

Architektur und Entwicklung eines Modellierungswerkzeuges zur kollaborativen synchronen Konstruktion von BPMN-Modellen

Daniel Hilpoltsteiner¹, Markus Schmidner, Christian Seel †, Holger Timinger

Zusammenfassung: Aktuell ist ein klarer Trend zu Kollaborationsfunktionen in Softwarelösungen erkennbar. Dies macht auch vor dem Bereich der Prozessmodellierungswerkzeuge nicht halt. Jedoch ist die Softwareunterstützung in diesem Bereich noch nicht sehr stark ausgeprägt. Daher werden im Rahmen dieses Beitrages umzusetzende Anforderungen an ein kollaboratives Werkzeug zur Modellierung von BPMN-Prozessmodellen erhoben und eine Architektur konzipiert. Zudem wird die Anwendbarkeit der kollaborativen Konstruktion von Prozessmodellen mit Hilfe eines Experiments dargelegt. Fokussiert wurde dabei die Übertragung von Änderungen im Prozessmodell über ein MQTT-Protokoll sowie die Latenzen zwischen verschiedenen Systemen.

Keywords: Kollaboration, BPMN 2.0, Prozessmodellierung, Softwareunterstützung

1 Motivation

Während Software um die Jahrtausendwende noch primär lokal installiert und on-premise vermarktet wurde, so hat sich in den letzten Jahren ein deutlicher Trend hin zum Software-as-a-Service(SaaS)-Modell entwickelt [WA16]. Zu den bekanntesten Vertretern gehören hierbei unter anderem die Produkte Google Docs, Microsoft Office 365 und Etherpad, welche ein synchrones Bearbeiten von Dokumenten ermöglichen. Ein Trend zur Kollaboration während der Prozessmodellkonstruktion ist hierbei ebenfalls zu erkennen. Allerdings implementieren Anbieter nur bedingt die Möglichkeit der verteilten Konstruktion von BPMN-Modellen. Hier setzt die Implementierung von ADAMO (ADAPtives MOdellierungswerkzeug) mit dem Kommunikationsansatz über MQTT, zum dezentralen Datenaustausch zwischen den Clients während der Kollaboration, an. Hierfür werden in diesem Beitrag Fragen zu Anforderungen an Kollaboration, der grundsätzlichen Architektur und der erreichbaren Geschwindigkeit des Datenaustausches beantwortet. Die folgenden Forschungsfragen sollen zur Klärung beitragen:

RQ1: Welche Anforderungen bestehen an die synchrone kollaborative Informationsmodellierung in Echtzeit?

RQ2: Welche Architektur eignet sich für ein System zum echtzeitbasierten Austausch von Informationen in BPMN-Modellen?

¹ Hochschule Landshut, Institut für Projektmanagement und Informationsmodellierung, Am Lurzenhof 1, 84036 Landshut, {daniel.hilpoltsteiner|markus.schmidner|holger.timinger}@haw-landshut.de

RQ3: Wie kann der Datenaustausch performant abgebildet werden und ist die Übertragungszeit gering genug, um kollaboratives Arbeiten zu ermöglichen?

Im folgenden Kapitel wird der Stand der Technik dargestellt. Des Weiteren werden in diesem Beitrag Anforderungen erhoben und genutzt, um eine Konzeption der Architektur sowie deren Implementierung aufzuzeigen. Die gewonnenen Erkenntnisse werden am Ende des Beitrages evaluiert.

2 Stand der Technik

Da sich der Begriff der Kollaboration - je nach dem Betrachtungswinkel des jeweiligen Forschungsgebietes - unterscheiden kann, wird dieser im Folgenden definiert. Der Beitrag folgt der Definition von LEIMEISTER [LE14], da sie mit Bezug auf das Collaborative Engineering getroffen wurde. Da Kommunikation, Koordination und Kooperation natürlich in großen Maßen von menschlichem Handeln geprägt sind, kann ein Modellierungswerkzeug alleine dieses Verhalten nicht sicherstellen. Es kann jedoch Funktionalitäten bereitstellen, die den Anwender in seinem Vorhaben unterstützen. So kann ein Modellierungswerkzeug die Kommunikation der Anwender verbessern, indem es die Möglichkeit bereitstellt, Daten über eine Netzwerkverbindung hinaus mit anderen Anwendern auszutauschen. Der Koordinationsaspekt wird dadurch unterstützt, dass die ausgetauschten Daten automatisiert zusammengeführt werden, ohne dass die beteiligten Anwender dies direkt initiieren müssten. Da zudem durch das Modellierungswerkzeug eine Möglichkeit zur zeitgleichen Arbeit an verschiedenen Orten geschaffen wird, unterstützt es dadurch die Anwender bei der mittelbaren Kooperation. Andere Modellierungswerkzeuge im Bereich der BPMN-Prozessmodellierung erlauben zwar auch die Kollaboration bei der Konstruktion. Allerdings ist der Ansatz bei den meisten Tools zentralisiert und der Datenaustausch findet nicht in Echtzeit über beispielsweise MQTT statt. In der Literatur stellt die Computer Supported Collaborative Work (CSCW) Aspekte vor die zur gemeinsamen Lösung einer Aufgabe relevant sind [SC01]. Einige Anforderungen, die im Rahmen der CSCW aufkamen, sind auch in den Bereich der kollaborativen Software übertragbar. Vor allem sind diese Anforderungen interessant für die Konzeption eines Softwarewerkzeuges zur kollaborativen Informationsmodellierung. Dabei wird zwischen synchronen und asynchronen Anwendungen unterschieden. [SC01] Bei synchronen Anwendungen ist es üblich, dass die Anwender zeitgleich am selben Objekt arbeiten und sich gegebenenfalls auch direkt in ihrer Arbeit beeinflussen können [HO01]. Im Gegensatz hierzu stehen die asynchronen Anwendungen, bei denen die Arbeitsschritte sequenziell von den einzelnen Anwendern ausgeführt werden. Zeitgleiches Arbeiten am selben Objekt wird nicht oder nur in sehr geringem Umfang unterstützt. [AP01] Das Vorhaben einer Echtzeitkollaboration in BPMN-Modellierungswerkzeugen ist also der Kategorie der synchronen CSCW-Anwendungen zuzuordnen.

3 Anforderungen und Konzeption

Da die Neukonzeption und Implementierung von Prototypen im akademischen Umfeld mit erheblichem Aufwand verbunden ist und gegenüber einer Wiederverwendung von existierenden Lösungen nur einen geringen Mehrwert bietet, wurde für die Implementierung von ADAMO² auf quelloffene Lösungen zurückgegriffen. Dabei wurde auf die Bibliothek bpmn-js aufgebaut und ein Administrationsbereich, in dem das Benutzer- und Rollenmanagement integriert ist, sowie eine Modellverwaltung implementiert. Der bisherige Entwicklungsstand beinhaltete Funktionalitäten zur Konstruktion adaptiver BPMN-Prozessmodelle [HI18]. Mit der Zeit wurden in Zusammenarbeit mit Projektpartnern und Modellierungsexperten aus kooperierenden kleinen und mittelständischen Unternehmen Anforderungen zur Kollaboration in der BPMN-Prozessmodellerstellung erkannt. Diese Anforderungen erweitern bisherige Anforderungen an ADAMO und sind in Form von User-Stories in Tabelle 1 dargestellt (RQ1).

User-Stories	Beschreibung
User-Story 1	Als Modellersteller möchte ich ein Diagramm in einer Datenbank speichern können, um es mit anderen zu teilen.
User-Story 2	Als Modellersteller möchte ich, dass Änderungen im Diagramm automatisch und zeitnah (< 500ms) an andere Anwender weitergeleitet werden, um Sie allen zur Verfügung zu stellen.
User-Story 3	Als Modellersteller möchte ich, dass meine Änderungen zeitnah mit den Anpassungen anderer zusammengeführt werden, um ein konsistentes Modell zu erhalten.
User-Story 4	Als Modellersteller möchte ich bei Fehlern in der Zusammenführung eine Meldung erhalten, um über weitere Schritte entscheiden zu können.
User-Story 5	Als Nutzer eines Modells möchte ich die Anzahl und die Namen der aktuell am Dokument arbeitenden Anwender angezeigt bekommen.

Tab. 1: Anforderungen zur Kollaboration an ADAMO als User-Stories

Die User-Stories 1-3 korrelieren hierbei mit dem Konzept des gemeinsamen Objektes aus der CSCW. Sie zielen darauf ab, dass es den Anwendern ermöglicht wird zeitgleich am selben Objekt zu arbeiten. Des Weiteren unterstützen die User Stories 2 und 3 zudem das Konzept der losen Koppelung und die kurze Reaktionszeit. User Story 4 weist einen direkten Bezug zur Fehlertoleranz auf, sodass bei Konflikten in der Konsistenzbildung entsprechende Maßnahmen ergriffen werden können. Die im CSCW geforderte Group

² Link zu ADAMO: <https://github.com/HAWMobileSystems/adamo>

Awareness wird durch die User-Story 5 abgebildet. Durch das Einblenden der Anzahl anderer Teilnehmer sowie deren Namen kann die Kommunikation im Team unterstützt werden. In diesem Abschnitt wurde die Forschungsfrage 1 (RQ1) beantwortet, indem Anforderungen an synchrone kollaborative Modellierung in Echtzeit definiert wurden.

4 Architektur und Implementierung

Verschiedene Studien aus dem Bereich „Internet of Things“ können herangezogen werden, da ähnliche Anforderungen an die Architektur gelten wie bei der kollaborativen Modellierung. DIZDAREVIĆ et al. [DI19] vergleichen verschiedene Kommunikationsprotokolle auf deren Eignung im IoT-Umfeld. Hervorzuheben ist dabei das MQTT-Protokoll, welches durch Performanz und vor allem Akzeptanz bei den Software-Entwicklern hervorsticht. Insbesondere wurde in verschiedenen herangezogenen Studien eine vergleichsweise geringe Latenz des MQTT-Protokolls aufgezeigt. [DI19] Es basiert auf einem Publish-Subscribe-Architekturmuster. Das Protokoll ermöglicht allen Clients sowohl Daten zu veröffentlichen (publish) als auch Themen abonnieren (subscribe) zu können. Die Kommunikation geschieht dabei über einen Server der auch „Broker“ genannt wird. Dadurch wird eine lose Kopplung zwischen den einzelnen ADAMO-Clients erreicht. Ein weiterer Vorteil von MQTT ist die themenbezogene Kommunikation über sogenannte Topics. So kann eingegrenzt werden, welche Informationen ein Abonnent erhält. Das MQTT-Protokoll findet in ADAMO Anwendung auf Kollaborations-Ebene. Dabei registrieren sich die Modellierungsoberflächen als Clients auf einem Topic, welches durch eine eindeutige ID des Informationsmodells identifizierbar ist. Änderungen am Modell werden veröffentlicht und an den Broker kommuniziert, welche weiter an alle Abonnenten verteilt. Diese erhalten zeitnah die Änderungen am Modell und binden diese in die Modellierungsoberfläche ein. Im Abschnitt der Evaluation wird auf diesen Punkt noch tiefer eingegangen, da die Geschwindigkeit der Übertragung maßgeblich die Nutzbarkeit der kollaborativen Modellierung beeinflusst (RQ3). Ein Client schickt dabei als Nachrichtinhalt die XML-Repräsentation des BPMN-Modells an den Broker. Sobald die Nachricht an den anderen Clients angekommen ist, wird das neue Modell in die Modellierungsoberfläche geladen, wodurch nach der CSCW ein optimistisches Verfahren zur Konsistenzsicherung implementiert wurde. Neue Clients, die sich auf ein Topic registrieren, bekommen den letzten Stand der über das MQTT-Protokoll kommuniziert wurde. Gleichzeitig verlassen Clients, die sich in einem inkonsistenten oder fehlerhaften Zustand befinden, diesen mit der nächsten Veröffentlichung einer Nachricht. Hierdurch ist es möglich eine schnelle Reaktionszeit für die einzelnen Anwender zu erzielen und die Fehlertoleranz zu erhöhen. Obwohl das Publish-Subscribe-Architekturmuster im Normalfall anonym arbeitet, wurde auf einer höheren Ebene bewusst eine Übermittlung der Client-ID eingeführt. Hierdurch ist es möglich zu verfolgen, welche Anwender derzeit welche Objekte bearbeiten. Die ursprüngliche Konzeption der Übertragung sah vor, nur die neue Differenz zum bestehenden Modell zu übertragen. Da aber die bereitgestellten

Informationen der bpmn-js Bibliothek keine vollständige Rekonstruktion der durchgeführten Änderungen ermöglichen, wurde an dieser Stelle das gesamte Modell übertragen. In diesem Abschnitt des Beitrages wurde die Forschungsfrage 2 (RQ2) beantwortet, indem eine Architektur für ein kollaboratives Modellierungswerkzeug vorgestellt wurde. Zudem wurde zur Beantwortung von Forschungsfrage 3 (RQ3) die zu nutzende Technologie vorgestellt. Die Implementierung des Prototyps wird im Folgenden in der Evaluierung verwendet, um die Relevanz zur Beantwortung von Forschungsfrage 3 (RQ3) zu überprüfen.

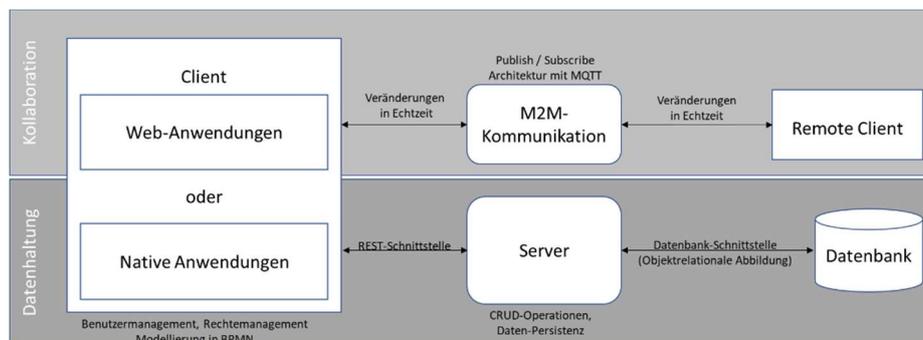


Abb. 1 Architektur zum Datenaustausch von ADAMO über Publish-Subscribe und REST

5 Evaluation

Ein wichtiger Bestandteil von ADAMO ist die Möglichkeit auf verschiedenen Systemen kollaborativ an einem Modell zu arbeiten. Für den Test wurden zwei Laptops und ein virtueller Server im LAN-Netzwerk genutzt. Beide Laptops nutzten Windows 10 als Betriebssystem und befanden sich im selben Netzwerk. Ein Laptop wurde zum Veröffentlichen von Änderungen (publish) genutzt, der andere stellte einen passiven Client dar. Zwischen den Endgeräten und dem Server lagen im Netzwerk 2 Zwischenstationen (Hops). Die Serverkomponente stellte eine Linux VM mit Ubuntu 18.04 dar. Das ADAMO-Backend wie auch der MQTT-Broker wurden mithilfe eines Prozessüberwachungstools zur besseren Auswertung gestartet. Zur Erhebung der Kommunikations-Daten wurden Zeitstempel gesammelt und diese mit der jeweiligen Aktion als Label markiert. Für die Zeitstempel gibt es einige wichtige Zeitpunkte, die dokumentiert werden sollten. Dazu zählt der Zeitpunkt der Änderung am Client A (publish), der Erhalt der Information an Client B und C (subscribe), das Echo der eigenen Information (subscribe-echo) und der Zeitstempel am Server.

Da in ADAMO ein optimistisches Verfahren zur Konsistenzprüfung eingesetzt wird und deshalb der vollständige XML-String als Nachricht übertragen wird, spielt die

Übertragungsgeschwindigkeit eine maßgebliche Rolle. Spannend ist auch die Betrachtung von Nachrichten die größer als 4000 Bytes betragen, da sie die Anzahl der Übertragungspakete erhöht, die zwischen den Clients ausgetauscht werden. Ergebnisse zeigen hier, dass die zunehmende Größe zu höheren Ende-zu-Ende-Verzögerungen führen. Gleichzeitig zeigte eine Studie, dass bei zunehmender Nachrichtengröße der Nachrichtenverlust in der Übertragung ansteigt, welche durch die Nutzung von Quality-of-Service (QoS) 2-Level wieder reduziert werden kann. Unterlagen der IETF zeigen eine Reduktion des Durchsatzes von bis zu 40 % mit höherer QoS wie auch einer Verlangsamung der Übertragungsgeschwindigkeit um bis zu 75 % [LA18] aufgrund eines 4 Wege-Handshakes. Jedoch zeigte die Studie auch, dass die Latenzzeit im Netzwerk in Verbindung mit der Nachrichtengröße den Nachrichtenverlust beeinflusst. Vor allem bei geringeren Latenzen wirkt sich dies stärker aus als bei hohen Latenzen [LE13]. Da die Größe der übertragenen Daten also Einfluss auf die Übertragungsdauer hat, wurde eine passende Musterdatei verwendet. Um die durchschnittliche Dateigröße zu ermitteln wurden 200 BPMN Dateien auf den Servern einer Organisation ausgewertet. Der Mittelwert der Dateigröße betrug dabei 28 Kilobyte. Aus diesem Grund wurde zu Testzwecken ein mit diesem Mittelwert vergleichbares Prozessmodell ausgewählt. Die mehrfache Wiederholung des Experiments gewährleistet eine Reproduzierbarkeit der Ergebnisse. Nach mehreren Durchläufen wurde bei den Messungen zeitliche Unstimmigkeiten identifiziert, die durch die Synchronisation der Geräte an einem Zeitserver gelöst wurde. Die zeitliche Abweichung der Clients bei der Wiederholung des Experiments betrug 81 Millisekunden und alle Werte wurden entsprechend der Zeitdifferenz bereinigt. In Abb. 2 sind die Latenzzeiten zwischen Client A (Publisher) sowie den weiteren Clients, die sich im Experiment befanden, dargestellt. Insgesamt lag der Mittelwert der Übertragungszeit bei ca. 146 Millisekunden für Client A(Echo), 151 Millisekunden für Client B und 155 Millisekunden für Client C.

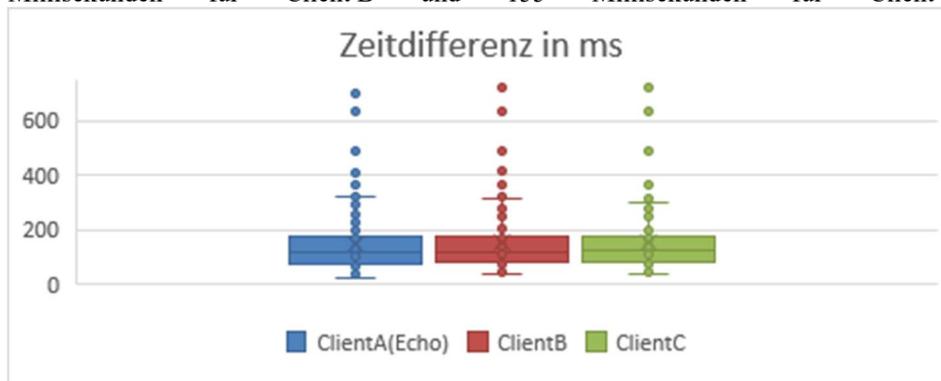


Abb. 2 Zeitdifferenzen zwischen Clients bei Datenaustausch über Publish-Subscribe

Der Zustand, dass zwei Benutzer zeitgleich eine Änderung am Modell erzeugen und sich dadurch gegenseitig blockieren ist in diesem Anwendungsfall eher die Ausnahme, sofern der Mittelwert herangezogen wird. Wird jedoch vom Worst-Case-Szenario mit 723

Millisekunden ausgegangen, so erhöht sich die Chance auf einen Konflikt. Auch mit einer steigenden Anzahl an Nutzern nimmt die Wahrscheinlichkeit zu, dass eine Konfliktsituation eintritt. Um diesen Zustand jedoch abzufangen, wird während der Aktualisierung des Modells die Funktion zum Bearbeiten für einen kurzen Moment gesperrt. Befindet sich der Anwender jedoch gerade in einer länger andauernden Aktion, wie z. B. dem Verbinden einer langen Kante, bei der ein Start und Endpunkt definiert werden muss, so muss die laufende Aktion abgebrochen werden. Dies ist die einzige Möglichkeit die Konsistenz des Modells zu erhalten, da ansonsten die Änderungen, die nach dem Beginn des Vorgangs von anderen Anwendern durchgeführt wurden, verloren gingen. Zusammenfassend eignet sich die verwendete Technologie sowohl theoretisch als auch praktisch zur kollaborativen Informationsmodellierung (RQ3).

6 Ausblick

In diesem Beitrag wurde eine Realisierung zur echtzeit-basierten Modellierung von BPMN-Prozessmodellen mithilfe der Verwendung des MQTT-Protokolls vorgestellt. Dazu wurde eine Architektur konzipiert und implementiert, die auf Basis von Anforderungen verschiedener Projektpartner und Modellierungsexperten erhoben wurden. Es konnte dargelegt werden, dass der kollaborative Ansatz mithilfe des MQTT-Protokolls einsatzfähig ist und das Modellieren mit mehreren Benutzern auf verschiedenen Endgeräten in nahezu Echtzeit genutzt werden kann. Allerdings konnte auch aufgezeigt werden, dass noch Optimierungspotenzial hinsichtlich der Größe der zu übertragenden Daten besteht. Die Implementierung der Übermittlung von Delta-Informationen über die einzelnen Clients wurde bereits getestet, jedoch konnte die Konsistenz der Daten zwischen den Clients nicht sichergestellt werden. Die Deltas wurden hierbei in Form von Befehlen zur Wiederherstellung des Zustandes extrahiert und übermittelt. Diese Implementierung scheint auf Basis der zugrunde liegenden Bibliothek nicht deterministisch, was zu inkonsistenten und fehlerhaften Modellen führte. Durch die Übertragung von Deltas ist es zudem möglich bessere Algorithmen zur Konsistenzbildung zu implementieren. Bei einer Übertragung von Deltas, könnten nur die Modellteile die tatsächlich einer Änderung unterliegen gesperrt werden, was die negative Beeinträchtigung des Anwenders beim Modellieren stark reduzieren dürfte.

Entstanden im Technologietransferprojekts „Kompetenznetzwerk Intelligente Produktionslogistik“ das aus Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE) – Operationelles Programm mit Ziel „Investition in Wachstum und Beschäftigung“ Bayern 2014 – 2020 gefördert wird. Förderkennzeichen: EU-1703-0001

7 Literaturverzeichnis

- [AP01] Appelt, W.; Busbach, U.; Koch, T.: Kollaborationsorientierte asynchrone Werkzeuge. In (Schwabe, G.; Streitz, N.; Unland, R. Hrsg.): CSCW-Kompendium. Springer, 2001; S. 194–203.
- [DI19] Dizdarević, J. et al.: A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. In ACM Computing Surveys, 2019, 2019; S. 1–29.
- [HI18] Hilpoltsteiner, D.; Seel, C.; Dörndorfer, J.: Konzeption und Implementierung eines Softwarewerkzeuges zum Management von BPMN-Prozessvarianten. In (Hofmann, R.; Alm, W. Hrsg.): Wissenstransfer in der Wirtschaftsinformatik. Fachgespräch im Rahmen der MKWI 2018. Hochschule Aschaffenburg, 2018; S. 15–24.
- [HO01] Holmer, T.; Haake, J.; Streitz, N.: Kollaborationsorientierte synchrone Werkzeuge. In (Schwabe, G.; Streitz, N.; Unland, R. Hrsg.): CSCW-Kompendium. Springer, 2001; S. 180–193.
- [LA18] Laaroussi, Z.; Morabito, R.; Taleb, T.: Service Provisioning in Vehicular Networks Through Edge and Cloud: An Empirical Analysis: 2018 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, 2018 - 2018; S. 1–6.
- [LE13] Lee, S. et al.: Correlation analysis of MQTT loss and delay according to QoS level: International Conference on Information Networking (ICOIN), 2013. IEEE, Piscataway, NJ, 2013; S. 714–717.
- [LE14] Leimeister, J. M.: Collaboration Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [SC01] Schwabe, G.; Streitz, N.; Unland, R. Hrsg.: CSCW-Kompendium. Springer, 2001.
- [WA16] Wang, Y.; Leblanc, D.: Integrating SaaS and SaaS with Dew Computing. In (Cai, Z. Hrsg.): Conference Workshop Applications of Science ABDS Applications BDCLOUDAPP in Bioinformatics BDACCB. IEEE, Piscataway, NJ, 2016; S. 590–594.