

Verfolgbarkeit in großen Prozessstandards am Beispiel der Raumfahrtindustrie

Ove Armbrust, Alexis Ocampo,
Martín Soto, Jürgen Münch

Fraunhofer-Institut für
Experimentelles Software
Engineering (IESE)
Fraunhofer-Platz 1
67663 Kaiserslautern
{armbrust, ocampo, soto,
muench}@iese.fhg.de

Masafumi Katahira,
Yumi Koishi, Yuko Miyamoto

Japan Aerospace
Exploration Agency
2-1-1 Sengen, Tsukuba, Ibaraki
305-8505 Japan

{katahira.masafumi, koishi.yumi,
miyamoto.yuko}@jaxa.jp

Abstract: Die Raumfahrtbranche ist bekannt für ihre konsequente Ausrichtung auf hohe Produktqualität, da Hardware- oder Softwarefehler potenziell katastrophale Folgen haben können. Daher regeln viele Standards die Entwicklung von Software für die Raumfahrt. In diesem Paper präsentieren wir einen Ansatz zur systematischen Etablierung und Erhaltung von Verfolgbarkeitsbeziehungen zwischen Standards als Basis für eine langfristige Evolution dieser Standards. Der Ansatz reichert die entsprechenden Textverarbeitungsdokumente mit zusätzlichen Meta-Informationen an, sodass sie für Programme zu verarbeiten sind. Eine Datenbankrepräsentation der Standards erlaubt tiefgehende Analysen. Wir berichten über unsere Erfahrungen aus zwei erfolgreichen Anwendungen des Ansatzes bei der Europäischen Raumfahrtorganisation ESA und der Japanischen Raumfahrtagentur JAXA.

1 Einleitung

Die Luft- und Raumfahrtindustrie legt traditionell großen Wert auf die Qualität sowohl der verwendeten Bauteile (Hardware) als auch der eingesetzten Software. Der Grund dafür liegt in strengen Anforderungen an Sicherheit und Verlässlichkeit. Ein weithin anerkannter Weg zur Erreichung der angestrebten Qualität ist die Anwendung von geeigneten, ausführlichen Standards. Diese Standards liegen oft in Form von Dokumenten für Microsoft Word, OpenOffice.org Writer oder im PDF-Format vor.

Da das abzudeckende Themengebiet sehr groß ist, reicht üblicherweise ein einzelner Standard nicht aus. Daher existiert in der Regel eine Vielzahl von hierarchischen, miteinander verbundenen Standards. Ein Beispiel für solche Standards sind jene der Euro-

pean Cooperation for Space Standardization (ECSS) [Eur09] für die Europäische Raumfahrtagentur ESA. Die ECSS-Standards enthalten Anforderungen an die Entwicklungs- und Managementprozesse von Organisationen, die der ESA als Zulieferer dienen (möchten). Innerhalb der ECSS-Standards regeln drei Serien die Entwicklungsprozesse (E-Serie), die Qualitätssicherungsprozesse (Q-Serie) und die Projektmanagementprozesse (P-Serie). Andere Beispiele sind etwa die entsprechenden Standards der U.S. National Aeronautics and Space Administration (NASA) [NAS09] oder der Japanischen Raumfahrtagentur JAXA.

Die hierarchische, stark miteinander verwobene Struktur solcher Standards stellt Prozessingenieure vor mehrere große Herausforderungen. Zum Beispiel wird das Tailoring von individuellen Standards durch die vielen Abhängigkeiten von anderen Standards erschwert. Die Weiterentwicklung und Wartung der Standards ist ähnlich herausfordernd, da alle gegenseitigen Abhängigkeiten berücksichtigt werden müssen. In vielen Fällen ist auch ein formaler Nachweis der Konformität zu übergeordneten Standards erforderlich, beispielsweise wenn ein organisationsspezifischer Standard entwickelt wird. Schließlich erfordert die Anwendung von fortgeschrittenen Konzepten wie z.B. Prozesslinien [Rom05] [AO09] [Ar08], welche die Konzepte von Software-Produktlinien auf Prozesse übertragen, ein großes Maß an Kontrolle über alle beteiligten Standards.

Um diese Aktivitäten erfolgreich bewältigen zu können, ist eine detaillierte und qualitativ hochwertige Verfolgbarkeit zwischen Elementen der einzelnen Standards erforderlich. Der Aufbau und die Aufrechterhaltung dieser Verfolgbarkeit sind momentan sehr arbeitsintensiv, da ein großer Teil der dafür notwendigen Aktivitäten manuell durchzuführen ist. Die Tatsache, dass die entsprechenden Standards oftmals als (mäßig brauchbar strukturierte) Word-Dokumente vorliegen, macht die Aufgabe nicht einfacher. Dieses Paper beschäftigt sich daher mit der Frage, wie Verfolgbarkeit in einer hierarchischen Multi-Standard-Umgebung systematisch und so wenig aufdringlich wie möglich hergestellt und beibehalten werden kann.

Das Paper ist wie folgt strukturiert. Kapitel 2 gibt einen kurzen Überblick über verwandte Arbeiten zu diesem Thema. Kapitel 3 stellt unseren Ansatz vor, Kapitel 4 den von uns erwarteten Nutzen. Kapitel 5 beschreibt zwei Fallstudien, die den vorgestellten Ansatz verwendet haben, sowie ihre Ergebnisse bezüglich des erwarteten Nutzens. Kapitel 6 zieht abschließende Schlüsse und skizziert mögliche weitere Arbeiten.

2 Verwandte Arbeiten

Ein Textverarbeitungsansatz, der dem hier vorgestellten Ansatz ähnelt, aber in einem völlig anderen Kontext angewendet wurde, ist in [PB03] beschrieben, wo ein XML-Editor zum Bearbeiten von Rechtstexten entwickelt wurde. Das XML-Format erlaubte dabei die Verarbeitung derartiger Texte und die Extraktion strukturierter Daten (Metadaten genannt: Art des Gesetzes, Nummer, Veröffentlichungsdatum etc.) für verschiedene Anwendungen.

Verfügbare kommerzielle Tools wie DOORS [Tel09] unterstützen beliebige Relationen zwischen Softwareanforderungen. In unserem Fall war es jedoch aufgrund der Vielzahl und der Inhomogenität der interessierten Parteien unbedingt notwendig, die Word-Dokumente beizubehalten. Dies ermöglichen auch kommerzielle Werkzeuge wie zum Beispiel IBM Rational Requisite Pro [IBM09], welches aber auf Anforderungen fokussiert und beispielsweise Verfolgbarkeitsbeziehungen nicht klassifizieren oder annotieren kann. Darüber hinaus existieren in Prozessstandards auch viele implizite Beziehungen zwischen Entitäten, beispielsweise zwischen Aktivitäten und ihren Arbeitsprodukten, die von solchen Werkzeugen nicht erfasst werden, aber bezüglich der Konsistenzsicherung enorm wichtig sind.

In der Software-Engineering-Gemeinde gewinnt die (Wieder-) Herstellung von Beziehungen (traceability recovery) zwischen existierenden Entwicklungsartefakten zunehmend an Bedeutung. Verschiedene Ansätze wie z.B. Latent Semantic Indexing (LSI) [Lu04] oder Ontologien [Zh06] bestimmen die Forschungslandschaft in dieser Hinsicht.

In der Raumfahrtindustrie wird schon lange aktiv die Standardisierung von Prozessen betrieben; eine ausführliche Begründung wird in [CMC05] gegeben. Ein Assessment-Modell zur Bestimmung der Prozesskonformität wurde ebenfalls entwickelt und unter dem Namen SPICE for Space (S4S) veröffentlicht [Ca01]. Verfolgbarkeit zwischen verschiedenen Standards ist insbesondere wichtig in Software-Prozesslinien-Umgebungen [Rom05] [AO09] [Ar08], in denen verschiedene Prozessvarianten aus einem gemeinsamen Kernstandard abgeleitet werden.

Oft wird in Bezug auf Verfolgbarkeit auch der umgekehrte Weg gewählt: Aus einem (mehr oder weniger) abstrakten Modell wird eine Dokumentation generiert, die für die Nutzung durch Menschen gedacht ist. Ein Beispiel für ein solches Vorgehen ist das V-Modell XT [Koo08], welches einem Metamodell folgt und auch als Modell bearbeitet wird, nicht als Fließtextdokument. Dadurch werden sämtliche Beziehungen explizit modelliert und können auch dementsprechend analysiert werden. Dies dient der Konsistenzsicherung, macht die Bearbeitung jedoch auch schwieriger als eine einfache Änderung von Textdokumenten.

Dieses Paper basiert auf einer früheren Veröffentlichung der Autoren [Ar09] und erweitert und detailliert den Beitrag.

3 Datenbankunterstützte Verfolgbarkeit

Für den Rest dieses Papers nehmen wir an, dass unsere Aufgabe die Identifizierung von Verfolgbarkeitsbeziehungen zwischen einem übergeordneten (Master-) Standard und einem davon abgeleiteten (Slave-) Standard ist. Wir nutzen dabei die Tatsache, dass Softwareprozessstandards üblicherweise eine Reihe von Konzepten nutzen, um den Inhalt zu strukturieren. Dazu gehören beispielsweise Entitäten wie Aktivitäten und

Arbeitsprodukte sowie deren hierarchische Untergliederung. Um die relevanten Inhalte zwischen Standards verfolgbar zu machen, schlagen wir folgende Schritte vor:

- (1) Regeln für Veränderungen festlegen
- (2) Dokumente mit Meta-Informationen anreichern
- (3) Informationen für Programme identifizierbar machen durch die Verwendung von Formatvorlagen
- (4) Verfolgbarkeit zwischen den Standards herstellen
- (5) Synchrones Abbild der Standards in der Datenbank erzeugen und erhalten durch direkte Verarbeitung ihrer XML-Repräsentation
- (6) Nutzung der Datenbank-Repräsentation der Standards zur Aggregation, Analyse und Visualisierung der aus den Standards extrahierten Informationen.

3.1 Regeln für Veränderungen festlegen

Softwareprozessstandards unterscheiden sich von „normalen“ Dokumenten wie beispielsweise Büchern oder Zeitungsartikeln unter anderem dadurch, dass bei ersteren teilweise sehr abstrakte Konzepte mittels Text dargestellt werden, während letztere mehr erzählerisch-beschreibend bestimmte Sachverhalte erläutern. Dazu sind solche Standards in einer ganz bestimmten Art und Weise strukturiert. Zum Beispiel wird üblicherweise eine Reihe von Aktivitäten beschrieben, die bestimmte Inputs benötigen und bestimmte Outputs erzeugen. Abhängig vom jeweiligen Standard werden oft auch noch Rollen beschrieben sowie benötigte Ressourcen, teilweise auch noch andere Arten von Informationen. Um die bei der Beschreibung von Softwareprozessen auftretende Komplexität zu beherrschen, werden weiterhin Strukturierungshilfen wie Hierarchien oder Sichten verwendet. Damit beschreibt ein Softwareprozessstandard im Grunde ein Modell, welches aus bestimmten Entitäten und ihren Beziehungen besteht.

Bei der Frage nach der Verfolgbarkeit zwischen solchen Standards ist daher meist nicht eine beliebige Verfolgbarkeit von Interesse, sondern nur ganz bestimmte Informationen. Dazu gehört beispielsweise die Information, welche Aktivität von welcher anderen Aktivität abgeleitet ist, oder welchen Output eine Aktivität erzeugt, und für welche anderen Aktivitäten diese Arbeitsprodukte wieder Input sind. Im hier betrachteten Fall müssen diese Entitäten im Textdokument kenntlich gemacht werden, um solche Verfolgbarkeitsbeziehungen ermitteln zu können. Basierend auf den solchermaßen identifizierten Entitäten und ihren Beziehungen definieren wir Regeln, welche die Änderungen und die Verfolgbarkeitsbeziehungen steuern und kontrollieren.

Eine solche Regel beschreibt damit erlaubte Operationen bei der Ableitung des Slave-Standards vom Master-Standard. Eine solche Operation könnte beispielsweise die Modifikation („Tailoring“) einer Entität sein. In diesem Fall würde die Verfolgbarkeitsbeziehung zwischen Master- und Slave-Entität vom Typ „Tailored“ (T) sein. Bei der

Analyse der Verfolgbarkeitsbeziehungen zeigt eine solche Beziehung an, dass die referenzierte Entität in beiden (Master- und Slave-) Standards enthalten ist, im letzteren jedoch modifiziert wurde.

Andere Beziehungstypen könnten beispielsweise „Tailored Out“ (TO) oder „Unchanged“ (U) sein. „U“ bedeutet dabei, dass die jeweilige Entität unverändert aus dem Master- in den Slave-Standard übernommen, „TO“, dass sie komplett entfernt wurde. Im Falle von „T“- und „TO“-Beziehungen muss üblicherweise eine Begründung gegeben werden, warum eine Entität entfernt oder verändert wird. Diese Information erfassen wir als Kommentar zur jeweiligen Beziehung.

3.2 Dokumente mit Meta-Informationen anreichern

Der nächste Schritt stellt sicher, dass die gesammelten Verfolgbarkeitsinformationen während des gesamten Lebenszyklus’ des Dokuments gültig bleiben. Da Standards üblicherweise stark strukturiert sind, erscheint die Nutzung von Abschnitten als Objekte der Verfolgbarkeit sinnvoll, führt aber zu einem Problem: Sowohl die Nummer des jeweiligen Abschnitts als auch die Überschrift können sich im Laufe der Zeit verändern. Das bedeutet, dass diese Informationen nicht zur Identifizierung von Abschnitten herangezogen werden können. Daher schlagen wir vor, in jeden Abschnitt eine Tabelle mit einigen zusätzlichen Meta-Informationen einzufügen, welche die Identifikation eines solchermaßen markierten Abschnitts zulässt.

Meta Information		
ID	<i>invariant ID of the respective section</i>	
Change log	<i>description of changes</i>	
Reviewer comments	<i>reviewer feedback</i>	
Traceability		
Master ID	Status	Comment
<i>master section ID</i>	<i>{U, T, TO}</i>	<i>comment on status</i>

Tabelle 1. Beispiel einer Tabelle mit Meta-Informationen

Tabelle 1 zeigt ein Beispiel einer solchen Tabelle mit Meta-Informationen. Die „ID“ ist eine eindeutige, unveränderliche Kennzeichnung des jeweiligen Abschnitts. Das „Change Log“ erlaubt es Autoren des Abschnitts, die vorgenommenen Veränderungen ausführlich zu beschreiben – dies ermöglicht es, alle Änderungen des kompletten Dokuments detailliert nachzuvollziehen. In die „Reviewer Comments“ schreiben die Reviewer ihre Anmerkungen zum jeweiligen Abschnitt. Der „Traceability“-Teil der Tabelle enthält schließlich Informationen darüber, auf welche Teile des Master-Dokuments sich der betrachtete Abschnitt bezieht, über die Art der Beziehung sowie Kommentare zum Status (z.B. was geändert wurde im Vergleich zum Master-Standard und warum).

3.3 Informationen für Programme identifizierbar machen

Wenn die einzelnen Abschnitte des Dokuments identifiziert werden können, besteht der nächste Schritt darin, bestimmte Informationen für Programme identifizierbar zu machen. Für Menschen ist es offensichtlich, dass ein „Erwartetes Ergebnis: Funktionale Anforderungen“ lautendes Stück Text ein bestimmtes Arbeitsprodukt beschreibt, das an dieser Stelle erzeugt werden soll – in diesem Fall die funktionalen Anforderungen. Für ein Programm ist dies allerdings nicht selbstverständlich. Daher muss ein solches Programm, das diese Informationen verarbeiten soll, zusätzliche Hilfe erhalten. Wir haben dies durch den Einsatz von Formatvorlagen (Styles) erreicht.

In modernen Textverarbeitungsprogrammen werden Formatvorlagen benutzt, um die Formatierung eines Textes zu definieren. In diesem Paper ist beispielsweise die „Überschrift 1“-Formatvorlage definiert als 12 Punkt Times New Roman fett. Diese Formatvorlage ist allen Überschriften der ersten Ebene zugewiesen, was das Textverarbeitungsprogramm anweist, alle diese Überschriften in 12 Punkt Times New Roman fett darzustellen. Während es dies einfach macht, einem Dokument ein konsistentes Aussehen zu verleihen, hat es auch einen weiteren nützlichen Nebeneffekt: Das Textverarbeitungsprogramm „weiß“, dass ein bestimmtes Stück Text die Formatvorlage „Überschrift 1“ zugewiesen bekommen hat.

Dies wiederum ist sehr hilfreich, wenn ein Programm solchermaßen formatierte Texte analysieren und Informationen daraus extrahieren soll. Beispielsweise ist es damit möglich (und in der Tat einfach), alle Überschriften erster Ebene automatisch zu extrahieren – indem einfach aller Text extrahiert wird, dem die Formatvorlage „Überschrift 1“ zugewiesen wurde. Im Fall unserer Standards haben wir weiterhin allen Inputs eine Formatvorlage namens „Input“ zugewiesen und allen Outputs eine Formatvorlage namens „Output“. Dies ermöglicht es uns, diese Informationen später automatisch zu extrahieren.

3.4 Verfolgbarkeit herstellen

Nachdem die Dokumente solchermaßen präpariert wurden, sind wir nun in der Lage, für jeden Abschnitt des Slave-Standards die Verfolgbarkeit zum Master-Standard herzustellen. Auch wenn dies eine stark manuell geprägte Tätigkeit ist, so kann doch entsprechende Werkzeugunterstützung unterstützend wirken [Lu09]. Insbesondere müssen sowohl Master- als auch Slave-Abschnitte verstanden werden, um festzustellen (a) ob sie überhaupt eine Beziehung haben und (b) falls ja, welcher Art diese Beziehung (U, T, TO) ist. Unser Ansatz unterstützt diese Analyse insofern, als dass die Autoren der Standards die Beziehungen direkt innerhalb der jeweiligen Abschnitte dokumentieren können (siehe Tabelle 1, „Traceability“-Teil) anstatt in einem anderen Dokument.

3.5 Synchrones Abbild der Standards in der Datenbank erzeugen und erhalten

An dieser Stelle sind alle Vorbereitungen innerhalb der Dokumente abgeschlossen. Der nächste Schritt ist die Erstellung eines Abbilds der Dokumente in der Datenbank. Im

XML-Format gespeichert werden Textverarbeitungsdokumente für externe Werkzeuge wie XML-Parser zugänglich. Einen solchen Parser haben wir benutzt, um die relevanten Informationen zu extrahieren. Im Wesentlichen haben wir den Parser angewiesen, für jede Aktivität (d.h. für jeden Abschnitt) die ID, die Überschrift, die Nummer der Überschrift im Dokument, das Change Log, die Reviewer-Kommentare und die Verfolgbarkeitsinformationen zu extrahieren. Darüber hinaus wurden alle Inputs und Outputs extrahiert und sämtliche Informationen in die Datenbank geschrieben. All dies konnte mit wenigen hundert Zeilen Python-Code [Lut01] erreicht werden. Das Ergebnis war eine 1:1-Repräsentation des Standards in unserer Datenbank – bereit zur Analyse. Nach jeder Änderung des Standards wird das Datenbank-Abbild aktualisiert, so dass Standard und Abbild immer synchron sind.

3.6 Aggregieren, analysieren und visualisieren

Nun, da die Essenz unseres Standards in einer durch Programme interpretierbaren Form vorliegt, stellt sich die Frage, was wir damit anfangen können? Zum Beispiel Daten aggregieren. Fragen wie „Welche Outputs werden während der Anforderungserfassungsphase erzeugt?“ können durch wenige Datenbankabfragen beantwortet werden. Weiterhin ist es möglich, einen schnellen Überblick über den Grad der Veränderungen des Standards zu bekommen (welche Teile wurden stark modifiziert, welche wurden entfernt, ...). Weiterhin können wir vielfältige Prüfungen bezüglich Inkonsistenzen durchführen, beispielsweise Outputs finden, die benötigt, aber nie produziert werden, oder Visualisierungen des Standards erstellen (siehe Abbildung 1).

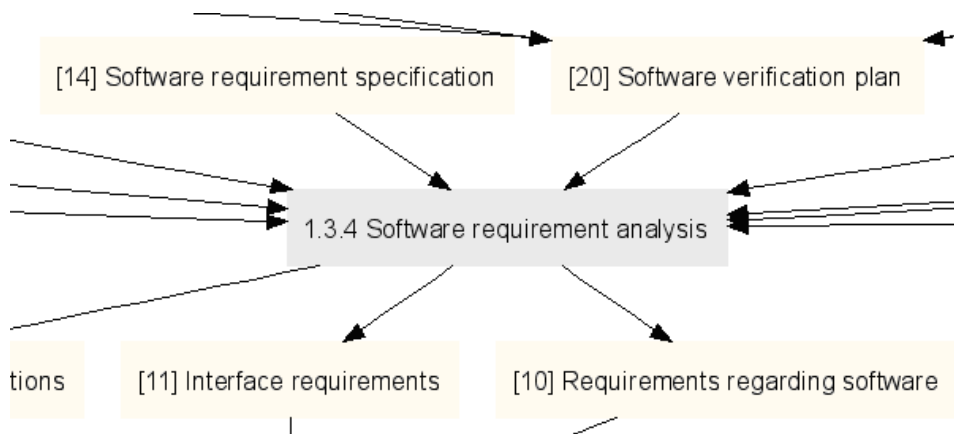


Abbildung 1. Beispiel der Visualisierung eines Standards

4 Erwarteter Nutzen

Wir erwarten verschiedenartigen Nutzen von unserem Ansatz. Erstens erwarten wir, dass die Autoren der Standards ihren gewohnten Ansatz zur (Weiter-) Entwicklung der Standards weiter verwenden können, d.h. die Verwendung von Microsoft Word oder OpenOffice.org Writer. Dies ist besonders wichtig, da die Autoren hauptsächlich diese Werkzeuge benutzen. Es ist kaum möglich, die Verwendung anderer, spezialisierter Werkzeuge zu erzwingen, da nachgelagerte Aktivitäten wie Reviews oder der weltweite Informationsaustausch empfindlich gestört würden.

Zweitens erwarten wir signifikante Vorteile bezüglich der internen Konsistenz (d.h. innerhalb einzelner Standards). Beispielsweise kann ein Programm, wenn es bestimmte Arbeitsprodukte und ihre Beziehungen untereinander erkennt, auch feststellen, ob bestimmte verbotene Zustände existieren oder nicht. Daher erwarten wir, dass wir die Abwesenheit bestimmter verbotener Zustände durch automatische Tests nachweisen können.

Drittens erwarten wir ähnliche Vorteile bezüglich der externen Konsistenz (d.h. zwischen verschiedenen Standards). Beispielsweise erwarten wir, garantieren zu können, dass jede Entität des Master-Standards betrachtet wurde, und dass wir für jede Entität im Slave-Standard garantieren können, dass ihr Verhältnis zur jeweiligen Master-Entität dokumentiert ist. Kurz gesagt erwarten wir, dass wir keine Entität im Zuge des Standard-Tailorings „verlieren“, d.h. sie ohne klare Begründung entfernt oder verändert wird.

Schließlich verspricht der datenbankbasierte Ansatz weiteren Nutzen, zum Beispiel Transparenz bezüglich des Tailoring-Grades (wie viele Entitäten wurden entfernt, hinzugefügt, modifiziert) oder bezüglich der Auswirkungen von Änderungen an einem Standard auf andere, abgeleitete Standards. Insgesamt erwarten wir eine höhere Qualität der Standards durch automatisierte Konsistenzprüfungen und dadurch zielgerichtete Reviews durch Menschen.

5 Fallstudien

Wir haben den beschriebenen Ansatz in zwei verschiedenen Umgebungen eingesetzt. Die erste Anwendung fand beim European Space Operations Centre (ESOC) statt, der für die Bodenanlagen (Ground Segment) zuständigen Organisation der Europäischen Raumfahrtagentur ESA. Die zweite Anwendung führten wir bei der Japanischen Raumfahrtagentur JAXA durch.

5.1 European Space Operations Centre (ESOC)

In diesem Fall haben wir die Verfolgbarkeit zwischen den ESA-Softwarestandards (ECSS E-40b, Q-80 und einer Reihe von M-Standards) und dem davon abgeleiteten ESOC-Prozessstandard für die Entwicklung und Wartung von Bodenanlagen herge-

stellt. Die ECSS-Standards formulieren Anforderungen, die von allen ESA-Prozessen erfüllt werden müssen (in unserem Fall etwa 1600), und standen uns in Form von PDF-Dokumenten zur Verfügung.

Der ESOC-Prozessstandard trägt den Namen SETG (Tailoring of ECSS Software Engineering Standards for Ground Segments in ESA) [ESA05]) und besteht aus zwei OpenOffice.Org-Writer/Microsoft-Word-Dokumenten, welche die Aktivitäten und Arbeitsprodukte beschreiben (SETG A und B), einem weiteren Dokument mit Vorlagen für verschiedene Arbeitsprodukte (SETG C) und einem Dokument, welches alle Verfolgbarkeitsinformationen enthält (SETG D). Die Verfolgbarkeit erstreckte sich von den ECSS-Anforderungen zu SETG A und B, welche zusammen etwa 110 Seiten umfassen. Bezüglich ECSS wurde jede Anforderung erfasst, zusammen mit den erwarteten Outputs und ihren Reviews. Auf SETG-Seite wurden Aktivitäten (z.B. „Determine Interface Requirements“) und Outputs erfasst. Damit konnte ESOC für jede ECSS-Anforderung und jeden erwarteten Output die SETG-Aktivität und den jeweiligen SETG-Output benennen, welche die ECSS-Anforderungen implementieren.

5.2 Japan Aerospace Exploration Agency (JAXA)

Die zweite Anwendung fand bei der Japanischen Raumfahrtagentur JAXA statt. JAXA unterhält eine hierarchische Standardarchitektur. Aus Level-1-Standards wie ISO/IEC 12207 [Int95] oder ECSS wird der generische JAXA-Level-2-Softwareentwicklungsstandard abgeleitet. Der Level-2-Standard wiederum dient als Basis für eine Reihe von domänenspezifischen Level-3-Standards, beispielsweise für Satelliten, Transportsysteme oder Bodenanlagen. Jeder domänenspezifische Standard wird dann weiter für jedes Projekt angepasst (Level-4). Bisher haben wir an Level-2- und Level-3-Standards gearbeitet, mit einer geplanten Ausweitung auf Level-1- und Level-4-Standards.

Alle JAXA-Level-2- und Level-3-Standards standen als Microsoft-Word-Dokumente zur Verfügung. Die Verfolgbarkeit haben wir für jede Aktivität und jedes Arbeitsprodukt (Input/Output) ermittelt. Da das Projekt noch nicht vollständig abgeschlossen ist, sind die berichteten Erfahrungen vorläufig, geben allerdings bereits interessante Hinweise.

5.3 Erfahrungen

Bezüglich des erwarteten Nutzens wurde unsere erste Erwartung (Weiternutzung von Word) in vollem Umfang erfüllt. Die Autoren der Standards empfanden die zusätzliche Metadatentabelle zu Beginn jedes Abschnitts und die leicht veränderte Art und Weise der Dokumentation von Inputs und Outputs nicht als Beeinträchtigung ihrer täglichen Arbeit. Insbesondere die Tatsache, dass sämtliche Verfolgbarkeitsinformationen direkt im jeweiligen Abschnitt bzw. beim jeweiligen Arbeitsprodukt notiert wurden, war deutlich anwenderfreundlicher als die Nutzung von (externen) Tabellen oder ähnlichen Lösungen. Der Verzicht auf Spezialwerkzeuge führte weiterhin dazu, dass der erweiter-

te Bearbeitungsprozess (internationale Reviews, Informationsaustausch etc.) unverändert ablaufen konnte.

Unsere zweite Erwartung (interne Konsistenz) wurde ebenfalls vollständig erfüllt. So haben beispielsweise automatische Konsistenzprüfungen eine Reihe von Inkonsistenzen im Produktfluss aufgezeigt, etwa Arbeitsprodukte, die als Input verlangt werden, jedoch an keiner Stelle auch als Output produziert werden, oder umgekehrt. Der Versuch der *manuellen* Erkennung *aller* dieser Inkonsistenzen in einem Standard mit Dutzenden von Aktivitäten auf mehreren Hierarchieebenen und Hunderten von Inputs und Outputs würde extremen Aufwand bedeuten und höchstwahrscheinlich doch einige vorhandene Probleme übersehen. Die automatisierte Konsistenzprüfung wiederum nahm nur einige Sekunden in Anspruch und hat *alle* diese Inkonsistenzen erkannt – und ersparte damit den menschlichen Reviewern diese langweilige und ermüdende Tätigkeit.

Unsere dritte Erwartung schließlich (externe Konsistenz) hat uns ebenfalls nicht enttäuscht. Durch Analysen der Datenbank konnten wir tatsächlich nachweisen, dass jede Entität des Master-Standards berücksichtigt wurde (d.h. wenigstens eine eingehende Verfolgbarkeitsbeziehung vom Typ T, TO oder U hat). Auf ähnliche Weise konnten wir für jede Slave-Entität nachweisen, dass ihre Verfolgbarkeitsbeziehungen zum Master-Standard ermittelt wurden (d.h. dass sie wenigstens eine ausgehende Verfolgbarkeitsbeziehung vom Typ T, TO oder U hat). Diese detaillierte Verfolgbarkeit zwischen Master- und Slave-Standards hat die Freigabe der SETG-Standards durch das Engineering Standardization Board (ESB) der ESA deutlich erleichtert. Das JAXA-Projekt ist noch nicht an diesem Punkt, wir erwarten hier jedoch ähnliche Effekte.

Über diese Punkte hinaus konnten wir noch einige weitere nützliche Effekte feststellen, die wir hier nur kurz erwähnen wollen. Während des JAXA-Projekts erzeugten wir (automatisch) einfache Visualisierungen der standardinternen und -externen Beziehungen (siehe Abbildung 1). Dies half den JAXA-Ingenieuren, die Komplexität der Standards besser zu verstehen und zu berücksichtigen: Sobald an einem Standard eine Veränderung vorgenommen wurde, konnten wir umgehend die Abschnitte der anderen Standards benennen, die aufgrund dieser Änderung ebenfalls einem Review unterzogen werden mussten. Die garantierte Abwesenheit bestimmter Inkonsistenzen und damit die Möglichkeit, Reviews besser zu fokussieren, erhöhte die Gesamtqualität der Standards – was allerdings schwer zu quantifizieren ist. Weiterhin erleichterte die Integration der Kommentarfelder für die Reviewer in den jeweiligen Abschnitt die Sammlung und Aufbereitung der Reviewkommentare.

6 Zusammenfassung und Ausblick

Wir haben in diesem Paper einen systematischen, den gewohnten Arbeitsablauf nur wenig beeinträchtigenden Ansatz für die Etablierung und Erhaltung der Verfolgbarkeit zwischen als Textverarbeitungsdokumente vorliegenden Softwareprozessstandards vorgestellt. Durch die Anreicherung dieser Dokumente mit Meta-Informationen und die Kenntlichmachung bestimmter Informationen für Programme sind wir in der Lage,

eine Repräsentation der Standards in einer Datenbank zu erzeugen, die wir für weitergehende Analysen und Visualisierungen nutzen können.

Der Ansatz hat unsere Erwartungen vollständig erfüllt und sogar darüber hinausgehenden Nutzen gezeigt. Die dokumentbasierten Verfolgbarkeitsinformationen waren dabei besonders hilfreich und benutzerfreundlich. In Bezug auf die Mächtigkeit der Analysen der Datenbankrepräsentation der Standards haben wir gerade erst begonnen, die Möglichkeiten auszuloten. Automatisierte Konsistenzprüfungen können Reviewern lästige und fehlerträchtige Arbeiten abnehmen, sodass sie sich auf Sachverhalte konzentrieren können, bei denen echtes menschliches Verständnis notwendig ist.

Die Fallstudien zeigen, dass der Ansatz auch für große Standards skaliert wie beispielsweise die des Raumfahrtbereichs. Ausgehend von unseren Erfahrungen erwarten wir, dass der Ansatz in anderen Domänen mit entsprechend dokumentierten Prozessen ebenfalls funktioniert. In Zukunft werden wir die Möglichkeiten, die die Datenbankrepräsentation bietet, weiter analysieren und insbesondere die Unterstützung der Standardautoren durch mehr und bessere Reports und Visualisierungen ausweiten.

Eine interessante Erweiterungsmöglichkeit bietet der *DeltaProcess*-Ansatz zum Vergleich von Prozessmodellen von einem der Autoren dieses Papers [SM06], [SM07]. Der *DeltaProcess* zugrundeliegende Algorithmus und die zugehörigen Softwarewerkzeuge bestimmen detailliert die Unterschiede zwischen verschiedenen Versionen von Prozessmodellen wie etwa den in diesem Paper beschriebenen Standards.

Der *DeltaProcess*-Ansatz überträgt dabei (1) Prozessmodelle in eine RDF-Notation (Resource Description Framework [MM04]), führt (2) einen Vergleich der RDF-Modelle durch und erlaubt dann (3) die Identifizierung aller Veränderungen durch die Analyse spezifischer struktureller Muster (Patterns) des Vergleichsergebnisses. Dadurch kann ein Prozessexperte mit *DeltaProcess* zuverlässig und automatisiert diejenigen Änderungen identifizieren, die eine Version eines Standards in eine andere überführen. Der Ansatz wurde beispielsweise in einer detaillierten Studie der Evolution des V-Modells XT angewandt, wobei jeweils zwei aufeinander folgende Versionen des V-Modells XT miteinander verglichen wurden, um die erfolgten Änderungen zu identifizieren [SOM08]. Durch die Visualisierung der so gewonnenen Informationen konnten wichtige Erkenntnisse über die (Weiter-) Entwicklung des V-Modells XT erworben werden.

DeltaProcess kann damit eingesetzt werden, um die Änderungen des Slave-Standards gegenüber seinem Master-Standard zu identifizieren, falls dies nicht bereits bei der Ableitung aufgezeichnet wurde. Damit können für die unveränderten Teile die Verfolgbarkeitsbeziehungen automatisch gesetzt werden, und nur die veränderten Teile müssen von Menschen analysiert werden – dies verspricht, abhängig vom Anteil der unveränderten Teile, eine signifikante Reduzierung des erforderlichen Aufwands.

DeltaProcess kann darüber hinaus dabei helfen, mit Änderungen sowohl im Master- als auch im Slave-Standard umzugehen, deren Verfolgbarkeitsbeziehungen früher bereits etabliert, aber nicht mit den Änderungen aktualisiert wurden. Offensichtlich bergen Änderungen in einem Modell immer die Gefahr, existierende Verfolgbarkeitsbeziehun-

gen zu anderen Modellen zu invalidieren oder neue erforderlich zu machen. Das bedeutet, dass die Verfolgbarkeitsbeziehungen während oder nach jeder Änderung erneuert, zumindest aber überarbeitet werden müssen. Da dies jedoch zeitintensiv und damit teuer ist, geschieht es nicht immer, sodass sich die Frage stellt, wie die Verfolgbarkeitsbeziehungen effizient wieder mit dem aktuellen Standard synchronisiert werden können. Auch hier kann *DeltaProcess* hilfreich sein.

Durch die Nutzung von *DeltaProcess* zum Vergleich des alten, mit Verfolgbarkeitsbeziehungen versehenen Standards mit der neuen Version werden die vorgenommenen Änderungen unmittelbar sichtbar. Dadurch können sich Prozessexperten und Reviewer auf die geänderten Teile konzentrieren und müssen auch nur für diese Teile die Verfolgbarkeitsbeziehungen überprüfen bzw. aktualisieren. Da der Anteil der Änderungen im Vergleich zum kompletten Standard üblicherweise klein ist, sorgt diese Fokussierung für eine signifikante Senkung des benötigten Aufwands. Tatsächlich können ganze Teile des Standards automatisch und ohne schädliche Nebenwirkungen von der Überprüfung der Verfolgbarkeitsbeziehungen ausgenommen werden – eben jene Teile, die nicht verändert wurden, und für die damit die bisherigen Verfolgbarkeitsbeziehungen weiter gültig sind. Da alle hierfür benötigten Informationen formal modelliert sind, könnte ein Verfolgbarkeitswerkzeug basierend auf *DeltaProcess* diese Teile automatisch und verlässlich identifizieren.

Da die Nutzung eines Vergleichswerkzeugs wie *DeltaProcess* eine deutliche Erweiterung der in diesem Paper beschriebenen Methoden darstellt, sind noch einige technische Herausforderungen zu meistern, bevor ein solcher integrierter Ansatz praktisch nutzbar ist. Wir erwarten jedoch, in naher Zukunft eine prototypische Integration der Ansätze vorstellen zu können.

Danksagungen

Wir danken Sonnhild Namingha für die Durchsicht dieses Papers. Weiterhin danken wir den anonymen Reviewern für ihre Kommentare, die dazu beigetragen haben, dieses Paper zu verbessern.

Literaturverzeichnis

- [Ar08] Armbrust, O., Katahira, M., Miyamoto, Y., Münch, J., Nakao, H., Ocampo, A.: Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards. In (Wang, Q., Pfahl, D., Raffo, D. M. Hrsg.) Proceedings of the International Conference in Software Process (ICSP2008), May 10-11, 2008, Leipzig, Germany, Lecture Notes in Computer Science 5007; S. 160-172.
- [AO09] Armbrust, O., Ocampo, A.: Software Process Lines and Standard Traceability Analysis. Proceedings of the 7th Workshop of Critical Software, January 14-15, 2009, Tokyo, Japan.

- [Ar09] Armbrust, O., Ocampo, A., Münch, J., Katahira, M., Koishi, Y., Miyamoto, Y.: Establishing and Maintaining Traceability Between Large Aerospace Process Standards. Proceedings of 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE'09), May 18, 2009, Vancouver, Canada; S. 36-40.
- [CMC05] Cendral, J., Merri, M., Choda, R.: Engineering Standardization, 2005.
- [Ca01] Cass, A., Völcker, C., Winzer, L., Carranza, J., Dorling, A.: SPICE for SPACE: A Process Assessment and Improvement Method for Space Software Development. European Space Agency Bulletin 107, 2001.
- [ESA05] BSSC Guides and Reports, http://www.esa.int/TEC/Software_engineering_and_standardisation/, last visited 2009-04-25.
- [Eur09] Collaboration website of the European Cooperation for Space Standardization, <http://www.ecss.nl/>, last visited 2009-4-25.
- [IBM09] IBM Rational Requisite Pro, <http://www-01.ibm.com/software/awdtools/reqpro/>, last visited 2009-06-18.
- [Int95] International Organization for Standardization: ISO/IEC 12207:1995. ISO/IEC, Geneva, Switzerland, 1995.
- [Koo08] Das V-Modell XT 1.2.1.1, <http://www.vmodellxt.de/>, last visited 2009-01-15.
- [Lut01] Lutz, M.: Programming Python, 2nd Edition. O'Reilly & Associates, Sebastopol, California, USA, 2001.
- [MM04] RDF Primer, W3C Recommendation, <http://www.w3.org/TR/rdf-primer/>, last visited 2009-06-19.
- [NAS09] NASA Technical Standards Program, <http://standards.nasa.gov/default.taf>, last visited 2009-01-21.
- [PB03] Palmirani, M., Brighi, R.: Metadata for the legal domain. Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03), September 1-5, 2003, Prague, Czech Republic; S. 553- 558.
- [Rom05] Rombach, H. D.: Integrated Software Process and Product Lines. In (Li, M., Boehm, B., Osterweil, L. J. Hrsg.) Proceedings of the International Software Process Workshop (SPW 2005), Beijing, China, Lecture Notes in Computer Science 3840/2006; S. 83-90.
- [SM06] Soto, M., Münch, J.: Process Model Difference Analysis for Supporting Process Evolution. Proceedings of the European Systems & Software Process Improvement and Innovation Conference (EuroSPI2 2006), Joensuu, Finland, Lecture Notes in Computer Science 4257; S. 123-134.
- [SM07] Soto, M., Münch, J.: Focused Identification of Process Model Changes. Proceedings of the International Conference on Software Process (ICSP 2007), Minneapolis, MN, USA, May 19-20, 2007; S. 182-194.
- [SOM08] Soto, M., Ocampo, A., Münch, J.: The Secret Life of a Process Description: A Look Into the Evolution of a Large Process Model. In (Wang, Q., Pfahl, D., Raffo, D. M. Hrsg.) Proceedings of the International Conference on Software Process (ICSP 2008), May 10-11, 2008, Leipzig, Germany, Lecture Notes in Computer Science 5007; S. 257-268.

- [Tel09] Telelogic Doors, <http://www.telelogic.com/>, last visited 2009-01-10.
- [Zh06] Zhang, Y., Witte, R., Rilling, J., Haarslev, V.: An Ontology-based Approach for the Recovery of Traceability Links. Proceedings of the 3rd International Workshop on Metamodels, Schemas, Grammars, and Ontologies for Reverse Engineering (ATEM), October 1st, 2006, Genoa, Italy.
- [Lu04] de Lucia, A., Fasano, F., Olivetto, R., Tortora, G.: Enhancing an artefact management system with traceability recovery features. Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM), September 11th - 17th, 2004, Chicago, IL, USA.
- [Lu09] de Lucia, A., Oliveto, R., Tortora, G.: Assessing IR-based traceability recovery tools through controlled experiments. Empirical Software Engineering, Springer Netherlands 14 (1), 2009; S. 57-92.