



□ Andreas Lux

[E-Mail: [a.lux@excellent.de](mailto:a.lux@excellent.de)]

ist Projektmanager und Gesellschafter der eXXcellent solutions GmbH in Ulm. Sein Schwerpunkt liegt in der ganzheitlichen Durchführung von modellgetriebenen und individuellen Business-Lösungen. Neben seiner Verantwortung für den Softwareentwicklungsprozess gilt sein besonderes Interesse dem Thema modellbasierte Softwareentwicklung und im speziellen dem Thema Requirements Engineering und dessen Integrationen in den Entwicklungsprozess.

## Requirements Interchange Format (ReqIF) – kollaborative Anforderungsanalyse in verteilten Projekten

Requirements Interchange Format (ReqIF) – ein Datenaustauschformat unter der Obhut der Object Management Group (OMG) – wurde entworfen, um die Interoperabilität zwischen Werkzeugen zur Anforderungsanalyse unterschiedlicher Hersteller zu gewährleisten. Aber reicht das? Unserer Erfahrung nach geht es um deutlich mehr, als um den reinen Austausch von Daten. Es geht um Kollaboration und um Integration – zwischen Werkzeugen, aber vor allem zwischen Menschen und den Prozessen, in denen sie interagieren. In diesem Artikel wird anhand des ReqIF Servers aufgezeigt, wie Teamarbeit in verteilten Teams mit ReqIF und dem ReqIF Server sinnvoll und effizient gestaltet werden kann. Es wird exemplarisch erläutert, wie man den ReqIF Server in diverse Prozesse integrieren kann. Außerdem wird skizziert, was man außer dem reinen Datenaustauschformat noch an weiteren Funktionalitäten benötigt, um mit diesem Ansatz auch in realen Projektsituationen bestehen zu können.

### Worum geht es?

Anforderungen erfassen – prüfen – priorisieren – freigeben – testen sind Standard-szenarien bei der Anforderungsanalyse. Bei genauerer Betrachtung stellt man oft fest, dass unterschiedliche Personen in unterschiedlichen Rollen aus unterschiedlichen Standorten und Organisationen zusammenarbeiten müssen und der gesamte Prozess über diverse Infrastrukturen und Werkzeuge hinweg reibungslos funktionieren muss. ReqIF, das Standardformat der OMG zum Austausch von Anforderungen, verspricht hier deutliche Linderung der operativen Schmerzen.

### Der fachliche Anwendungsfall

Um Anforderungsanalyse und -management in möglichst hoher Effizienz und Qualität umsetzen zu können, haben sich über die Zeit diverse Werkzeuge herausgebildet, mit denen sich Anforderungen formulieren und managen lassen. Die Vielfalt der Werkzeuge hat dazu geführt, dass abhängig vom jeweiligen Umfeld und der

Arbeitsweise der Teams das jeweils passende Werkzeug zur Verfügung steht.

Es hat aber leider auch dazu geführt, dass in kollaborativ geführten Projekten über Unternehmensgrenzen hinweg kommuniziert werden muss, hier aber oft unterschiedliche Werkzeuge zum Einsatz kommen.

Dies hat erfahrungsgemäß negative Auswirkungen auf die Möglichkeiten des Datenaustauschs von Anforderungen zwischen den unterschiedlichen Teams und stellt ein messbares Hindernis der Produktivität dar.

Das Problem hierbei ist, dass die unterschiedlichen Anforderungswerkzeuge unterschiedliche Philosophien haben, wie sie ihre Daten strukturieren. Dies spiegelt sich z. B. in der Art und Weise wider, wie die Werkzeuge ihre Daten im- und exportieren. Um diesem Problem zu begegnen, wurde Requirement Interchange Format (RIF) entworfen.

### Historie

RIF wurde von ProSTEP [pro], einem Verein, der sich um die Standardisierung

von Prozessen und Lösungsansätzen vornehmlich im Automotive-Umfeld kümmert, entwickelt. Es soll einen herstellerunabhängigen Datenaustausch zwischen Werkzeugen zur Anforderungsanalyse ermöglichen. Die Definition von RIF wurde inzwischen unter dem Namen ReqIF an die OMG übergeben und wurde Anfang April dieses Jahres unter der Version 1.0.1 veröffentlicht [omg].

### ReqIF – Was ist das?

Bei RIF/ReqIF handelt es sich um ein XML Schema, über das sich komplette Anforderungsgraphen austauschen lassen. Die – in diesem Artikel vereinfacht dargestellte – Idee dieses Formats ist es, komplexe Informationsobjekte (SpecObjects) und Beziehungen (SpecRelations) mit deren Attributen (AttributeValues) zu kommunizieren. SpecObjects lassen sich auch wahlweise in unterschiedlichen Hierarchien (SpecHierarchies) strukturieren und somit in einen semantischen Zusammenhang

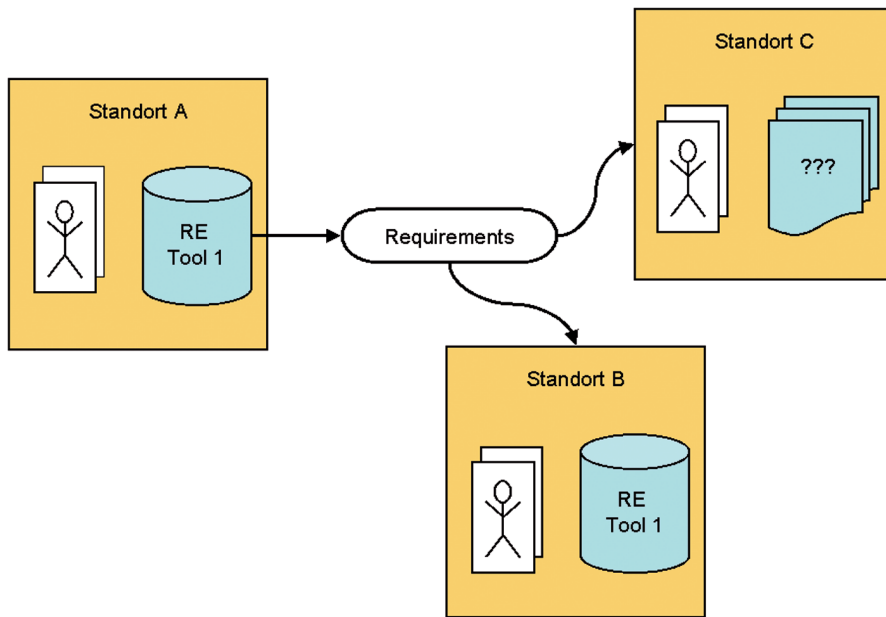


Abb. 1: Übersicht Datenaustausch

bringen. SpecObjects und SpecRelations basieren auf einer Typdefinition (SpecType), die definiert, welche AttributeValues ein Objekt enthalten kann/soll und von welchem Typ (Data-typeDefinition) die Attribute sind.

Welche Typen es gibt, wird pro ReqIF-Datei festgelegt. Somit ist es möglich, Objekttypen, wie z. B. „Requirement“ oder „Use-Case“, mit jeweils der passenden Struktur (sprich: Attributierung) zu definieren. Dasselbe gilt für Beziehungstypen, wie z. B. „depends“, „includes“ o. ä., auch diese können in ReqIF auf einer Typdefinition basieren und eine eigene Attributierung besitzen.

Angereichert wird die inhaltliche Definition noch mit Access Policies, die definieren, ob Informationselemente im importierenden Werkzeug editierbar, löscherbar oder erzeugbar sein sollen.

Mit diesem typisierten und hierarchischen Konzept ist es nun möglich, auch sehr komplexe Informationsstrukturen zielgruppengenau zu kommunizieren.

**Kollaboration – die Herausforderungen**

Um Anforderungen in verteilten Teams auszutauschen, ist die Definition des Datenaustauschformats nur eine der notwendigen Maßnahmen. Eine weitere Herausforderung, und vermutlich die komplexere, ist die Einführung des richtigen

Szenarios mit den beteiligten Partnern (z. B. den Zulieferern).

Die Frage, die es hier zu beantworten gilt, ist: Mit welchem Ziel werden die Partner angebunden?

- Sollen sie über aktuelle Inhalte und Stände informiert bleiben?
- Sollen von ihnen Zustimmungen oder Kommentare eingeholt werden?
- Sollen sie die an sie kommunizierten Inhalte autark ändern dürfen?
- Müssen die geänderten Daten wieder in andere Systeme rücküberführt werden können?
- Ist mehr als ein Partner an der Kommunikation beteiligt?
- Dürfen / sollen / müssen diese untereinander die geänderten Inhalte sehen oder müssen Kommentare / Ergänzungen / Änderungen der unterschiedlichen Partner untereinander versteckt bleiben?

Nicht immer existiert bei den beteiligten Partnern ein bereits eingeführtes Werkzeug zur Anforderungsanalyse, mit dem die Szenarios durchprozessiert werden können. Auch sind nicht alle beteiligten Partner bereit / fähig, in so ein Werkzeug zu investieren, um die Szenarios optimal zu unterstützen. Darüber hinaus unterstützt nicht jedes der bei den beteiligten Partnern bereits eingeführten Werkzeuge die genannten Szenarios.

**Basisfunktionalität einer Kollaborationsplattform**

Vor allem in größeren Organisationen sind die an einem Projekt beteiligten Personen oft über mehrere Standorte verteilt. Sollen externe Partner mit in den Prozess integriert werden, müssen sehr oft Firmengrenzen überwunden werden. Sollen alle Partner gemeinsam an einem System arbeiten ist es notwendig, den Beteiligten einen möglichst einfachen Zugriff auf den gemeinsamen Datenbestand zu ermöglichen.

Um möglichst wenige Hürden aufzubauen, bleibt als Lösung oft nur die Möglichkeit, eine Webanwendung zu verwenden. Rich Client-Anwendungen einzusetzen ist aufgrund der Anforderungen an die IT-Netzwerklandschaft der beteiligten Unternehmen oft nicht möglich.

Auch das Argument, Webanwendungen würden eine nur unzureichende Bedienbarkeit bieten, ist schon lange nicht mehr tragbar. Der Bedienkomfort aktueller AJAX-Anwendungen ist inzwischen so hoch, dass er sich nicht mehr hinter der Bedienbarkeit von Desktopanwendungen verstecken muss.

**Informationsbedarf**

Mit einer webbasierten Anwendung ist es nun leicht möglich, alle beteiligten Personen sehr einfach mit Informationen zu einem Projekt zu versorgen und sie in den Prozess einzubinden. Hierbei muss man sich die Frage stellen, welchen Informationsstand die beteiligten Personen sehen sollen.

Personen, die beispielsweise an der Gestaltung von Anforderungen mitarbeiten, sollten möglichst die letzte Arbeitsversion einer Anforderung sehen und eventuell auch bearbeiten können. Personen, die konkrete Anforderungen für ihr Projekt umsetzen sollen, benötigen dagegen einen konsolidierten und freigegebenen Stand der Anforderungen zu einer dedizierten Version. Dies lässt sich mit bestehenden Werkzeugen oft nicht einfach umsetzen.

**Editierbarkeit**

Eine sinnvolle Integration von Partnern ist oft nur dann gegeben, wenn diese auch in der Lage sind, Inhalte zu modifizieren. Hierbei geht es oft gar nicht darum, Anforderungen komplett neu zu erfassen, sondern Partner in einen Kollaborationsprozess, beispielsweise zu Abstimmungen und Freigaben zu bestehenden

Anforderungen, zu integrieren. Um dies tun zu können, muss es für die Partner möglich sein, bestimmte Attribute einer Anforderung zu ändern. Beispielsweise können Partner so in einen Reviewprozess einbezogen werden, bei dem sie einzelne Anforderungen freigeben und kommentieren können.

In realen Projekten ist es erfahrungsgemäß oft gegeben, dass Partner aus unterschiedlichen Organisationen und in konkurrierenden Rollen (z.B. mehrere Zulieferer) in einem Projekt zusammenarbeiten. In diesem Fall entsteht oft die Situation, dass alle beteiligten Zulieferer zwar eine gemeinsame Sicht auf die Anforderungen eines Projekts oder einer Version haben, ihre Kommentare und Statusänderungen zu einer Anforderung aber für andere Zulieferer nicht sichtbar sein dürfen.

Dem entgegengesetzt könnte ein Projektleiter nun die Aufgabe haben, die unterschiedlichen Rückflüsse der Zulieferer zu konsolidieren und in einen gemeinsamen Stand zu überführen. Um diese Aufgaben von allen Beteiligten bewältigen zu können, ist ein ausgefeiltes Rechtemanagement notwendig, das es erlaubt, Rechte nicht nur auf Projektebene zu vergeben, sondern Sichtbarkeiten und Aktionen attributgenau festlegen zu können.

**ReqIF Server**

Eine mögliche Lösung für die bisher genannten Punkte stellt der ReqIF Server [req] dar. Abstrakt betrachtet handelt es sich beim ReqIF Server um eine serverbasierte Anwendung, die in der Lage ist, sowohl RIF als auch ReqIF Modelle inklusive der referenzierten Anhänge, wie Bilder und Dokumente, und unter Betrachtung der AccessPolicies zu laden und auch wieder zu exportieren.

Die geladenen Modelle können über die integrierte Webanwendung ReqIF Views betrachtet und bearbeitet werden. Darüber hinaus bietet der ReqIF Server die Möglichkeit, für eine automatisierte Einbindung in individuelle Prozesse auf jegliche Modellinhalte sowohl lesend als auch schreibend über ein RESTful Webservice Interface zugreifen zu können.

**Generische Darstellung der Inhalte im WEB**

Die importierten Inhalte werden dem Anwender über einen generischen Webclient, genannt ReqIF Views, visualisiert. Hierbei

|                     |   |
|---------------------|---|
| <b>Rich Client:</b> | Anwendung, bei der ein Großteil der Logik in der Clientanwendung ausgeführt wird und die oft über spezielle Protokolle mit einem Server/einer Datenbank kommuniziert. |
| <b>AJAX:</b>        | Technologie, mit der sich Webanwendungen so bauen lassen, dass diese sich für den Anwender ähnlich komfortabel „anfühlen“ wie Desktopanwendungen.                     |
| <b>Webservice:</b>  | XML-basierte Schnittstellentechnologie, die mit Standard Web-Technologien arbeitet und deshalb oft problemlos über Firmennetzwerke hinweg einsetzbar ist.             |
| <b>RESTful WS:</b>  | Sehr einfache und flexibel verwendbare Variante eines zustandslosen Webservices.  |

*Tabella: Technologie*

betrachtet die ReqIF Views-Anwendung die im RIF/ReqIF-Import definierten Typen und Hierarchien und bietet dem Anwender ohne weitere Installation die Möglichkeit, die Inhalte entsprechend ihrer Struktur und unter Berücksichtigung der in der Datei enthaltenen AccessPolicies zu betrachten oder zu editieren und durch die Projekte zu navigieren. Die Navigation erfolgt hierbei über die im Importfile hinterlegten Hierarchien, die definierten SpecTypes oder entlang von SpecRelations, die unterschiedliche SpecObjects miteinander verbinden.

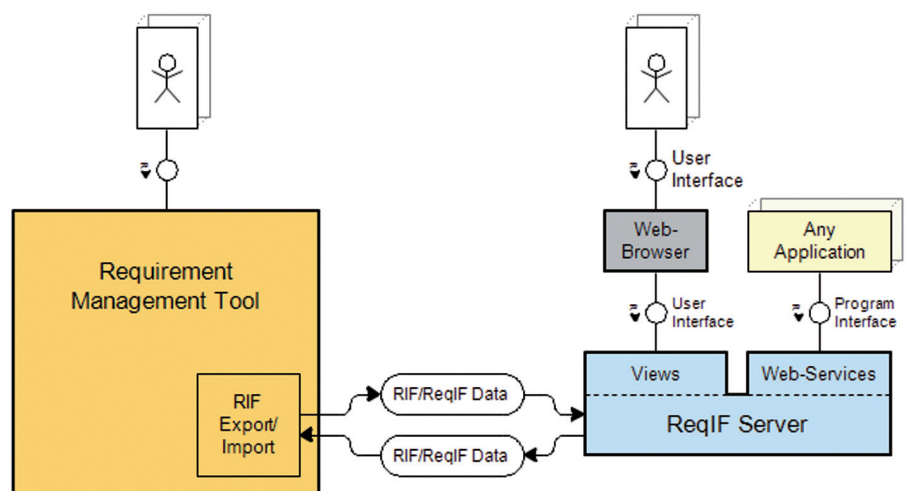
**Zielgruppenspezifische Darstellung**

Bei der Darstellung der Inhalte in ReqIF Views wurden zwei grundlegend unterschiedliche Bedürfnisse bzw. Gewohnheiten der Anwender betrachtet:

- Anwender aus einer eher dokumentenzentrischen Welt (wie z.B. DOORS-Anwender) haben erfahrungsgemäß das Bedürfnis, die einzelnen Anforderungen wieder in einer Art „Dokument“ zu sehen. Dies ist wichtig für das Verständnis der einzelnen Anforderungen, da diese oft einzelnen Absätzen aus physikalischen Dokumenten entsprechen und die Reihenfolge der Anforderungen untereinander eine große Rolle zum ganzheitlichen Verständnis spielen kann.
- Anwender aus einer eher eintragszentrischen Welt dagegen wollen die konkrete Anforderung als atomare Einheit mit all ihren Feinheiten und Beziehungen zu anderen Anforderungen betrachten. Die Reihenfolge der Anforderungen untereinander spielt dabei eine eher untergeordnete Rolle.

**Maschinenlesbarer Zugriff über RESTful Webservices**

Betrachtet man den kompletten Lebenszyklus bei der Durchführung von Projekten oder bei der Erstellung von Produkten, ist die Anforderungsanalyse „nur“ einer von mehreren, wenngleich auch einer der zen-



**Abb. 2:** Datenaustausch mit ReqIF Server

tralsten, Arbeitsschritten im Projekt. Die in der Anforderungsanalyse erstellen Inhalte stellen die Basis für alle nachfolgenden Prozessschritte und Artefakte dar. Folglich ist es im Sinne der Nachvollziehbarkeit essenziell, dass die originär erstellten Inhalte auch eindeutig referenziert werden können. Typischerweise wird dies in vielen Werkzeugen über eindeutige IDs o. ä. gemacht.

Die im ReqIF Server umgesetzte Möglichkeit zur Identifizierung eines Eintrags geht hier noch einen deutlichen Schritt weiter. Alle Inhalte im ReqIF Server sind über eindeutige URLs referenzierbar. Hinter diesen URLs verbirgt sich ein umfangreiches System von RESTful Webservices. Diese ermöglichen es dem Aufrufer, die Inhalte des referenzierten Eintrags zu lesen, über Relations zu den referenzierten Einträgen zu navigieren und Einträge zu modifizieren oder auch zu löschen.

Durch die Einführung des REST Service Interface können die im ReqIF Server verwalteten Inhalte von so gut wie allen im konkreten Projekt verwendeten Werkzeugen einfach referenziert werden.

Die URL zum Lesen eines Eintrags könnte wie folgt aussehen:

```
https://www.excellent.de/rif-server/v0.33/projects/<project-id>/spec-objects/<object-id>
```

Durch die manipulativen REST Interfaces besteht darüber hinaus die Möglichkeit, die Inhalte des ReqIF Servers nahtlos in automatisierte Prozesse zu integrieren, mit denen z. B. Anforderungen im ReqIF Server automatisch an Zustandsänderungen von Einträgen aus Drittsystemen angepasst werden.

Beispielsweise kann der Zustand einer Anforderung im ReqIF Server automatisch über das entsprechende Rest Interface auf „tested“ gesetzt und das Testprotokoll an die konkrete Anforderung angehängt werden, wenn der entsprechende Test im verwendeten Testmanager erfolgreich durchgeführt wurde.

**Versionierung**

RIF und ReqIF wurden entworfen, um den standardisierten Austausch von Anforderungsdefinitionen zwischen Werkzeugen zu ermöglichen.

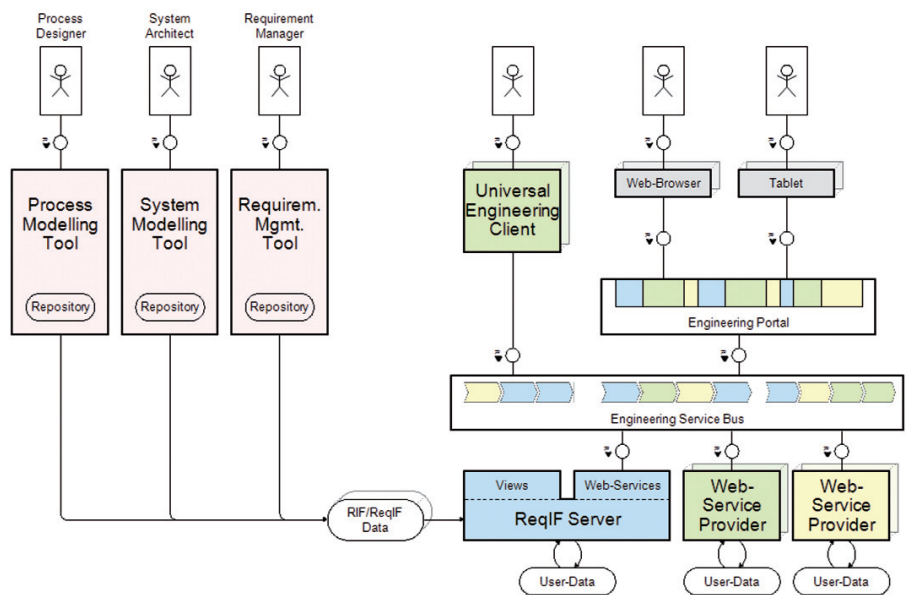


Abb. 3: ReqIF Server als Service in automatisierten Prozessen

derungsdefinitionen zwischen Werkzeugen zu ermöglichen. Hierbei wird angenommen, dass die Inhalte, die sich in den Dateien befinden, als eine Art „Snapshot“ gesehen werden.

Ein System wie der ReqIF Server, der einen simultanen transaktionalen Zugriff unterschiedlicher Benutzer und Rollen auf die Inhalte ermöglicht, muss weiteren Ansprüchen gerecht werden. Wenn mehrere Benutzer unabhängig voneinander Inhalte manipulieren können, ist eine Versionierung der Inhalte aus Gründen der Transparenz im Prozess absolut notwendig.

Im ReqIF Server werden neue Versionselemente erstellt, sobald die Änderung an einem SpecObject oder einer SpecRelation gespeichert wird. Dieses Verhalten wird sowohl in ReqIF Views als auch in den REST Interfaces vollumfänglich unterstützt. Konsequenterweise ist in beiden Schnittstellensystemen sowohl ein Zugriff auf den jeweils aktuellen Eintrag als auch auf Einträge zu dedizierten Versionen möglich. Somit ist es jederzeit möglich, das Beziehungsnetzwerk eines Eintrags zu einem dedizierten Zeitpunkt zu analysieren.

**Fazit**

Der ReqIF Server mit ReqIF Views und den RESTful Webservices kann eine überaus sinnvolle und auch preiswerte Ergänzung der bestehenden Werkzeuglandschaft zur Anforderungsanalyse sein. Er schließt die Lücke, wo der Einsatz der bestehenden Werkzeuge nicht ausreicht oder Partner eingebunden werden sollen, bei denen der Einsatz eines eigenen Werkzeugs zur Anforderungsanalyse nicht sinnvoll erscheint.

Durch die mitgelieferten REST Services lassen sich die Inhalte, die im ReqIF Server verwaltet werden, nahtlos und maschinenverarbeitbar in eine komplexe Toollandschaft integrieren und ermöglichen es so, höchst effektive Prozesse zu implementieren, die einem realen Projektleben gewachsen sind.

**Referenzen**

- [pro]: <http://www.prostep.com>
- [omg]: <http://www.omg.org/spec/ReqIF/>
- [req]: <http://www.reqif.de>