Karlsruhe Reports in Informatics 2016,6
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Studierendenprogramm der Fachtagung „Modellierung 2016“

Herausgeber:
Robert Heinrich und Rainer Neumann

2016

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Studierendenprogramm der Fachtagung „Modellierung 2016“

Robert Heinrich[1], Rainer Neumann[2]

Im Jahr 2016 bietet die Fachtagung „Modellierung“ zum ersten Mal ein Studierendenprogramm an. Das Studierendenprogramm soll es Studierenden ermöglichen an einer wissenschaftlichen Tagung als Vortragende teilzunehmen. Studierende können dadurch einen Einblick in aktuelle Forschungsthemen erlangen und präsentieren ihre Ergebnisse in einem wissenschaftlichen Umfeld. Das Studierendenprogramm soll den Austausch zwischen Studierenden und Forschern sowie Praktikern aus allen Bereichen der Informatik mit starkem Bezug zu Modellierung fördern. Darüber hinaus ermöglicht das Studierendenprogramm das Kennenlernen von Forschenden und Lehrenden aus anderen Hochschulen.

Die Fachtagung „Modellierung“ wird vom Querschnittsfachausschuss Modellierung der Gesellschaft für Informatik e.V. seit 1998 durchgeführt und hat sich als einschlägiges Forum für Grundlagen, Methoden, Techniken, Werkzeuge sowie Domänen und Anwendungen der Modellierung etabliert. Die „Modellierung“ führt Teilnehmerinnen und Teilnehmer aus allen Bereichen der Informatik sowie aus Wissenschaft und Praxis zusammen. Die Tagung zeichnet sich traditionell durch lebendige und fachgebietsübergreifende Diskussionen und engagierte Rückmeldungen aus, weshalb sie gerade auch für Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler interessant ist.

Das diesjährige Studierendenprogramm umfasst drei Beiträge von Studierenden. Dominik Werle berichtet in seinem Beitrag „*A Domain-Specific Language for Model Consistency*“ über einen Ansatz zur Konsistenzhaltung bei Änderungen in verschiedenen Modellen, die überlappende Informationen enthalten. Maximilian Madlung untersucht in seinem Beitrag „*Modellierung von Prozessen und Datenflüssen basierend auf gesetzlichen Vorgaben*“ die Eignung aktueller Prozessnotationen gesetzliche Vorgaben bezüglich des Schutzziels Datensicherheit zu repräsentieren. Ein ähnliches Thema behandelt Michael Junker in seinem Beitrag „*Modellierung von Prozessen und Datenflüssen im Hintergrund des Datenschutzes*“ indem er aktuelle Prozessnotationen anhand des Datenschutz-Prinzips der Zweckbindung untersucht.

[1] Karlsruher Institut für Technologie (KIT), heinrich@kit.edu
[2] Hochschule Karlsruhe, rainer.neumann@hs-karlsruhe.de

A Domain-Specific Language for Model Consistency

Dominik Werle[1]

If there exist different models that contain overlapping information about a system, changes to the information in only one of the models create inconsistencies. It can be infeasible to create a single model that comprises all the information about the system from which views are only projected and where changes are made directly in the underlying model.

Model transformations are used to specify how a model is derived automatically from a related model. In the synchronization case, the model that has been changed is used as a *source* model and all overlapping models are *target* models. If the model transformation engine executes a model transformation in *batch mode*, a *new* target model is created from the source model. In our scenario, targets often do not contains less or equal information than the source and thus cannot be completely derived. If a model transformation is executed *incrementally*, only the parts in the target that are affected by the change in the source model are modified and no information in the target is lost in the process.

For our approach to model synchronization we designed a domain-specific model transformation language, the *mapping language*. General purpose model transformation languages are applicable to a broad spectrum of *model transformation problems* which includes model synchronization, but also e.g. the execution or the combination of models. By tailoring the model transformation language to this specific application scenario, we aim to provide suitable abstractions that may not be possible in arbitrary contexts.

The user of the mapping language can specify simple consistency constraints and consistency repair mechanisms for a pair of meta models in a textual syntax. Each *mapping* consists of a set of *meta classes* and associated constraints for each meta model, forming a *signature*. If a concrete instance of one of the signatures is newly created after a change in one of the models, an appropriate structure in the corresponding model is created by the consistency repair mechanism. Similarily, corresponding structures are destroyed, if a previously valid instance of a signature is not valid anymore. The consistency preservation system keeps track of created instances and their counterparts in appropriate data structure.

Additionally to the patterns for each meta model, a mapping also contains constraints that describe how attribute values of the elements in both models are related to each other, e.g. in the simplest case by being equal to each other. A mapping can also reference an arbitrary number of instances of other mappings and allows the constraints to include already mapped elements for those mappings.

[1] Karlsruhe Institute of Technology (KIT), Institue for Program Structures and Data Organization (IPD), Chair for Software Design and Quality (SDQ), Am Fasanengarten 5, 76131 Karlsruhe, dominik.werle@student.kit.edu

The mappings language is *declarative*: The constraints on the signature elements must both be *checkable* for determining if a set of model elements is a valid instance of a signature, and *enforcable* for creating a new corresponding structure automatically. We provide a set of basic constraints which are compiled to code that checks and enforces them consistently. For example the developer can require the existence of a reference between model elements in a signature. The developer may also manually specify both directions in imperative code, for which they must ensure consistency themselves.

Futhermore, the mappings are *bidirectional*. All mapping specifications are symmetric. Every mapping can be used to derive the code to execute if any of the involved models changes. Attribute values are propagated from the changed signature instances to corresponding structures. We also provide a set of constraints for which we can automatically derive appropriate propagation code, e.g. relation of attribute values. Furthermore the developer may (similarly to signature constraints) specify blocks of imperative code that are used to propagate information for each direction.

For the language design, we inspected different case studies and identified needed functionality with respect to model transformations. We extended our implementation of the basic mapping functionality with both functionality that is necessary for the technical space our approach operates in and features for a more simple and compact specification of both complex and common structures.

Mappings are similar to Triple Graph Grammars (TGG) rules [Sch95]. Since mappings are part of a set of languages for model consistency they are designed to be simpler than TGGs. For example it is not possible to explicitly specify the witness structure (the middle graph of a TGG rule) – however, this makes the mapping specification simpler. Mappings also provide other attribute constraints than current TGG synchronizations.

The mapping language is part of the MIR language (mappings, responses, invariants) which is currently being developed as part of VITRUVIUS (**vi**ew-cen**tr**ic Engineering **u**sing a **vi**rtual **u**nderlying **s**ingle model) [KBL13]. The approach provides a framework for transformations that react to model changes. The MIR language is designed to simplify the specification of such transformations. Additionally to *mappings*, the user can also specify code for checking invariants and reacting to violations to them, and attach code blocks to specific change types using an (imperative) language for reactive programming.

References

[KBL13] M. E. Kramer, E. Burger und M. Langhammer. “View-centric engineering with synchronized heterogeneous models”. In: *Proceedings of the 1st Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling*. VAO ’13. ACM, 2013, 5:1–5:6.

[Sch95] A. Schürr. “Graph-Theoretic Concepts in Computer Science: 20th International Workshop, WG ’94 Herrsching, Germany, June 16–18, 1994 Proceedings”. In: Springer Berlin Heidelberg, 1995. Kap. Specification of graph translators with triple graph grammars, S. 151–163.

Modellierung von Prozessen und Datenflüssen basierend auf gesetzlichen Vorgaben

Maximilian Madlung[1]

Abstract: Gesetze wie es das Recht kennt sind Texte, Prozesse und Datenflüsse hingegen sind meist modelliert. Gesetze müssen in den Geschäftsprozessen allerdings umgegesetzt und validiert werden. Bisher gibt es keinen Standard zur Umsetzung von Gesetzen in Modellen. Diese Arbeit stellt einen Ansatz zur Modellierung des Schutzziels Datensicherheit vor. Dies wird in einem Geschäftsprozess eines Verkaufssystems validiert werden.

Keywords: Modellierung, Datenschutz, Datensicherheit, Bundesdatenschutzgesetz, Geschäftsprozessmodellierung, Validierung.

Immer mehr Bereiche unseres Lebens werden durch digitale Prozesse beeinflusst. Dafür gibt es Gesetze. Die Prozesse so zu entwickeln und danach zu überprüfen ob die Gesetze entsprechend eingehalten werden ist anspruchsvoll. Gesetze sind in natürlicher Sprache verfasst. Die Syntax ist beispielsweise Deutsch. Die Semantik der Gesetze wird von Menschen interpretiert und ist durch viele Referenzen, Fachbegriffe und Formulierungen schwer zu verstehen. In der Informatik gibt es formale Sprachen. Es gibt eine Syntax, es gibt Semantik und es gibt einen deterministischen Interpreter. Zur Beschreibung eines Systems gibt es in der Informatik mehrere Modellierungssprachen. Können Gesetzestexte in einer solchen Modellierungssprache ausgedrückt werden um danach zu überprüfen ob Gesetze eingehalten wurden? Bei jeder Art von Prozzess muss Datenschutz und insbesondere Sicherheit beachtet werden. In der Informatik wird dabei meist von einer Querschittsfunktionalität gesprochen, weil es nicht eine Anforderung ist, die zu einem bestehenden Produkt einfach hinzugefügt werden kann, sondern von Anfang an in jeder Komponente beachtet werden muss. Daher kann die Sicherheit auch nicht einmal implementiert werden, sondern muss im Entwurf des Gesamtsystems vorhanden sein. Dieser Entwurf ist bei professioneller Entwicklung in einer Modellierungssprache festgehalten. Ziel soll es daher sein, die vorher in der Modellierungssprache festgehaltenen Gesetzestexte im Entwurf des Systems nachzuweisen. CIA In der Informatik gibt es für die Datensicherheit respektive Informationssicherheit drei Schutzziele, welche mit CIA abgekürzt werden. CIA steht für Confidentiality, Integrity und Availability. Es soll also die Vertraulichkeit, Integrität und Verfügbarkeit sichergestellt werden. Vertraulichkeit bedeutet, dass nur für die jeweilige Operation auf den Daten entsprechend autorisierte Benutzer diese ausführen können. Im Falle einer Datenkommunikation dürfen beispielsweise nur die beiden Kommunikationspartner die Nachrichten lesen. Integirät stellt sicher, dass auf den Daten keine unbemerkte Operation durchgeführt wird. In unserem Kommunikationsbeispiel bedeutet dies, dass die Nachricht nicht von einem dritten verändert werden kann, ohne dass die anderen Teilnehmer es bemerken. Verfügbarkeit sichert zu, dass ein Dienst funktioniert. Die Datenkommunikationsplattform muss also dauerhaft reagieren, Nachrichten empfangen und verschicken. Das letztere Schutzziel ist im Gegensatz zu den ersteren beiden

[1] Karlsruher Institut für Technologie, Lehrstuhl Software-Design und -Qualität, Am Fasanengarten 5, 76131 Karlsruhe, maximilian.madlung@student.kit.edu

aufgrund von DDOS-Attacken nicht allein algorithmisch zu lösen. Es gibt kein Datensicherheitsrecht. Allerdings gibt es das Bundesdatenschutzgesetz (BDSG), welches sich von der Datenschutzrichtlinie der EU ableitet. Richtlinie bedeutet allerdings, dass das entsprechende im Mitgliedstaat verabschiedete Gesetz, also das Bundesdatenschutzgesetz respektive die entsprechenden Landesdatenschutzgesetze der Bundesländer gelten. Das Ziel des BDSG ist es „den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persöhnlichkeitsrecht beeinträchtigt wird“ (1 BDSG). Anders als in der Informatik, wo technisch jede Information neutral betrachtet und entsprechend gesichert werden kann, findet im Recht eine Wertung statt. Zur Modellierung von Geschäftsprozessen gibt es in der Informatik verschiedene Notationen. Die wohl verbreiteste ist die Business Process Modeling Notation (BPMN). Weiterhin gibt es die Aktivitätendiagramme der Unified Modeling Language (2.0) sowie Petri-Netze. Die Unterschiede der verschiedenen Modelle sind vorallem ihre Anwendungszwecke. So sind Petri-Netze eher für formale Systeme zur Verifizierung eines Verhaltens eines Computers geeignet, UML Aktivitäten Diagramme werden zur Beschreibung von Algorithmen verwendet und BPMN-Diagramme vorallem zur Beschreibung von Geschäftsprozessen aller Art. Insbesondere also auch Geschäftsprozesse die nichts mit Informatik zu tun haben. Im weiteren Verlauf dieser Arbeit wird zunächst auf die Datensicherheit eingegangen, dazu wird erläutert, was im Recht darunter verstanden wird, wie sich die Datensicherheit im Recht einordnet und in welchen Rechtsnormen sie zu finden ist. Danach werden Modellierungen der Informatik vorgestellt, welche genutzt werden sollen um das Beispiel und die Rechtsnormen abzubilden und zusammenzuführen. Im darauffolgenden Kapitel werden die bisherigen Ansätze zur Modellierung von Datenschutzanforderungen vorgestellt. Hierbei werden die eingeführten Modellierungssprachen verwendet. Dann folgt eine Umsetzung der eingeführten Datensicherheit mit den vorgestellten Modellierungstechniken anhand des Beispiels CoCoME (Common Component Modeling Example). Darauf folgend wird diskutiert werlche der vorgestellten Lösungen welche Vorteile bietet. Hier werden insbesondere die Rechtsaspekte des §9 betrachtet. Welche können in welcher vorgestellten Lösung implementiert werden. Schließlich wird resümiert wie die Rechts- und Informatikwelt näher aneinander geführt werden konnten und ein Ausblick auf potentiell zu verfolgende Fragen gegeben. Wichtigste Quellen dieser Arbeit sind [He14], [BM12] und [Bi07].

Literaturverzeichnis

[Bi07] Bizer, Johann: Sieben goldene Regeln des Datenschutzes. Datenschutz und Datensicherheit-DuD, 31(5):355, 2007.

[BM12] Bock, Kirsten; Meissner, Sebastian: Datenschutz-Schutzziele im Recht. Datenschutz und Datensicherheit-DuD, 36(6):427, 2012.

[He14] Heinrich, Robert: Quality Modeling within Business Process Models. In: Aligning Business Processes and Information Systems, S. 59–77. Springer Fachmedien Wiesbaden, 2014.

Modellierung von Prozessen und Datenflüssen im Hintergrund des Datenschutzes

Michael Junker[1]

1 Modellieren von Prozessen und Datenflüssen anhand des Prinzips der Zweckbindung

Nicht nur beim Online-Shopping spielt Datenschutz eine große Rolle. Auch bei der Verwendung von Google, Facebook und Co müssen Datenschutzgesetze von Seiten der Betreiber eingehalten werden. Wichtig dabei ist, dass der Nutzer stets darüber informiert ist, was mit seinen persönlichen Daten geschieht, und auch zu welchem Zweck diese genutzt werden.
Dies wird durch das *Prinzip der Zweckbindung* sichergestellt. Das Prinzip der Zweckbindung ist nach [Bi07] dabei eines von sieben Prinzipien im Datenschutz. Dabei ist Zweckbindung bereits im Bundesdatenschutzgesetz (BDSG) verankert. Dort heißt es beispielsweise in §28 Abs. 1, dass die Zwecke der Erhebung der personenbezogenen Daten konkret festzulegen sind. Vage Vermutungen über den Zweck anzustellen, ist also nicht legal. Weiterhin ist in §4 Abs. 3 BDSG festgelegt, dass der Betroffene über die Verwendungszwecke unterrichtet werden muss. So können beispielsweise keine personenbezogenen Daten an Dritte verkauft werden, ohne dass dies vorher festgelegt und der Betroffene darüber informiert wurde.
Das technische Pendant zum Prinzip der Zweckbindung ist die sogenannte *Nicht-Verkettbarkeit*. Diese ist nach [BM12] so definiert, dass „personenbezogene Daten nicht oder nur mit unverhältnismäßig hohem Aufwand für einen anderen als den ausgewiesenen Zweck erhoben, verarbeitet und genutzt werden können“. Damit wird nicht zuletzt auch das widergespiegelt, dass bereits im BDSG verankert ist, sondern auch im Telemediengesetz steht. Dort heißt es in §15 Abs. 3 unter anderem, dass Anbieter eines Telemediendienstes für bestimmte Zwecke Nutzungsprofile bei Verwendung von Pseudonymen erstellen darf. Diese dürfen jedoch nicht mit den Daten über den Träger des Pseudonyms zusammengeführt werden. Die Nutzungsprofile dürfen also nicht mit den personenbezogenen Daten des Trägers verkettet werden.
Zu den Maßnahmen Nicht-Verkettbarkeit gewährleisten zu können, gehören zum einen die Pseudonymisierung oder Anonymisierung der personenbezogenen Daten aber auch die Trennung von Prozessen und IT-Systemen ([BM12]). Ziel der eigentlichen Arbeit ist es also Nicht-Verkettbarkeit in einem Szenario mit Hilfe dieser Maßnahmen oder entsprechenden Symbolen zu modellieren. Das gewählte Szenario ist der Bestellvorgang von

[1] Karlsruher Institut für Technologie, Lehrstuhl Software-Design und -Qualität, Am Fasanengarten 5, 76131 Karlsruhe, michaeljunker@gmx.net

Waren in einem Online-Shop einer Supermarktkette. Die Waren sind nach Beendigung der Bestellung in einer gewählten Filiale verfügbar und können abgeholt werden. Dieses Szenario stellt eine Erweiterung zu CoCoMe dar ([He15]).
Modelliert wurde dieses Szenario mit den Basisnotationen der Modellierungssprachen *Ereignisgetriebene Prozessketten* (kurz: EPK), *Unified Modeling Language* (kurz: UML) und *Business Process and Modeling Notation* (kurz: BPMN). Für UML wurden Aktivitätsdiagramme zur Darstellung des Szenarios verwendet. Keine dieser Modellierungssprachen bietet eine eigene Symbolik an, um Nicht-Verkettbarkeit darstellen zu können, weswegen nur eine Modellierung der Maßnahmen in Bezug auf das Szenario in Frage kam. Dabei wurde festgestellt, dass über EPK's nicht einmal eine vernünftige Modellierung von Pseudonymisierung möglich ist, da Entitäten in IT-Systemen immer bidirektional verknüpft sind. Dadurch lässt sich das Pseudonym niemals von den personenbezogenen Daten trennen. Mit UML-Aktivitätsdiagrammen lässt diese Maßnahme hingegen modellieren. Es ist ebenso möglich mit Signalen zu arbeiten. Das wiederum stellt eine Vereinfachung der Trennung von Prozessen und IT-Systemen dar, da neue Prozesse nun nur auf ein eintreffendes Signal warten müssen. Sie sind also einfacher austauschbar. BPMN erweitert diese Funktionalität noch dahingehend, dass auch die Art des Signals festgelegt werden kann. Weiterhin kann für jeden Prozess ein Verantwortlicher bestimmt werden, der den Prozess ausführt. Verantwortliche können dabei auch Web-Services sein.
Für alle drei Modellierungssprachen existieren jedoch Erweiterungen, die die Modellierung von Nicht-Verkettbarkeit auf die ein oder andere Weise erleichtern. Bei EPK's können dadurch Paragraphen als Annotation an die einzelnen Funktionen geheftet werden, die dann auf Nicht-Verkettbarkeit hinweisen können. UML bietet über UML-Profiles einen Mechanismus an um eigene Stereotypes zu definieren. Dadurch ließe sich beispielsweise Pseudonymisierung als Stereotype verwenden. Zuletzt gibt es in [He14] eine Erweiterung für BPMN, die Symbole für Qualitätsaspekte einführt, wodurch sich auch ein Symbol für Nicht-Verkettbarkeit einführen lässt.
Schlussendlich besteht jedoch auch hier noch Verbesserungsbedarf, da bei Einhaltung aller Datenschutzprinzipien die Modelle sehr schnell unübersichtlich werden und selbst die Erweiterungen hierbei nur bedingt Abhilfe schaffen.

Literaturverzeichnis

[Bi07] Bizer, Johann: Sieben Goldene Regeln des Datenschutzes. Datenschutz und Datensicherheit - DuD, 31(5):350–356, 2007.

[BM12] Bock, Kirsten; Meissner, Sebastian: Datenschutz-Schutzziele im Recht. Datenschutz und Datensicherheit - DuD, 36(6):425–431, 2012.

[He14] Heinrich, Robert: Quality Modeling within Business Process Models. In: Aligning Business Processes and Information Systems, S. 59–77. Springer Fachmedien Wiesbaden, 2014.

[He15] Heinrich, Robert; Gärtner, Stefan; Hesse, Tom-Michael; Ruhroth, Thomas; Reussner, Ralf; Schneider, Kurt; Paech, Barbara; Jürjens, Jan: A Platform for Empirical Research on Information System Evolution. In: Proceedings of the Twenty-Seventh International Conference on Software Engineering and Knowledge Engineering (SEKE'15). KSI Research Inc., S. 415–420, 2015.