

Mit Versionierungsinformationen im Requirements Interchange Format (ReqIF) echte Wiederverwendung von Anforderungen erreichen

Dr.-Ing. Oliver Alt
LieberLieber Software GmbH
Handelskai 340/5, A-1020 Wien
Oliver.Alt@lieberlieber.com

1 Einleitung

Die Wiederverwendung von Anforderungen spielt in der industriellen Praxis eine große Rolle. Insbesondere dort, wo ein Produkt nicht von Grund auf neu entwickelt, sondern eine neue Produktgeneration darstellt, können viele Anforderungen des Vorgängerproduktes unverändert wiederverwendet werden.

Leider unterstützen viele der am Markt befindlichen Anforderungsmanagementsysteme eine Wiederverwendung oftmals nur unzureichend oder sogar mangelhaft. Die Wiederverwendung besteht dann oft darin eine Kopie bestehender Anforderungen zu machen und diese im neuen Entwicklungsprojekt zu bearbeiten. Dadurch geht zumeist die Nachvollziehbarkeit zwischen Original und Kopie verloren und es werden Daten verdoppelt anstatt sie gemeinsam zu nutzen.

2 Requirements Interchange Format

Einen möglichen Ausweg aus dieser Situation bietet der Einsatz des Requirements Interchange Format (ReqIF) [OMG11]. ReqIF ist ein Datenformat, das entwickelt wurde um Anforderungsspezifikationen über verschiedene Anforderungsmanagementwerkzeuge hinweg auszutauschen. Ursprünglich aus einer Arbeitsgruppe im Automobilbereich als RIF [HIS12] hervorgegangen wird das Format nun als ReqIF seit 2011 durch die Object Management Group (OMG) als offizieller Standard gepflegt und veröffentlicht. Es reiht sich damit in andere Standards des Software- und Systems Engineering (z.B. UML und SysML) der OMG ein und kann damit nun als universell – nicht auf die Automotive-Domäne beschränkt – angesehen werden.

Mit Hilfe von ReqIF werden Anforderungen und andere Daten aus Spezifikationen als XML repräsentiert und können durch Werkzeuge, die das Format unterstützen, exportiert und importiert werden.

Neben dem Gedanken ein Austauschformat für Anforderungen bereit zu stellen beinhaltet die Datenstruktur von ReqIF aber auch noch ein Konzept, das die Wiederverwendung von Anforderungen sehr gut ermöglicht: Ein *Sichtenkonzept*. Dies bedeutet, dass Daten und die Anzeige von Daten zwei getrennte Dinge sind. Wenn man eine Anforderungsspezifikation (*ReqIF-Specification*) in ReqIF vorliegen hat, beinhaltet diese lediglich eine Struktur von Verweisen auf die eigentlichen Anforderungsdaten (*ReqIF-SpecObject*) – ähnlich dem Hyperlink Konzept der World Wide Web im Internet. Auch hier liegen die verlinkten Da-

ten einmal vor und können beliebig oft über Hyperlinks verlinkt werden.

Genau das Gleiche ist auch mit ReqIF möglich. Auf die gleichen Anforderungsdaten (ReqIF-SpecObject-Elemente) kann beliebig oft in Spezifikationen verwiesen werden.

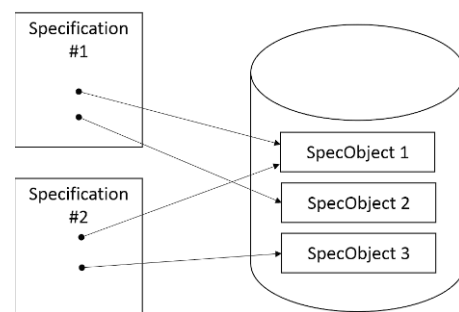


Abbildung 1: Prinzip der Sichten bei ReqIF

Abbildung 1 veranschaulicht das Prinzip der Sichtenbildung. Hier im Beispiel wird sowohl in Specification 1 als auch in Specification 2 SpecObject 1 mit eingebunden. In beiden Spezifikationen taucht also die entsprechende Anforderung auf.

3 ReqIF und Versionierung

Damit löst ReqIF schon das Problem der mehrfachen Verwendung gleicher Anforderungen in verschiedenen Spezifikationen. Was aber, wenn sich in einem neuen Projekt eine bestehende Anforderung ändert? Die hier vorgeschlagene Lösung zu diesem Problem ist die Kombination von ReqIF mit Techniken des Konfigurationsmanagement und der Versionskontrolle, wie es schon seit langem in der Softwareentwicklung üblich ist. Hierzu werden Dateien oder Datensätze mit einer Version versehen und es ist möglich eine neue Version durch Veränderung zu erzeugen, gleichzeitig aber auch die alte Version wieder herzustellen. Typische Operationen im Konfigurationsmanagement sind auch das Abzweigen von neuen Versionen von einem Hauptzweig (*Branching*), sowie das Zusammenführen von zwei Versionen zu einer neuen gemeinsamen (*Merging*).

Um ReqIF und Versionierung zu kombinieren wird folgendes vorgeschlagen: Jedes Element in ReqIF hat eine eindeutige Kennung (*Unique ID*). Diese besteht zumeist aus einer generierten Folge von Ziffern und Zeichen (GUID). Wenn man dieser GUID zusätzlich Versi-

onskennungen in der Form '/' + <Zweigname> + '-' + <Versionsnummer> als Postfix anhängt, dann ist es möglich ein Element wie ein ReqIF-SpecObject mit entsprechender Werkzeugunterstützung über GUID und Version anzusprechen. Ändert sich ein ReqIF-SpecObject, wird keine neue GUID erzeugt, sondern ein neues Element mit gleicher GUID, aber anderer, neuerer Versionskennung.

Dieser Ansatz hat den Vorteil, dass er keine Änderung des ReqIF-Standards benötigt, da die Unique ID als String definiert ist und dies auch weiterhin durch Kombination von GUID und Versionskennung gewährleistet ist. ReqIF-Daten mit Versionsinformationen sind damit auch noch von ReqIF-unterstützenden Werkzeugen zu bearbeiten, die (noch) keiner Unterstützung des Versionskonzepts bieten. Sofern noch andere Informationen für das Konfigurationsmanagement (z.B. BranchedFrom, PreviousVersion etc.) benötigt werden, können diese über die in ReqIF frei definierbaren Attribute eines ReqIF-SpecObject definiert werden.

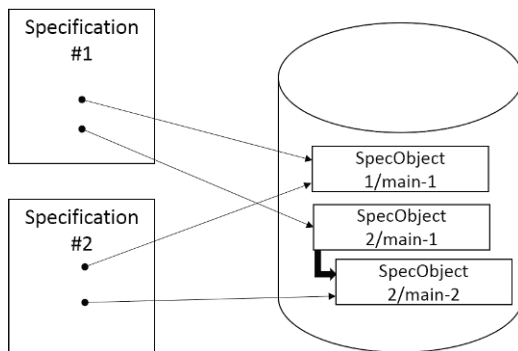


Abbildung 2: Versionierung mit ReqIF

In Abbildung 2 ist das Prinzip von nun mit Versionsinformationen versehenen ReqIF-SpecObject-Elementen gezeigt. Beide Spezifikationen nutzen Element 1 in Version main-1 gemeinsam. Zusätzlich verweisen die Spezifikationen aber nun beide auf *verschiedene Versionen* von Element 2. Dies kann dann zum Beispiel der Fall sein, wenn Element 2 in Spezifikation 2 weiterentwickelt wurde. Dadurch, dass hier sowohl das Referenz- als auch das Versionierungskonzept kombiniert eingesetzt werden, können zum einen Anforderungen wiederverwendet und zum anderen Änderungen zu 100% nachvollzogen werden.

Es kann den Fall geben, dass Attribute eines SpecObject-Elements sich ändern, die selbst keinen Einfluss auf den Inhalt der Anforderung haben. Dies sind z.B. „Anforderung ist getestet“ und andere Statusinformationen. Solche Attribute würden zu einer neuen Version ohne Inhaltsänderung führen und die Wiederverwendung damit aushebeln. Man kann dies dadurch vermeiden, dass man solche Informationen nicht beim SpecObject selbst, sondern in einem zweiten SpecObject ablegt, welches mit dem anderen durch eine ReqIF-Relation, die wiederum versioniert sein kann, verlinkt ist. Solche Vernetzungen von Entwicklungsdaten und Trackinginformationen sind aus Konfigurationsmanagementsystemen seit langem be-

kannt und werden dort bereits eingesetzt. Beim Aufbau des ReqIF-Datenmodells sollten solche Aspekte von vornherein berücksichtigt werden, dann gibt es in der Anwendung später auch keine Probleme mit der Wiederverwendung.

4 Fazit

Der Einsatz von ReqIF als standardisiertes und universelles Format zum Austausch von Spezifikationsdaten erlaubt es, Anforderungen auch über Werkzeuggrenzen hinweg auszutauschen. Neben der reinen Nutzung als Austauschformat bringt ReqIF mit dem Sichtenprinzip aber auch ein Konzept mit, das die echte Wiederverwendung von Anforderungen ohne Verdoppelung der Daten erlaubt. ReqIF eignet sich damit auch hervorragend als direkt verwendetes Datenmodell (Metamodell) für Anforderungsmanagementsysteme.

Mit der hier vorgeschlagenen Erweiterung der Unique-ID um Versionsinformationen kann eine Lücke in der Nachvollziehbarkeit von Anforderungsänderungen geschlossen werden, ohne dass dazu eine Anpassung des ReqIF-Standards notwendig wäre. Mit solchen ReqIF-Daten können Anforderungen wiederverwendet und alle Änderungen nachvollziehbar dokumentiert werden, wie es in modernen Entwicklungsprozessen notwendig ist. Dies erleichtert die Arbeit der Anforderungsingenieure und führt letztendlich auch zu schnellen und konsistenten Arbeitsergebnissen. Natürlich wäre es wünschenswert, dass sich der ReqIF-Standard in eine Richtung bewegt, die eine solche, auf textuelle ID-Erweiterungen basierende Lösung überflüssig macht. Damit wäre dann auch sichergestellt, dass die ReqIF-Werkzeuge alle ID-Informationen und – sofern unterstützt – die Versionierungsinformationen auf jeden Fall richtig interpretieren und speichern.

Literatur

- [HIS12] HIS. *Homepage zum RIF-Format*. Herstellerinitiative Software, 2012. <http://www.automotive-his.de/rif/doku.php>, zuletzt besucht 29.11.2012.
- [OMG11] OMG. *Requirements Interchange Format (ReqIF) Version 1.0.1*. Object Management Group, 4 2011.