemata

„Component_Coordination“		<i>Reihenfolgebeziehungen zw. den Diensten der Komponenten</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	TemporalOCL	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>Stellt eine wohlgeformte, semantisch korrekte TemporalOCL-Datei dar</i>
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Coordination, TemporalOCL
Es existieren keine interrelationalen Integritätsbedingungen in der Abstimmungsebene.

Objektypendiagramm siehe Abbildung 65.

A.7.1.6 Verhaltensebene

Die Verhaltensebene spezifiziert in formaler Notation (OCL, siehe Abschnitt 3.5.2.8) die Vor- und Nachbedingungen sowie Invarianten der Komponenten.

Relationenschemata

„Component_Behaviour“		<i>Vor- und Nachbedingungen sowie Invarianten der Komponenten</i>				Integritätsbedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	OCL	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>Stellt eine wohlgeformte, semantisch korrekte OCL-Datei dar</i>
Integritätsbedingungen: <i>(keine)</i>						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Behaviour, OCL
Es existieren keine interrelationalen Integritätsbedingungen in der Verhaltensebene.

Objektypendiagramm siehe Abbildung 65.

A.7.1.7 Schnittstellenebene

Die Schnittstellenebene umfasst neben der direkt von Programmierumgebungen verwendbaren IDL der Komponenten die Zuordnung von IDL-Funktionen zu Aufgaben der Aufgabenebene sowie von IDL-Datenobjekten zu Termini der Terminologieebene, wodurch semantische Korrektheit gewährleistet werden kann. Eine IDL-Funktion kann hierbei mehreren Aufgaben zugeordnet werden, ein Datenobjekt jedoch nur mit einem Terminus übereinstimmen.

Relationenschemata

„Component_Interface“		<i>Beschreibung der Komponentenschnittstellen in IDL</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	IDL	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>Stellt eine wohlgeformte, semantisch korrekte IDL-Datei dar</i>
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Interface, IDL

„Component_Interface_Function“		<i>Auflistung der IDL-Funktionen einer Komponente</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	Name	VarChar	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Interface_Function

„Component_Interface_DataObject_Term“		<i>Zuordnung der IDL-Datenobjekte zu Termini der Terminologieebene</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_DataObject.Component	
2	NameOfDataObject	VarChar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_DataObject.Name	
3	Term	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Interface_DataObject, Term

„Component_Interface_Function_Tasks“					<i>Materialisierte m:n-Beziehung zw. IDL-Funktionen zu Aufgaben der Komponenten</i>	
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_Function. Component	
2	Name	VarChar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Interface_Function.Name	
3	Task	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component_Tasks. ID_SentenceOfComponent	
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Interface_Function, Component_Tasks

„Component_Interface_DataObject“					<i>Auflistung der IDL-Datenobjekte einer Komponente</i>	
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	Name	VarChar	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Component, Component_Interface_DataObject

Interrelationale Integritätsbedingungen

- $\forall i. \text{Component_Interface_Function}[i] \in \text{Functions}(\text{IDL})$
- $\forall i. \text{Interface_DataObject}[i] \in \text{DataObjects}(\text{IDL})$

Objektypendiagramm siehe Abbildung 65.

Objektypendiagramme von Abstimmungs-, Verhaltens-, Schnittstellen- und Qualitätsebene

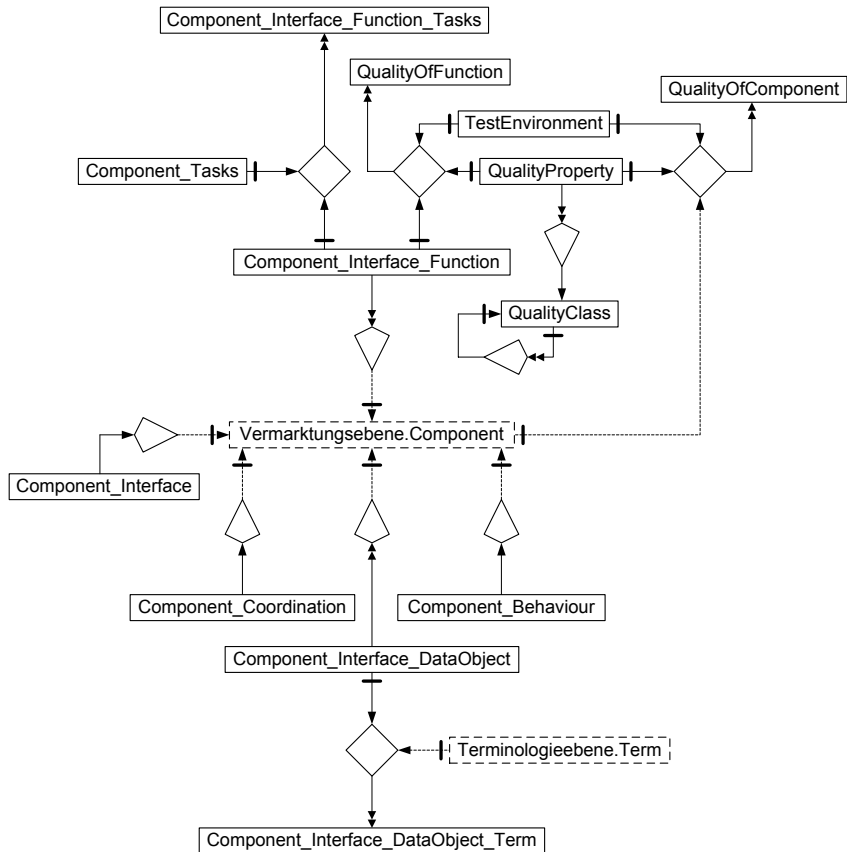


Abbildung 65: Objektypendiagramm DBKat – Abstimmungs-, Verhaltens-, Schnittstellen- und Qualitätsebene

A.7.1.8 Unterstützung semantischer Stichwortsuchen

Die Unterstützung semantischer Suchen anhand von Stichwörtern erleichtert das gezielte Auffinden von Komponenten insbesondere im Bereich mobiler Endgeräte, welche aufgrund gegebener Geräterestrictionen keine komplexen Suchanfragen auf den im Rahmen von DBKat zur Verfügung gestellten Relationenschemata gestatten. Stichwörter besitzen hierbei eine „Nähe“ zueinander, sodass von eingegebenen Stichwörtern zu „Nachbarn“ (gemäß den im System abgelegten Distanzmaßen) übergegangen werden kann (fuzzy search).

Relationenschemata

„Keyword“		<i>Speicherung eines Suchschlüsselwortes</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Keyword	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Keyword

„Keyword_Connection“		<i>Gewichtete (gerichtete) Netzstruktur zw. den Schlüsselwörtern</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Keyword1	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Keyword.ID_Keyword	
2	Keyword2	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Keyword.ID_Keyword	
3	Proximity	Real	<input type="checkbox"/>	<input type="checkbox"/>		$0 \leq Proximity \leq 1$
Integritätsbedingungen:						
➤ $Keyword1 \neq Keyword2$						

Anmerkungen:

- verwendete Fachbegriffe: Keyword, Keyword_Connection
- „Proximity“ bestimmt die semantische und pragmatische „Nähe“ zweier Schlüsselwörter.
- Synonyme können bspw. dadurch realisiert werden, dass sie symmetrisch mit Proximity „1.0“ verknüpft werden (alle weiteren Verbindungen müssen lediglich eines der Synonyme betreffen).

„Component_Keywords“		Zuordnung von Schlüsselwörtern zu Komponenten				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Component	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	
2	Keyword	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Keyword.ID_Keyword	
3	Fit	Real	<input type="checkbox"/>	<input type="checkbox"/>		$0 \leq \text{Fit} \leq 1$

Integritätsbedingungen: (keine)

Anmerkungen:

- verwendete Fachbegriffe: Component, Keyword
- „Fit“ bestimmt die semantische und pragmatische „Nähe“ eines Schlüsselworts zu einer Komponente.

Es existieren keine interrelationalen Integritätsbedingungen in diesem Abschnitt.

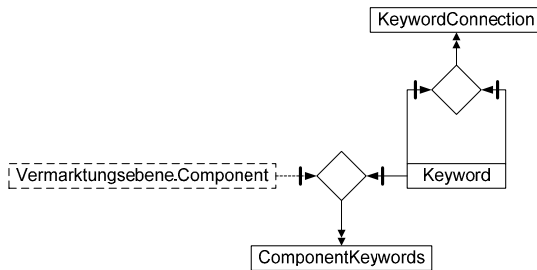


Abbildung 66: Objektypendiagramm DBKat – Semantische Stichwortsuche

A.7.1.9 Objekttypendiagramm von DBKat

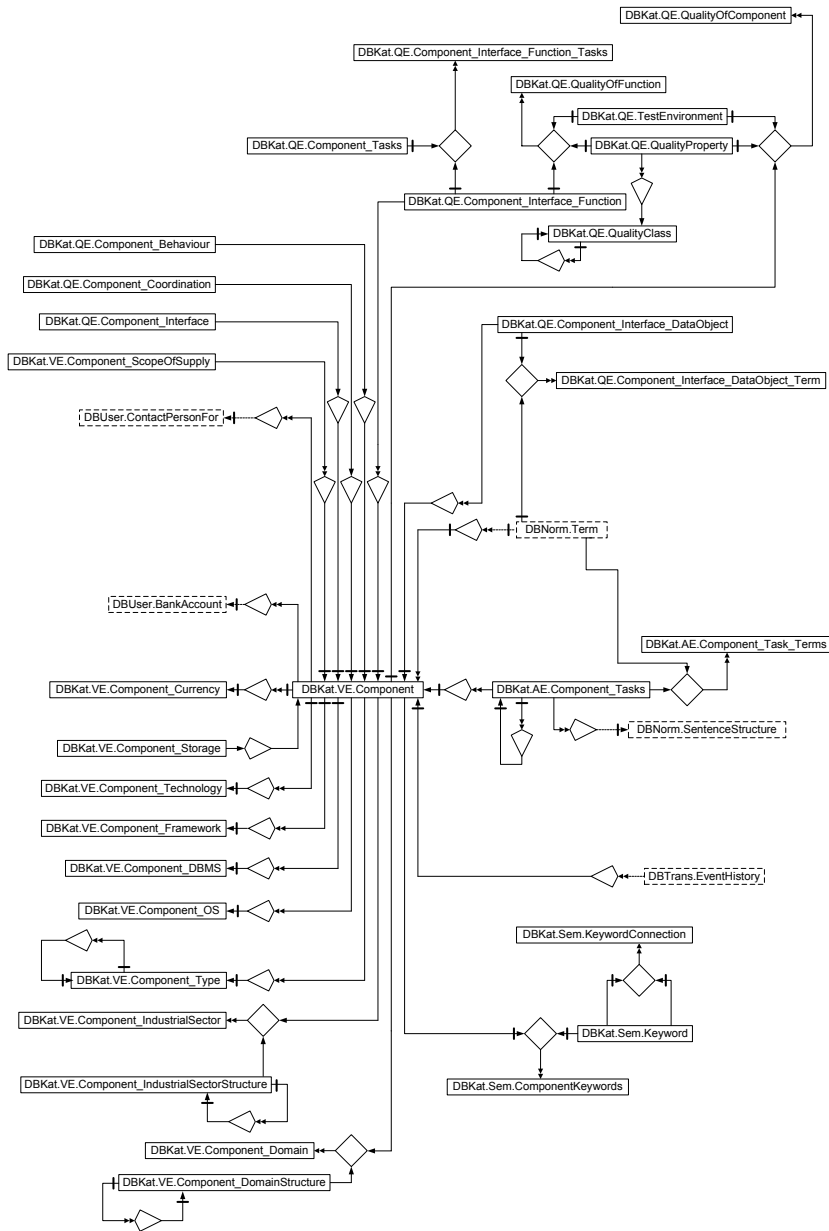


Abbildung 67: Objekttypendiagramm DBKat

A.7.2 DBUser – Benutzer-Datenbank

DBUser speichert sämtliche dem System bekannten natürlichen und juristischen Entitäten strukturiert ab. Hierbei wird unterschieden zwischen natürlichen Personen (Datenstruktur „Person“), welche sowohl als Systembenutzer (verbunden mit einem Login-Account) als auch als Ansprechpartner für Unternehmen (Datenstruktur „ContactPerson“ als Resultat der Konnexion mit „Company“) fungieren können und juristischen Personen, welche zur Identifikation des Herstellers der Komponenten dienen. Sowohl Unternehmen als auch Benutzer können über Bankkonten verfügen, deren Daten einheitlich in „BankAccount“ gehalten werden.

Relationenschemata

„Person“		<i>Hauptstruktur einer natürlichen Person (muss nicht angemeldet sein im System)</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_Person	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Salutation	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
3	FirstName	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
4	MiddleNames	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
5	Surname	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
6	Street	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
7	Sex	Char	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
8	ZIP	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
9	City	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
10	Country	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
11	eMail	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
12	Telephone	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
13	Mobile	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
14	Fax	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
15	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen:						
➤ \neg NULL ((Street \wedge ZIP \wedge City \wedge Country) \vee eMail \vee Telephone \vee Mobile \vee Fax)						

Anmerkungen:

- verwendete Fachbegriffe: Person

„LoginAccount“		<i>Account für den Systemzugang</i>				Integritäts- bedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	ID_LoginAccount	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Person	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Person.ID_Person	
3	LoginName	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>Unique</i>
4	LoginPassword	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>evtl. Sicherheitsanforderungen</i>
5	BankAccount	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	BankAccount.ID_BankAccount	
6	IsVendor	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: <i>(keine)</i>						

Anmerkungen:

- verwendete Fachbegriffe: LoginAccount, BankAccount, Vendor

„ContactPersonFor“		<i>Materialisierte m:n-Beziehung, die Personen Firmen zuordnet</i>				Integritäts- bedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	Company	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Company.ID_Company	
2	Person	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Person.ID_Person	
3	Position	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: <i>(keine)</i>						

Anmerkungen:

- verwendete Fachbegriffe: ContactPerson, Company, Person

„Company“		Herstellerfirmen der Komponenten				Integritätsbedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	ID_Company	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	BankAccount	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	BankAccount.ID_BankAccount	
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
4	Street	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
5	ZIP	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
6	City	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
7	Country	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
8	eMail	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
9	Telephone	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
10	Mobile	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
11	Fax	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
12	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Company, BankAccount

„BankAccount“		Kontodaten; Konten werden von Personen oder Firmen besessen				Integritätsbedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	ID_BankAccount	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	AccountNumber	Int	<input type="checkbox"/>	<input type="checkbox"/>		
3	BankID	Int	<input type="checkbox"/>	<input type="checkbox"/>		
4	BankName	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
5	IsCreditCard	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
6	CreditCardNumber	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen:						
➤ $\neg \text{IsCreditCard} \vee (\text{CreditCardNumber ist gültig})$						

Anmerkungen:

- verwendete Fachbegriffe: BankAccount

Es existieren keine interrelationalen Integritätsbedingungen in DBUser.

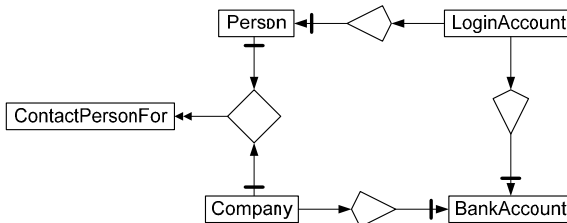


Abbildung 68: Objekttypendiagramm DBUser

A.7.3 DBMeg – Endgeräte-Datenbank

DBMeg dient der strukturierten, hierarchischen Gliederung der vom Marktplatz unterstützten mobilen Endgeräte. Neben Subsumption des jeweiligen Gerätetyps unter eine der hierarchisch gegliederten Geräteklassen werden Informationen abgelegt, die zur Darstellung der Marktplatz-Bedienungsumgebung auf dem Endgerät sowie der Unterstützungsfunktionalität bezüglich Downloads von anderen Marktplatzkomponenten verwendet wird.

Relationenschemata

„DeviceClass“		<i>Daten über die Geräteklassen, denen ein oder mehrere Gerätetypen zugeordnet werden</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_DeviceClass	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
4	FatherClass	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	DeviceClass.ID_DeviceClass	
Integritätsbedingungen:						
➤ es sollte genau ein Tupel ohne Eintrag in „FatherClass“ existieren						

Anmerkungen:

- verwendete Fachbegriffe: DeviceType, DeviceClass
- „FatherClass“ ermöglicht eine hierarchische Strukturierung der Komponententypen.

Es existieren keine interrelationalen Integritätsbedingungen in DBMeg.

„DeviceType“		<i>Daten über einen einzelnen Gerätetyp</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_DeviceType	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Producer	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
4	DeviceClass	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	DeviceClass.ID_DeviceClass	
5	ResolutionX	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
6	ResolutionY	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
7	ColorDepth	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
8	MemorySize	Int	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
9	OS	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component_OS.ID_OS	<i>optional</i>
10	WMLSupport	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
11	HTMLSupport	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
12	Description	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen:						
➤ (WMLSupport \vee HTMLSupport)						

Anmerkungen:

- verwendete Fachbegriffe: DeviceType, DeviceClass, OS
- Producer hängt nicht mit entsprechenden Herstellertabellen von DBUser und DBKat zusammen.

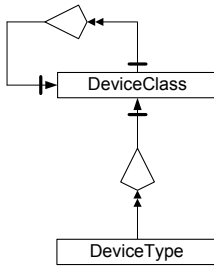


Abbildung 69: Objekttypendiagramm DBMeg

A.7.4 DBTrans – Transaktionen-Datenbank

DBTrans dient als operative Logbuch-Datenbank; jegliche systemrelevanten Ereignisse werden hier chronologisch – zu Debugging-Zwecken, als Datenfundament für Statistiken und für Sicherheitsabfragen (bspw. versuchtes Mehrfach-Einloggen) etc. – protokolliert.

Systemrelevante Ereignisse lassen sich untergliedern in benutzer-, komponenten- und systemorientierte Ereignisse. Benutzerorientierte Ereignisse sind Login und Logout von Benutzern, Registrierung und Abmeldung vom Marktplatzsystem sowie Änderung von Benutzerdaten. Komponentenorientierte Ereignisse sind Auswählen der Detailansicht einer Komponente, Kauf und Download von Komponenten, Einfügen, Ändern und Löschen von Komponenten. Eine strukturierte Gliederung der im Gesamtsystem möglichen Fehlerfälle ist den systemorientierten Ereignissen zu subsumieren (siehe hierzu auch DBError).

Relationenschemata

„EventType“		Beschreibung der versch. Ereignistypen				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_EventType	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	TableName	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
3	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen:						
➤ <i>TableName ist entweder NULL oder verweist auf eine im System existente Relation</i>						

Anmerkungen:

- verwendete Fachbegriffe: EventType

„EventHistory“		<i>Hauptstruktur zur chronologischen Speicherung relevanter Ereignisse</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	ID_Event	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Timestamp	Timestamp	<input type="checkbox"/>	<input type="checkbox"/>		
3	EventType	Int	<input type="checkbox"/>	<input type="checkbox"/>	EventType.ID_EventType	
4	LoginAccount	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	LoginAccount.ID_LoginAccount	<i>optional</i>
5	Component	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component	<i>optional</i>
6	UserIPAddress	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>optional</i>
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Event, EventHistory, EventType, LoginAccount, Component

„EventType_Login“		<i>Nähere Informationen zu Login-Ereignissen</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Event	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EventHistory.ID_Event	
2	EnteredLoginName	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	EnteredPassword	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Event, EventType
- Eine Tabelle „EventType_Logout“ wäre unnötig, da entsprechende Informationen des korrelierenden Login-Eintrags herangezogen werden können.

„EventType_BuyComponent“		Kauf von Komponenten				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Event	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Event.ID_Event	
2	BankAccount	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	BankAccount.ID_BankAccount	

Integritätsbedingungen: (keine)

Anmerkungen:

- verwendete Fachbegriffe: Event, EventType, Component, BankAccount

„EventType_Error“		Fehlerereignisse				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Event	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Event.ID_Event	
2	ErrorCode	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Errors.ID_Error	
3	ErrorMessage	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		

Integritätsbedingungen: (keine)

Anmerkungen:

- verwendete Fachbegriffe: Event, EventType, Component, BankAccount

Es existieren keine interrelationalen Integritätsbedingungen in DBTrans.

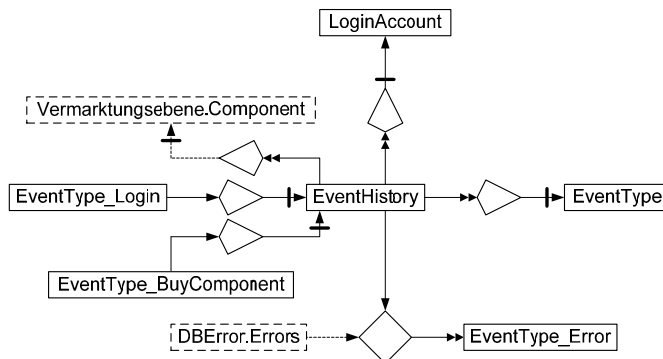


Abbildung 70: Objekttypendiagramm DBTrans

A.7.5 DBMIS – Metainformationssystem

DBMIS stellt Metainformationen, d. h. Informationen über die im System existierenden Schemata (Konzepte wie Relationen, Attribute, Integritätsbedingungen etc.) strukturiert dar. Hierbei erfolgt eine transparente Auflösung von Synonymen auf Attributenebene durch den Bezug auf die normsprachliche Daten (DBNorm).

Relationenschemata

„Attribute“		Metadaten über im System existenten, den Tabellen zugeordneten Attributen				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_Attribute	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	AttributeName	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	RelationName	VarChar	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Relation.ID_Relation	
4	Term	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	<i>optional</i>
5	DataType	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
6	PartOfPrimaryKey	Boolean	<input type="checkbox"/>	<input type="checkbox"/>		
7	ForeignKey	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Attribute.ID_Attribute	
8	Constraint	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
9	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Attribute, Relation, Term, DataType, PrimaryKey, ForeignKey, Constraint

„Relation“		Metadaten über im System existente Tabellen				Integritätsbedingungen
#	Attribut		Schlüsseleigenschaften			
	Name	Typ	PK	FK	Verweis	
1	ID_Relation	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	Term	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	<i>optional</i>
4	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
5	ExecutiveAdmin	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
6	CreationDate	Date	<input type="checkbox"/>	<input type="checkbox"/>		
7	DateOfLastBackup	Date	<input type="checkbox"/>	<input type="checkbox"/>		

Integritätsbedingungen: (keine)

Anmerkungen:

- verwendete Fachbegriffe: Relation, Term

Es existieren keine interrelationalen Integritätsbedingungen in DBMIS.

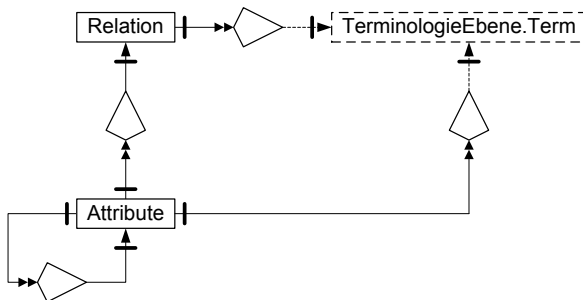


Abbildung 71: Objektypendiagramm DBMIS

A.7.6 DBNorm – Normsprachenschema

Die Normsprachenebene stellt eine ontologiebasierte Funktionalität für zwei Sprachenebenen bereit: Auf der objektsprachlichen Ebene werden Termini und Satzbaupläne zur Beschreibung von im Marktplatz gehandelten Komponenten abgelegt, auf der Metaebene hingegen werden metasprachlichen Konstrukten (Attribute, Relationen) Termini zugeordnet, sodass eine transparente Synonymitätsauflösung ermöglicht wird.

Zur Erreichung dieser Ziele erfolgt eine explizite, schrittweise und zirkelfreie Definition von Fachbegriffen (Termini), welche nicht mit den Schlüsselbegriffen (keywords) zu verwechseln sind: Während erstere in der Aufgabenebene verwendet werden und in der Regel Datenstrukturen von Komponenten normalsprachlich bezeichnen, dienen Schlüsselwörter der semantischen Suche nach Komponenten. Definiert werden Termini nach den gängigen Definitionsregeln – neben obligatorischer Kurzdefinition sind Langdefinitionen, Prädikatorenregeln und Verwendungsbeispiele möglich. Desweiteren können Termini zueinander in Beziehungen (z. B. „ist Gattung von“ oder „ist Schlüssel für“) stehen.

Objektsprachliche Termini können entweder global – dies ist insbesondere zur Standardisierung und der daraus resultierenden Schaffung von Interoperabilität wichtig – oder lokal für einzelne Komponenten gelten. Metasprachliche Termini sind in diesem Sinne stets als global anzusehen.

Relationenschemata

„Term“		<i>Hauptstruktur zur Definition eines (globalen oder lokalen) Terms</i>			
#	Attribut		Schlüsseleigenschaften		Integritätsbedingungen
	Name	Typ	PK	FK	
				Verweis	
1	ID_Term	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>	
3	Component	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Component.ID_Component <i>optional</i>
4	LanguageLevel	Int	<input type="checkbox"/>	<input type="checkbox"/>	
5	ShortDefinition	VarChar	<input type="checkbox"/>	<input type="checkbox"/>	
Integritätsbedingungen:					
➤ <i>(LanguageLayer = 1 ∨ Component = NULL)</i>					

Anmerkungen:

- verwendete Fachbegriffe: Term, ShortDefinition, LanguageLevel
- Termini gehören einer bestimmten Sprachebene (LanguageLevel) an
 - „1“ signalisiert die Objektsprachebene und damit die handelbaren Komponenten des Marktplatzes; Termini dieser Ebene können global definiert sein (in diesem Falle besitzt das Feld „Component“ den Wert „NULL“) oder zu einer spez. Komponente gehören (in diesem Falle verweist „Component“ auf die entsprechende Komponente).

- „2“ signalisiert die Metasprachebene und damit Strukturen des Marktplatzes (z. B. Relationen, Attribute, Integritätsbedingungen etc.) – Sprachkonstrukte dieser Ebene beziehen sich nicht auf Komponenten und besitzen demgemäß im Feld „Component“ den Wert „NULL“.

„Term_PredicationRules“		<i>Prädikatorenregeln zur Definitionen der Termini</i>			
#	Attribut		Schlüsseleigenschaften		Integritätsbedingungen
	Name	Typ	PK	FK	
				Verweis	
1	Term	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Term.ID_Term	
2	ID_PredicationRuleOfTerm	Int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	PredicationRule	VarChar	<input type="checkbox"/>	<input type="checkbox"/>	<i>well-formed formula; die referenzierten Fachbegriffe müssen existieren</i>
Integritätsbedingungen:					
➤ <i>ID_PredicationRuleOfTerm sollte innerhalb jedes Terminus fortlaufend verwendet werden</i>					

Anmerkungen:

- verwendete Fachbegriffe: Term, PredicationRule

„Term_ExemplaryUsages“		<i>Definition eines Terms über Einsatzbeispiele</i>			
#	Attribut		Schlüsseleigenschaften		Integritätsbedingungen
	Name	Typ	PK	FK	
				Verweis	
1	Term	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Term.ID_Term	
2	ID_ExemplaryUsageOfTerm	Int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	ExemplaryUsage	VarChar	<input type="checkbox"/>	<input type="checkbox"/>	
Integritätsbedingungen:					
➤ <i>ID_ExemplaryUsageOfTerm sollte innerhalb jedes Terminus fortlaufend verwendet werden</i>					

Anmerkungen:

- verwendete Fachbegriffe: Term, ExemplaryUsage

„Term_LongDefinition“		Ausführliche Definitionen der Termini				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Term	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	
2	LongDefinition	CLOB	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Term, LongDefinition

„Term_Relationships“		Beziehungen zw. verschiedenen Termini				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK		
				Verweis		
1	Term1	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	
2	Term2	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Term.ID_Term	
3	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen:						
➤ <i>Term1 ≠ Term2</i>						

Anmerkungen:

- verwendete Fachbegriffe: Term, Term_Relationship
- Beziehungen zwischen Termini sind weder zwingend symmetrisch noch antisymmetrisch.
- Beispiele für mögliche Beziehungen sind „ist Spezialfall von“, „ist äquipollent zu“, „ist synonym zu“, „enthält“ etc.

„SentenceStructure“		Satzbaupläne für die Beschreibung von Komponentenaufgaben				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_SentenceStructure	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	RelationType	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SentenceStructure_RelationType. ID_RelationType	
3	PropositionalForm	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>Platzhalter werden durch „_“ gekennzeichnet</i>
4	NoOfPlaceholders	Int	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen:						
➤ „NoOfPlaceholders“ muss der Anzahl der Platzhalter in „PropositionalForm“ entsprechen						

Anmerkungen:

- verwendete Fachbegriffe: SentenceStructure, RelationType, PropositionalForm, Placeholder

„SentenceStructure_RelationType“		Baumstruktur der Satzbauplanarten				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_RelationType	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
3	FatherRelationType	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SentenceStructure_RelationType. ID_RelationType	<i>optional</i>
Integritätsbedingungen:						
➤ es sollte genau ein Tupel ohne Eintrag in „FatherRelationType“ existieren						

Anmerkungen:

- verwendete Fachbegriffe: RelationType
- „FatherRelationType“ ermöglicht eine hierarchische Strukturierung der Beziehungstypen.

Es existieren keine interrelationalen Integritätsbedingungen in DBNorm.

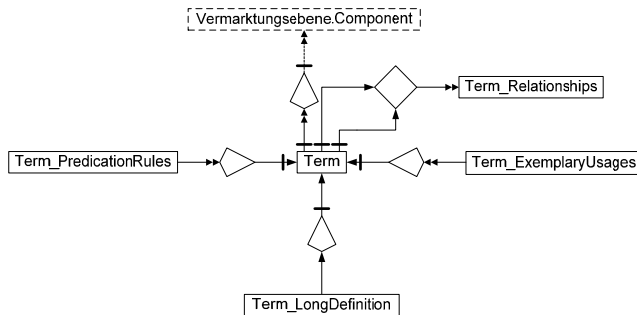


Abbildung 72: Objektypendiagramm DBNorm – Term

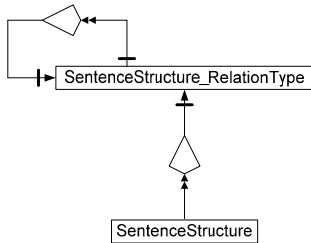


Abbildung 73: Objektypendiagramm DBNorm – Sentence

A.7.7 DBError – Fehler-Datenbank

Ziel von DBError ist die zentralisierte Administration sämtlicher potenziell möglicher Fehlerereignisse inkl. Gliederung in ihrerseits baumstrukturartig festgehaltenen Fehlerklassen. DBError ist noch als „explorativ“ und somit ausbaufähig einzustufen.

Relationenschemata

„Errors“		Mögliche Fehlercodes des Systems inkl. Zuordnung zur jeweiligen Fehlerklasse				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_Error	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	ErrorClass	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ErrorClasses.ID_ErrorClass	
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		Unique
4	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Error, ErrorClass

„ErrorClasses“		<i>Hierarchische Strukturierung der möglichen Fehlerklassen</i>				
#	Attribut		Schlüsseleigenschaften			Integritätsbedingungen
	Name	Typ	PK	FK	Verweis	
1	ID_ErrorClass	Serial	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Father	Int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ErrorClasses.ID_ErrorClass	
3	Name	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		<i>Unique</i>
4	Description	VarChar	<input type="checkbox"/>	<input type="checkbox"/>		
Integritätsbedingungen: (keine)						

Anmerkungen:

- verwendete Fachbegriffe: Error, ErrorClass
- „FatherClass“ ermöglicht eine hierarchische Strukturierung der Fehlerklassen.

Es existieren keine interrelationalen Integritätsbedingungen in DBError.

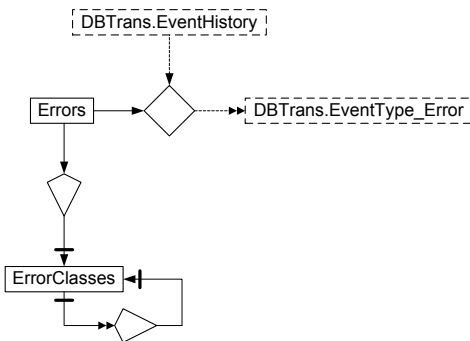


Abbildung 74: Objekttypendiagramm DBError

A.7.8 Gesamtdiagramm

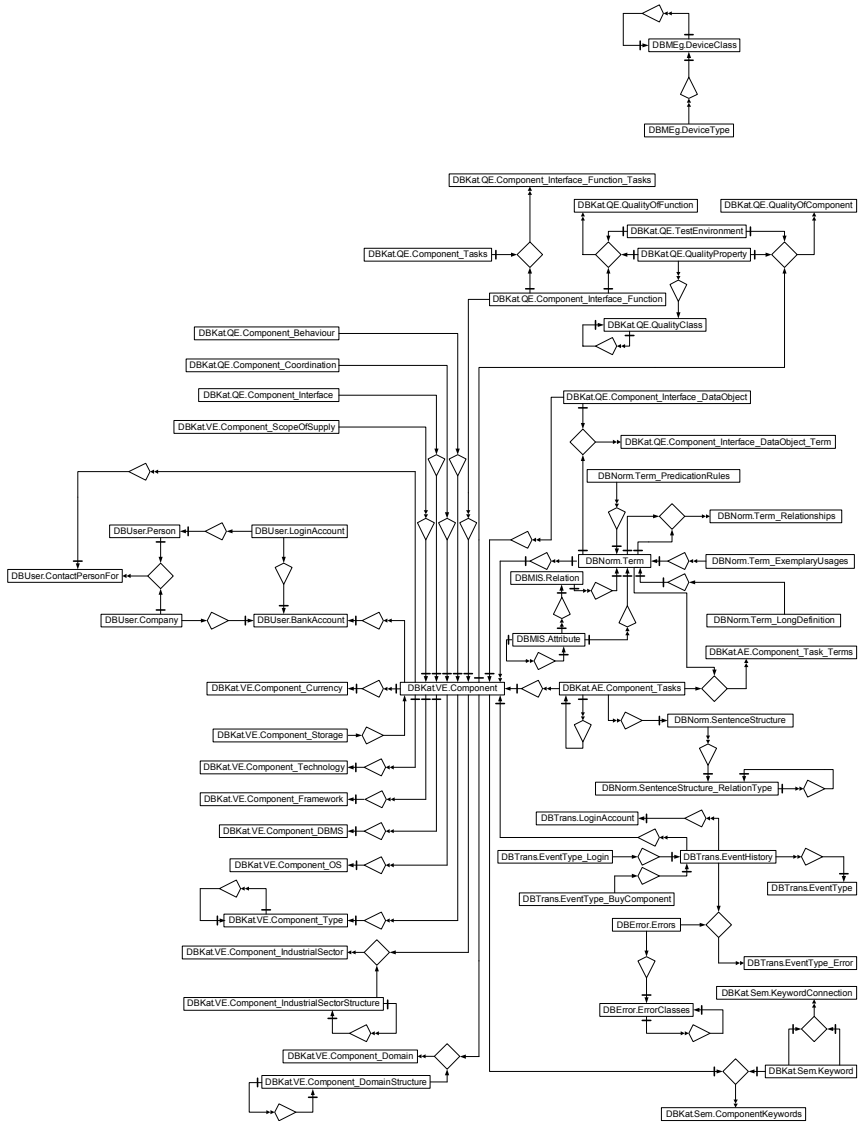


Abbildung 75: Objekttypendiagramm mobiCOMP