

Konzeption eines Frameworks zur automatisierten Erstellung nutzerspezifischer IT-Systemdokumentationen

Robert Krawatzek, Frieder Jacobi, André Müller, Marcus Hofmann

Technische Universität Chemnitz

Fakultät Wirtschaftswissenschaften, Professur Wirtschaftsinformatik II

Reichenhainer Str. 70, D-09126 Chemnitz

Abstract

Die Komplexität von BI-Systemen als heterogene IT-Systeme steigt in Folge einer immer dynamischeren Unternehmensumwelt zunehmend. Softwaredokumentationen helfen, trotz steigender Systemkomplexität einen detaillierten Überblick über die Arbeitsweise von BI-Systemen zu erhalten. Der Erstellungsprozess von qualitativ hochwertigen Dokumentationen ist jedoch aufwändig und wird – unter Vernachlässigung nachträglicher Änderungen und Anpassungen – meist nur in der Planungs- oder Entwicklungsphase von BI-Systemen vorgenommen. Dieser Beitrag beschreibt den Entwurf eines Frameworks, welches durch die Anwendung des Ansatzes der modellgetriebenen Softwareentwicklung eine vollständig automatisierte Erstellung von nutzerspezifischen Softwaredokumentationen erlaubt, von denen sowohl BI-Systementwickler als auch Fachanwender profitieren.

1 Problemidentifizierung und Motivation

Der Begriff Business Intelligence (BI) beschreibt einen analytischen Prozess, der die Sammlung, Aufbereitung und Präsentationen von unternehmens- und marktbezogenen Daten unter Verwendung mathematischer Modelle und analytischer Methoden unterstützt, dessen Ergebnisse in ökonomische Entscheidungsprozesse einfließen. Softwarelösungen, die im BI-Umfeld zum Einsatz kommen, werden als Business Intelligence Systeme (BI-Systeme) (Vercellis, 2009) bezeichnet.

Durch die Verwendung von BI-Systemen bei taktischen und strategischen Unternehmensentscheidungen entsteht ein hoher Anspruch auf Korrektheit der präsentierten Informationen. Fehler und daraus resultierende Fehlentscheidungen können weitreichende Folgen haben.

Die Anforderungen an ein BI-System sind durch ein komplexes und dynamisches Unternehmensumfeld geprägt und unterliegen daher im Verlauf der Zeit Veränderungen. Das führt zu einem permanenten Anpassungsprozess des Systems, der von den Datenquellen über das zentrale Data Warehouse und Data Marts bis hin zu den Berichten reicht. Dieser Anpassungsprozess kann durch geeignete Systemdokumentationen positiv beeinflusst werden.

Die Dokumentation von Softwaresystemen hat die Aufgabe, deren Arbeitsweise – sowohl aus einer technischen als auch aus einer Endanwendersicht – zu beschreiben (K. C. Laudon & J. P. Laudon, 2010). Wie bei anderen Software-Produkten verringert sich ohne eine ausreichende und qualitativ gute Dokumentation auch bei einem BI-System dessen Nutzen, denn es lässt sich unter Umständen nur schlecht oder gar nicht beurteilen, ob die Software den Anforderungen genügt (Wallmüller, 2001, S. 150).

Dokumentationen, die in frühen Phasen eines Softwareprojekts, wie dem Design, der Konzeption oder der Entwicklungsphase erstellt und nach Änderungen nicht aktualisiert wurden, veralten, was letztendlich zu einem sinkenden Nutzen der Dokumentationen führt (Forward & Lethbridge, 2002; Hofmann, Gluchowski, Jacobi, & Kurze, 2011; Wallmüller, 2001). Weitere Ursachen für eine unzureichende Dokumentation bilden qualitative Mängel, wie schlecht lesbare oder schwer zu verstehende Inhalte, was in gegenwärtigen Softwareprojekten häufig vorkommt (Wallmüller, 2001, S. 149). Eine vollständig veraltete, nicht sinnvoll nutzbare Dokumentation entspricht dem Fehlen einer Dokumentation. Re-Dokumentation versucht dieses Problem zu lösen, indem es die nachträgliche Erzeugung einer Dokumentation von existierenden Systemen, für die keine Dokumentation verfügbar ist, vorschlägt (Chikofsky & Cross, J. H., 1990, S. 15).

BI-Systeme sind heterogen aufgebaut und bestehen aus mehreren, meist unterschiedlichen Systemen, was die Nachvollziehbarkeit und Transparenz ohne eine ausreichende Dokumentation im Vergleich zu monolithischen Softwaresystemen zusätzlich erschwert. Gerade in einem derart komplexen System ist es wichtig, eine gute Dokumentation bereitzustellen, um das Vertrauen in das System zu stärken. Steht bei monolithischen Systemen die Dokumentation der Systeme *selbst* im Vordergrund, so existieren bei heterogenen Systemen weitere Anforderungen an den Umfang einer Dokumentation: neben der jeweiligen Systembeschreibung der *einzelnen beteiligten* Systeme wird auch eine Beschreibung der *statischen Gesamtarchitektur* sowie der zwischen den beteiligten Systemen existierenden *Datenflüsse* benötigt.

Der Umstand, dass eine Dokumentation im Kontext von BI-Systemen häufig nur einmalig beim Entwurf oder der Entwicklung stattfindet, birgt die Gefahr einer mangelnden Dokumentation. Spätere Dokumentationsanpassungen verursachen oft einen hohen Auf-

wand (Hofmann et al., 2011). Ein Praxisprojekt bei einem großen deutschen Tourismusanbieter zeigte, dass es Probleme mit der Dokumentation des BI-Systems geben kann, wenn „beim Aufbau von BI-Systemen häufig sehr spontan vorgegangen und eine konzeptionelle Systemplanung zugunsten eines iterativen Prototyping eher vernachlässigt“ wird (Gluchowski & Kurze, 2010, S. 672). Es wurde ebenfalls gezeigt, dass sich mit Hilfe von automatisierten Dokumentationsprozessen der Aufwand für die Erstellung von Dokumentationen von Datenstrukturen (Spezifika der Datenhaltungskomponenten) reduzieren lässt (Gluchowski & Kurze, 2010, S. 676).

Die in diesem Paper dargestellte Forschungsarbeit zielt auf die Erhöhung des Nutzens eines BI-Systems durch die Bereitstellung aktueller und qualitativ hochwertiger Dokumentationen. Dadurch können sowohl die verschiedenen Adressaten – Fachanwender, Entwickler und Manager (Pomberger & Blaschek, 1996) – als auch das durch das BI-System unterstützte Unternehmen wie folgt profitieren.

Fachanwender erhalten Unterstützung bei der *sicheren und funktionsgerechten Handhabung* des Systems. Das wird durch eine hohe Transparenz bezüglich der Datenqualität, d.h. der Herkunft, Verarbeitung und Aktualität der durch das System bereitgestellten Informationen ermöglicht.

Entwickler erhalten Informationen über den aktuellen technischen Aufbau, um das System kennenzulernen und beurteilen zu können. Das ermöglicht Entwicklern ein schnelleres und leichteres Umsetzen von *Änderungen und Erweiterungen*. Dem Entwickler hilft eine sowohl detaillierte als auch vollständige Dokumentation über alle Teilbereiche des Systems, wobei verschiedene Abstraktionsstufen (beispielsweise – in steigender Abstraktionsstufe – Attribut, Tabelle, Datenbank, OLTP-System) zur Bewahrung der Übersichtlichkeit beitragen.

Manager erhalten Informationen aus organisatorischer, kalkulatorischer und führungs-politischer Sicht. Die von Ihnen genutzte Projektdokumentation beinhaltet im Wesentlichen Projekt- und Organisationspläne, Berichte, Ressourceneinsatz sowie Zieldefinition und -erreichung. Das ermöglicht die Auswertung des Projektes im Hinblick auf die *bessere Durchführung zukünftiger Projekte*.

Unternehmen profitieren von der automatisierten Erstellung von Dokumentationen durch *Kosteneinsparungen*:

- durch den Wegfall von hohem manuellen Aufwand und der damit verbundenen Zeit bei der Erstellung von Dokumentationen,

- durch eine verbesserte Qualität und daraus resultierender zusätzlich gewonnen Zeit, die andernfalls benötigt wird, um mittels einer mangelhaften Dokumentation das System zu verstehen (Chikofsky & Cross, J. H., 1990, S. 17) – insbesondere neue Mitarbeiter bekommen die Möglichkeit, sich schneller in die Systeme einzuarbeiten um mit diesen produktiv zu arbeiten,
- da Expertenwissen explizit vorliegt und beim Ausscheiden von Wissensträgern keine mit hohem Aufwand zu füllende Wissenslücke verbleibt.

Nach der einleitenden Diskussion der Problemstellung und der Motivation für konfigurierbare, automatisch zu erstellende Dokumentationen werden im Folgenden erst die Zielvorgaben einer möglichen Lösung aufgestellt. Dem anschließend vorgestellten Entwurf eines Frameworks folgen schlussendlich eine Zusammenfassung der wichtigsten Punkte und eine Ausführung des weiteren Forschungsvorgehens.

Bei dem vorliegenden Beitrag handelt es sich um Research-in-Progress, wobei sich zur Sicherstellung der wissenschaftlichen Rigorosität an den Design Science Forschungsprozess von Peffers et al. (2007) in Kombination mit den Handlungsanweisungen für die einzelnen Forschungsaktivitäten von Hevner et al. (2004) gehalten wird.

2 Definition der Zielvorgaben

(Wallmüller, 2001, S. 151) empfiehlt, sich bei der Erstellung von qualitativ guten Dokumentationen an den vorgeschlagenen Qualitätsmerkmalen für Dokumente zu orientieren. Demnach sollte eine hochwertige Dokumentation folgenden acht Merkmalen genügen (Dt. Ges. für Qualität e.V. u.d. Nachrichtentechn. Ges. im VDE (NTG), 1995; zitiert nach Wallmüller, 2001, S. 151): Änderbarkeit, Aktualität, Eindeutigkeit, Identifizierbarkeit, Normkonformität, Verständlichkeit, Vollständigkeit und Widerspruchsfreiheit.

Mit dem Ziel, den Dokumentationsbedarf von verschiedenen Anspruchsgruppen zu erfüllen, ist darauf zu achten, die Dokumentationen entsprechend den Interessen der verschiedenen Gruppen auszurichten (Wallmüller, 2001, S. 153). Um eine hochqualitative, zielgruppengerechte Dokumentation bereitstellen zu können, erfolgt zuerst eine Unterteilung der Dokumentqualitätsmerkmale in zielgruppenabhängige und zielgruppenunabhängige Merkmale.

Unter dem Begriff *Dokumentattribute* fassen Forward & Lethbridge (2002, S. 26) Informationen über Dokumente zusammen, welche über den reinen Dokumentinhalt hinausgehen. Beispiele für solche Attribute sind das Dokumentformat, der Schreibstil, die Dokumentenstruktur und die Länge einzelner Artikel. Um den größtmöglichen Nutzen zu erhalten, sind diese zielgruppengerecht anzupassen. Die Qualitätsmerkmale *Eindeutigkeit*,

Verständlichkeit und *Vollständigkeit* werden als Dokumentattribute, welche zielgruppenabhängig sind, aufgefasst und entsprechend unterschiedlich gestaltet.

Änderbarkeit, *Aktualität*, *Identifizierbarkeit*, *Normkonformität* und *Widerspruchsfreiheit* sind Qualitätsmerkmale die nicht an individuelle Bedürfnisse der Zielgruppen gebunden sind und sollten stets im gleichen Maße erfüllt sein. Analog zu dem Begriff der Dokumentattribute, werden diese zielgruppenunabhängigen Dokumentanforderungen im Folgenden unter *Erstellungsprozessattribute* zusammengefasst.

Unter Berücksichtigung der Wirtschaftlichkeit bei der Erstellung und Pflege von qualitativ hochwertigen Dokumentationen ergibt sich folgendes Forschungsziel:

Entwicklung einer Methode zum kostengünstigen Erstellen hochqualitativer und nutzerspezifischer BI-Systemdokumentationen.

Zur Erreichung dieses Ziels werden im weiteren Verlauf die folgenden Forschungsfragen untersucht:

1. Wie lassen sich die Anforderungen der Erstellungsprozessattribute (Qualität der Dokumentation) bei der Dokumentationserstellung kostengünstig erfüllen?
2. Wie lassen sich die Anforderungen der Dokumentattribute (nutzerspezifische Dokumentation) bei der Dokumentationserstellung kostengünstig erfüllen?

3 Entwurf und Entwicklung

Wie bereits dargestellt ist die manuelle Erstellung von Dokumentationen zeitintensiv und fehleranfällig. Durch einen höheren Automatisierungsgrad lässt sich der manuelle Aufwand jedoch reduzieren (Forward & Lethbridge, 2002, S. 28). Die zu dokumentierenden Systeme und Teilsysteme beinhalten bereits einen großen Teil der für die Erstellung einer Dokumentation nötigen Informationen in Form von Quelltext¹ (Forward & Lethbridge, 2002, S. 30).

Modellgetriebene² Ansätze für die Entwicklung von Data Warehouses (Kurze, 2011; Mazón & Trujillo, 2008) beinhalten – über die im Quelltext enthaltenen Informationen

¹ In BI-Systemen entspricht der Quelltext, der üblicherweise in der niedrigsten Abstraktionsstufe angeordnet ist, beispielsweise Datenbankschemata.

² Innerhalb der modellgetriebenen Softwareentwicklung ist die modellgetriebene Architektur (Model-Driven Architecture, MDA) ein von der Object Management Group (OMG) standardisierter Ansatz, der auf der strikten Trennung zwischen Spezifikation und Implementierung eines Systems basiert (Object Management Group, 2003). Dieses Prinzip wird durch die Einführung verschiedener Abstraktionsebenen – berechnungsunabhängige Modelle (Computation Independent Model, CIM), plattformunabhängige Modelle (Platform Independent Model,

hinaus – weitere, für die Erstellung von nutzerspezifischen Dokumentationen erforderliche Informationen in Form von Metadaten über die Systeme, abgelegt in den Modellen. Durch die automatisierte Erzeugung einer Dokumentation direkt aus den vorliegenden Modellen wird ein Großteil der angestrebten Erstellungsprozessattribute erfüllt:

- *Änderbarkeit* und *Aktualität*: der Wegfall von manuellen Aufwänden erlaubt es, Systemänderungen jederzeit durch eine Neugenerierung der Dokumentation zu veröffentlichen;
- *Identifizierbarkeit*: durch die zur Verfügung stehenden Metadaten ist es möglich, Zusammenhänge zwischen einzelnen Objekten, die im Quelltext u.U. nicht abbildbar sind, in die Dokumentation einfließen zu lassen;
- *Widerspruchsfreiheit*: durch die automatische Erzeugung von Dokumentationen aus den Modellen wird in der Dokumentation genau der Grad an Widerspruchsfreiheit abgebildet, der in den Modellen vorliegt.

Im Folgenden wird dargestellt, wie sich diese Metadaten erweitern und verknüpfen lassen, um eine konfigurierbare Erstellung von Dokumentationen zu ermöglichen. Um die Dokumentation nutzerspezifisch zu generieren, sind die bereits vorhandenen Modelle um eine Möglichkeit der Konfiguration bei der Verknüpfung von Informationen zu erweitern. Diese Konfiguration sollte in das modellgetriebene Vorgehen eingebettet werden, also in Form von Modellen vorliegen. Dieser Ansatz erlaubt die Konfiguration der Nutzerzugehörigkeit zu einem Fachbereich und damit den bereitzustellenden Inhalt (beeinflusst *Vollständigkeit* und *Eindeutigkeit*), des Layouts der Dokumentation (beeinflusst *Verständlichkeit*) und des Ausgabeformats (beeinflusst *Eindeutigkeit*).

Die bisher vollzogenen Überlegungen führen zur Konzipierung des folgenden Frameworks.

3.1 Konzept des Frameworks

Dieser Abschnitt stellt das Konzept eines Frameworks zur automatisierten Dokumentation von BI-Systemen vor, welches Abbildung 1 schematisch darstellt. Nachfolgend werden die Komponenten des Frameworks in Richtung des Datenflusses erläutert.

BI-Systemlandschaft

Die BI-Systemlandschaft kann aus einer Vielzahl verschiedener Systeme bestehen, die je nach Architektur operative Systeme, ETL-Prozesse, Datenhaltungs- und Berichtssysteme enthält. Zusätzlich beinhaltet die BI-Systemlandschaft auch das Architekturmodell, d.h. die Einbettung der Systeme in die Gesamlandschaft, und Beschreibungen über Datenflüsse und Prozesse zwischen den Teilsystemen. Diese Aspekte werden im Folgenden unter dem Begriff „Dokumentationssubjekte“ zusammengefasst, die wiederum selbst aus einer Vielzahl kleinerer Entitäten bestehen, die im Folgenden „Bausteine“ genannt werden.

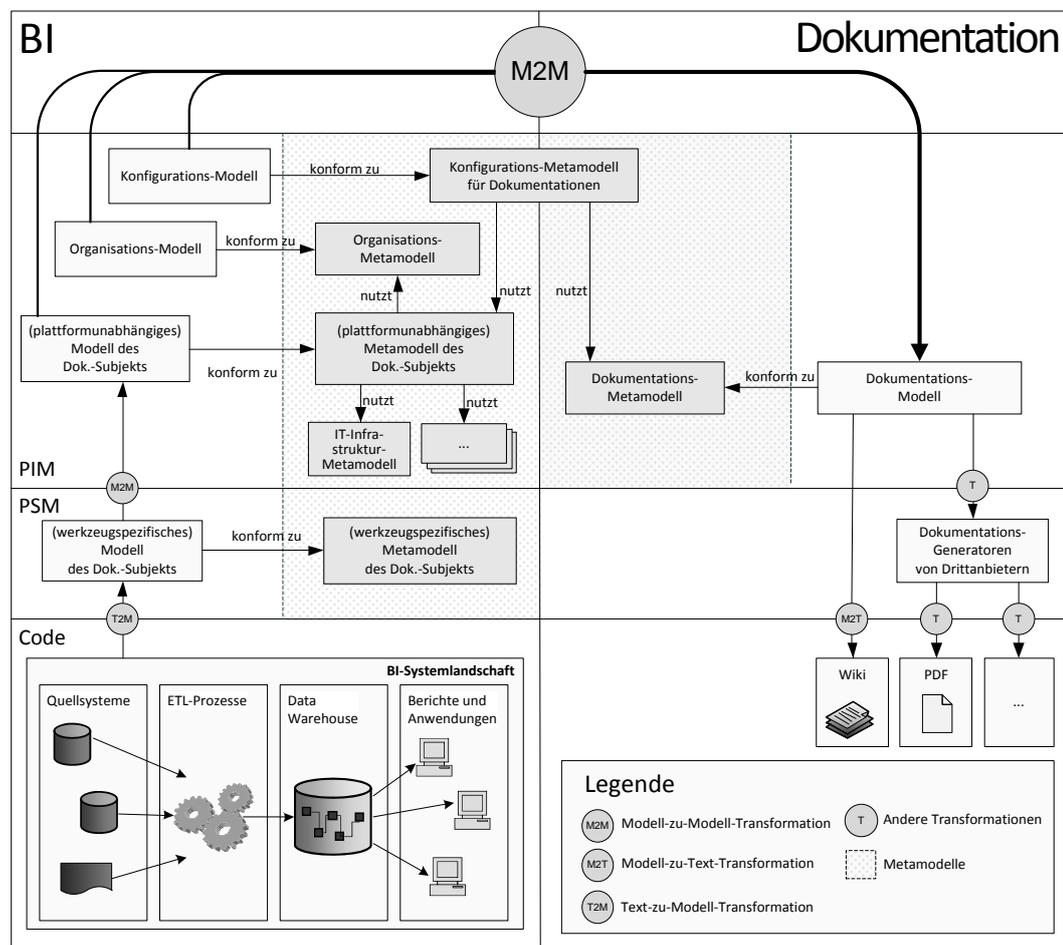


Abbildung 1: Framework zur automatisierten Erstellung nutzerspezifischer BI-Systemdokumentationen

Werkzeugspezifische Metamodelle

BI-Werkzeuge bieten üblicherweise eine Möglichkeit zum Speichern und Laden der modellierten Komponenten, die als Dokumentationsobjekte dienen können. In den meisten Fällen entspricht das Ausgabeformat jedoch keinem Standard wie beispielsweise dem Common Warehouse Model (CWM), sondern basiert auf einem proprietären Format. Dem modellgetriebenen Ansatz folgend ist deshalb für jedes singuläre Format ein Metamodell zu erstellen, welches die jeweiligen Eigenheiten des Werkzeugs abbildet und somit für die Verwendung im Dokumentations-Framework kapselt. Für die Transformation in ein dem Metamodell entsprechendes Modell muss das proprietäre Format gelesen und semantisch korrekt interpretiert werden können. Abhängig von der Art des Formats kann das zu zusätzlichen Entwicklungsaufwänden führen. Im Ergebnis entsteht eine Menge von Metamodellen, von denen jedes spezifisch für ein konkretes Werkzeug und ein damit modellierbares Dokumentationsobjekt ist.

Werkzeugunabhängige Metamodelle

Wie im vorhergehenden Abschnitt dargestellt, kann für jedes Dokumentationsobjekt eine Vielzahl von werkzeugabhängigen Metamodellen existieren. Diese Metamodelle sind naturgemäß nicht kompatibel, da jedes Werkzeug – trotz gleicher Dokumentationsobjekte – verschiedene Attribute und Modellierungsmöglichkeiten bietet. Eine unmittelbare Transformation dieser Artefakte zu einer Dokumentation führte zu einer großen Anzahl von ähnlichen Transformationen, was wiederum einen großen Wartungsaufwand und eine hohe Fehleranfälligkeit bei Anpassungen mit sich brächte. Dem an MDA angelehnten Ansatz weiter folgend werden daher werkzeugunabhängige Metamodelle verwendet, die den Inhalt von Dokumentationsobjekten auf einer konzeptionellen Ebene zusammenführen.³ Werkzeugspezifische Modelle werden dann mit Hilfe von Modell-zu-Modell-Transformationen zunächst in eine werkzeugunabhängige Darstellung überführt. Die werkzeugunabhängigen Modelle bilden somit den Ausgangspunkt für weitere Transformationen.

Transformation aus werkzeugunabhängigen Modellen zur Dokumentation

Das werkzeugunabhängige Modell eines Dokumentationsobjekts enthält dessen vollständige Beschreibung auf konzeptioneller Ebene. Um eine für Nutzer wertvolle Dokumentation zu erstellen, sind jedoch weitere Verarbeitungsschritte notwendig. In Abhän-

³ Darüber hinaus lassen sich mit Hilfe dieser Metamodelle Beziehungen der Dokumentationsobjekte untereinander ausdrücken, was – insbesondere bei der Verwendung von Werkzeugen verschiedener Anbieter – auf werkzeugspezifischer Ebene nur selten möglich ist.

gigkeit des Fachbereichs und der Rolle des Adressaten der Dokumentation im Unternehmen müssen die Informationen unterschiedlich gefiltert und strukturiert werden.

Die dafür notwendigen Daten können nicht aus der Beschreibung des Dokumentationsobjekts gewonnen werden, sondern müssen auf anderem Wege bereitgestellt werden. Dafür sind weitere, dokumentationsobjektunabhängige Metamodelle zu konzipieren, deren Instanzen die nötigen Informationen enthalten:

- Ein Organisationsmodell bildet die Organisationsstruktur eines Unternehmens ab, einschließlich von Abteilungen, Nutzern, Nutzerrollen und deren Beziehungen. In Kombination mit einer Beschreibung der IT-Landschaft lassen sich Zuordnungen zwischen Bausteinen der Dokumentationsobjekte und der Einordnung des Adressaten im Unternehmenskontext bilden, was es ermöglicht, die Inhalte adressatengerecht zu filtern.
- Um die durch die Dokumentattribute spezifizierten Anforderungen der Nutzerrollen an die Dokumentation gerecht zu werden, wird im vorgestellten Konzept ein Konfigurations-Modell verwendet, das es ermöglicht, spezifische Bausteine (beispielsweise Tabellen in einer Beschreibung von Datenbankschemata) bestimmten Dokumentationskonzepten (wie beispielsweise Kapitel, Abschnitte oder Diagramme) in einem dem Informationsbedarf angepassten Detaillierungsgrad zuzuordnen.

Die strukturierte Kombination des werkzeugunabhängigen Modells mit den eben beschriebenen Modellen ermöglicht nun die Erzeugung von adressatengerechten und damit hochwertigen Inhalten.

Dokumentations-Metamodell und Ausgabeformate

Dem Ziel des hier vorgestellten Frameworks nach einer hochgradig flexiblen und gut anpassbaren Lösung folgend ergibt sich die Notwendigkeit zur Unterstützung verschiedenartiger Ausgabeformate. Analog zu dem oben beschriebenen Problem des Mehraufwands bei der Umsetzung durch direkte Transformationen aus den beschriebenen Modellen zu verschiedenen Ausgabeformaten wird auch hier eine plattformunabhängige (also ausgabeformatunabhängige) Abstraktionsebene etabliert. In diesem Dokumentationsmodell liegen die entsprechend aufbereiteten Inhalte in einer dokumentenähnlichen Struktur vor. Dieses Modell lässt sich mit Hilfe von Transformationen einerseits direkt zu verschiedenen Ausgabeformaten (wie beispielsweise einem Mediawiki) oder mittelbar unter Ausnutzung existierender Dokumentationsgeneratoren wie z.B. DITA (DITA, 2011) oder

DocBook (DocBook, 2011) zu den von diesen Drittanbietern unterstützten Formaten transformieren.

4 Zusammenfassung und Ausblick

Der vorliegende Beitrag präsentierte die konzeptionelle Entwicklung eines Frameworks zur automatisierten Erstellung von qualitativ hochwertigen, nutzerspezifischen Software-dokumentationen. Eine praktische Umsetzung dieses Ansatzes verspricht eine signifikante Reduktion des Erstellungs- und Pflegeaufwands von Software-dokumentationen. Weiter zeichnen sich die so erstellten Dokumentationen durch einen hohen Grad an Aktualität aus, da die Anwendung des Paradigmas der modellgetriebenen Softwareentwicklung eine einfache Integration des Dokumentationsprozesses in den Wartungsprozess von BI-Systemen gestattet. Die vollständig automatische Erzeugung von nutzerspezifischen Dokumentationen führt ferner zur Vermeidung manueller Eingriffe sowie Medienbrüche und schließt so potentielle Fehlerquellen aus, was zu qualitativ höherwertigeren Software-dokumentationen führt.

Dem von Peffers et al. vorgeschlagenen Design Science Forschungsprozess folgend, ist als nächstes die Umsetzbarkeit des entwickelten Frameworks zu demonstrieren. Hierfür ist, auf Grundlage des vorgestellten Frameworks, die Entwicklung eines Prototypens für automatisierte, konfigurierbare ETL-Prozess-Dokumentation geplant. Dieser Prototyp stellt weiterhin die Grundlage für die ausstehende Evaluation des Dokumentationsframeworks dar. Der Nachweis der Reduktion des Erstellungs- und Pflegeaufwandes von Dokumentationen geschieht im Rahmen einer Pilotierung (Österle et al., 2010, S. 668). Die in diesem Zusammenhang anfallenden Dokumentationen bilden zusammen mit dem Standard „Systems and software engineering – Requirements for designers and developers of user documentation“ (ISO-IEC 26514; International Organisation for Standardization, 2008) den Ausgangspunkt zum Nachweis des noch ausstehenden Dokumentqualitätsmerkmals der Normkonformität.

5 Anmerkungen

Das „Computer-Aided Warehouse Engineering“ (CAWE) Projekt, in dessen Rahmen der vorliegende Beitrag entstanden ist, wird mit Mitteln des ESF und des Freistaates Sachsen gefördert.

Literaturverzeichnis

- Chikofsky, E. J., & Cross, J. H., I. (1990). Reverse engineering and design recovery: a taxonomy. *IEEE Software*, 7(1), 13-17.
- DITA (2011). <http://www.oasis-open.org/committees/dita>.
- DocBook (2011). <http://www.oasis-open.org/docbook>.
- Forward, A., & Lethbridge, T. C. (2002). The Relevance of Software Documentation, Tools and Technologies: a Survey. *Proceedings of the 2002 ACM symposium on Document engineering - DocEng'02* (pp. 26-33). New York, USA: ACM Press.
- Dt. Ges. für Qualität e.V. u.d. Nachrichtentechn. Ges. im VDE (NTG). (1995). *Software-Qualitätssicherung : Aufgaben, Möglichkeiten, Lösungen*. (2nd ed.). VDE-Verlag.
- Gluchowski, P., & Kurze, C. (2010). Modellierung und Dokumentation von BI-Systemen. *CONTROLLING - Zeitschrift für erfolgsorientierte Unternehmensplanung*, 22, 676-682.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75-105.
- Hofmann, M., Gluchowski, P., Jacobi, F., & Kurze, C. (2011). Computer-Aided Warehouse Engineering: Dokumentation und Modellierung komplexer Data-Warehouse-Systeme. *Presentation slides of the 11th European TDWI Conference with BARC@TDWI-Track*.
- International Organisation for Standardization. (2008). *Systems and software engineering -- Requirements for designers and developers of user documentation (ISO-IEC 26514)*.
- Kurze, C. (2011). *Computer-Aided Warehouse Engineering: Anwendung modellgetriebener Entwicklungspfadparadigmen auf Data-Warehouse-Systeme*. Verlag Dr. Kovač.
- Laudon, K. C., & Laudon, J. P. (2010). *Management information systems: managing the digital firm* (11th ed.). Upper Saddle River, New Jersey: Pearson.
- Mazón, J.-N., & Trujillo, J. (2008). An MDA approach for the development of data warehouses. *Decision Support Systems*, 45(1), 41-58.
- Object Management Group. (2003). MDA Guide V1.0.1. Retrieved from <http://www.omg.org/cgi-bin/doc?omg/03-06-01>.
- Peppers, K., Tuunanen, T., Rothenberger, M. a, & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45-77.

- Pomberger, G., & Blaschek, G. (1996). *Software Engineering: Prototyping und objektorientierte Software-Entwicklung* (2nd ed.). Carl Hanser Verlag.
- Vercellis, C. (2009). *Business intelligence: data mining and optimization for decision making*. Hoboken, N.J.: Wiley.
- Wallmüller, E. (2001). Die Rolle der Dokumentation in Software-Projekten. *Software-Qualitätsmanagement in der Praxis: Software-Qualität durch Führung und Verbesserung von Software-Prozessen* (2nd ed., pp. 149-156). Hanser Fachbuch.
- Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., et al. (2010). Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. *Angewandte Informatik*, 62(6), 664-672.