

Business Process Context for Message Standards

Nenad Ivezic^{1,*}, Miroslav Ljubicic¹, Marija Jankovic¹, Boonserm Kulvatunyou¹,
Scott Nieman², and Garret Minakawa³

¹ National Institute of Standards and Technology, Gaithersburg, MD, USA
{nivezic,miroslav.ljubicic,marija.jankovic,serm}@nist.gov

²Land O'Lakes, Shoreview, MN, USA

stnieman@landolakes.com

³Oracle, Redwood City, CA, USA

garret.minakawa@oracle.com

Abstract. Despite unrelenting increase in complexity of message standards for enterprise systems integrations, there are no effective means to address this complexity issue in practice. We describe an effort to address the issue by advancing message standards development and use methods. The new effort relies on business process model life-cycle management, which is essential for context definition of message standards usage. Context is essential as it describes the intent for the message standards usage for a specific systems integration case. We report results of a preliminary assessment of the approach for an industry use case.

Keywords: systems integration, message standards, life-cycle management, business process model, context

1 Introduction

Efficient, practical, systems integration continues to be a great challenge for enterprises of all sizes, in great part because of the increasing complexities of message standards for the integration. The Open Applications Group, Inc. (OAGi) is one of the original consortia that standardize message-exchange standards [1]. Without a means to manage a shareable context specification, OAGi members have seen the message standards becoming complex and their management unwieldy.

Business processes are prime candidate to supply context specification for the messages involved in information exchanges. This has been recognized for many decades, starting with the activity modeling language IDEF0 where inputs and outputs capture the business data to be exchanged between activities [2]. The OAGi consortium has taken first steps to offer BPMN-based standards for business processes to provide precise context for message exchanges [3].

Recently, BPMN 2.0, with its BPMN.xsd representation and runtime execution capability, has accelerated the design, development, and implementation of message and process standards [3]. However, problems still exist in 1) consistency and interoperability between business process modeling tools, 2) adequacy of the content captured in the process model, and 3) process cataloging for reuse and adaptability.

In this paper, we describe a new approach for business-process model life-cycle management (BPM LCM) to tackle these problems. The outcome of this approach will be a useful, shared, business process-based context definition for message standards development and use. Such a definition will allow enterprises to accelerate systems integration efforts.

2 What are Message Standards, anyway?

Message standards are data standards that define both the structure and the semantics of the message. Such standards govern information exchange among applications, services, and other actors. By doing so, message standards facilitate the systems integration. Effective information exchange, however, is hindered by the growth of individual message standards, in both size and complexity.

Presently, message standards, such as Open Applications Group Integration Specification (OAGIS), are growing complex for multiple reasons [1]. First, systems today are deployed on a variety of computing platforms, each using a different computer language. Second, these standards support a wide range of enterprise business processes and sectors including aerospace, automotive, chemical, and electronics that are subject to various quality criteria, regulations, and other factors. Third, new industrial integration use cases continue to expand definitions of message standards. Thus, critics often claim these standards are ‘bloated.’

To use such “bloated” message standards today, most companies must perform manual, time-intensive adaptations of message standards to address systems integration requirements. These adaptations result in subsets or profiles of the standard for actual use in the context of each such systems integration case. To create this profile, implementers and business analysts must first determine which elements of the message standard are applicable to their integration use case. Then, they must manage and relate that part to edge application APIs. This results in time-intensive, error-prone, manual interpretations of standards. In addition, these efforts must be repeated every time a new computing platform is introduced.

As a strategy to address these problems, industry has shown interest in using life-cycle management (LCM) methods to advance message standards. LCM methods are processes for the development, use, adaptation, operation, and maintenance of a standard. These methods are expected to cover all phases of the standard from the requirements gathering to the end of life. Examples of OAGIS message standards, and their associated LCM methods, can be found in [1].

The current message-standards LCM methods, however, lack the ability to manage the growing complexity of message standards. The reasons are 1) they treat integration use cases independently and 2) they provide only for additions of required data elements to standard message definitions. In other words, these methods do not capture the underlying business processes that drive the integration in the first place; nor, do they identify the data elements that are shared as part of that integration. Consequently, there is no consistent, shareable definition of the intended integration uses of any message standard – that is, definition of context.

Such a definition could inform and specify those intended uses including the necessary adaptations and refinements of message standard across different integration situations. To do so, the definition must provide usage information that includes (1) customized or profiled message standard, (2) intent for the customized message standard, and (3) accumulation of data at each step of the business process used to customize the message standard.

Our work addresses the absence of the means to provide and manage usage information of messaging standards. To understand our approach, consider the current (As-Is) state of message-standards life-cycle management (MS LCM) in Fig. 1 and the envisioned (To-Be) state of MS LCM in Fig. 2. There are three areas where we seek advancements in MS LCM (as indicated in the figures), which currently have very limited tool support:

1. *Integration Requirements Definition* where today Business Process Analysts inefficiently and in an ad-hock manner specify the requirements in natural language (typically using non-standard business process models created in Visio or Powerpoint) and based on a target business process to be supported.
2. *Message Standards Adaptation* where today Software Developers inefficiently work with the integration requirements provided in natural language form to identify and adapt (e.g., prune and extend) standard messages for specific application schemes present in the integration case. The developers also review application APIs to identify required fields that may have been missed, and may refactor message standard profile to include these fields.
3. *Profile Message Generation* where presently Software Developers engage in cost-inefficient transformations of one implementation language-specific profile message definition into another.

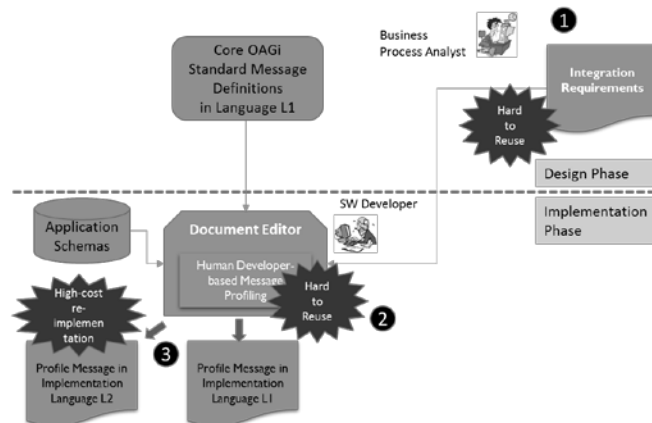


Fig. 1. As-is state of Message Standards Life-Cycle Management

Our focus in this paper is on the first area where a new business process model life-cycle management (BPM LCM) approach is introduced for message standards

context definition management. The approach allows greater reuse and automation in the Integration Requirements Definition area, and in the other two areas of interest.

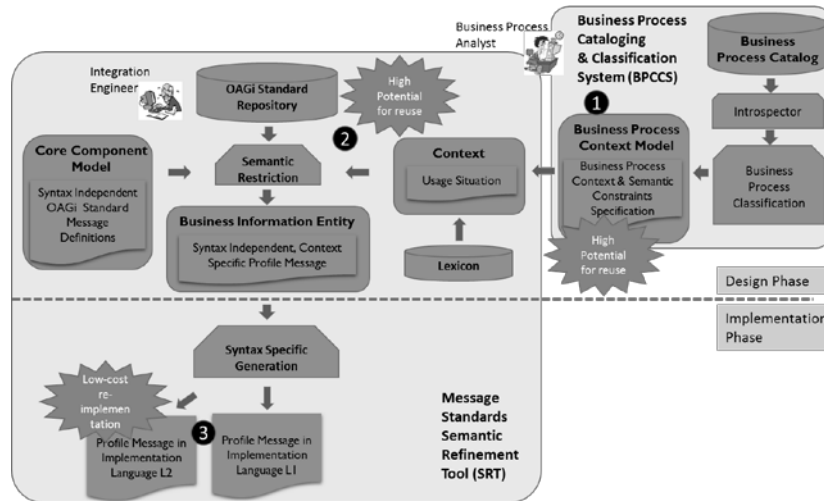


Fig. 2. To-be state of Message Standards Life-Cycle Management

3 Tools to migrate from the As-Is to the To-Be state

Our research, which was done with OAGi, has led to the development of two tools: 1) Message Standards Semantic Refinement Tool (MSSRT), which improves the message standards LCM process; and 2) Business Process Cataloging and Classification System BPCCS) to link process models to a usage meta-model, enabling Business Process Models Life-Cycle Management (BPM LCM).

3.1 Overall Architecture

Fig. 2 shows responsibilities of the two tools – BPCCS and MSSRT – in the To-Be state of MS LCM. BPCCS is being developed as part of a BPM LCM method to manage the life-cycle of both reference and context-specific OAGIS business-process models. A core focus of the BPCCS tool is to provide shared definitions of required concepts and terms for business processes that span across an enterprise and its multi-tier supplier network. The BPCCS tool performs three major functions. The first is to create and manage the Context Model. The second is to provide to the Business Process Analyst a user interface by which the context model is specified along with additional semantic constraints on the process model. The third is to communicate this usage information to the MSSRT tool.

The MSSRT tool will apply this usage information to the syntax-independent, standard, message definition to create a context-specific profile message. MSSRT also transforms a syntax-independent form of the profile message into an implementation-specific profile message. In providing these capabilities, the

MSSRT tool will enable business-process-model discovery and reuse. Towards that goal, we are planning to use introspection functionality (allowing discovery of the model properties at runtime) to harvest business- process-model information for context definition.

MSSRT is intended to aid the systems integrators and users in generating and cataloging the message-standard profiling information using a new, CCS-compliant OAGIS meta-model [4]. This tool is used as a web-based, application-software environment for the life-cycle management of message standards. In parallel, the tool will be utilized to experiment with new methods to create, maintain, and use message standards. The software tool includes collaborative, multi-tenant methods and meta-models for life-cycle management of message standards. Those methods and models can be shared with the community and can facilitate the creations of extensions made by the community to be submitted into the standard. It provides core functionalities that can be extended and commercialized by industry. Among the functionalities being explored are those to allow us to deal with natural language issues, such as term matching and synonym handling (indicated in Fig. 2 as Lexicon).

The following sections focus on our systems engineering approach in developing BPCCS, the current state of the development, and initial tool assessment results.

3.2 Business Process Cataloging and Classification System (BPCCS)

Requirements Gathering. We collected use cases to gather requirements and identify activities to support using BPM LCM functions of the BPCCS. For example, one use case is an end-to-end, product-procurement scenario with the goal to support BPM discovery and reuse. The case starts with the customer issuing a purchase order and ends with the customer receiving the goods, the shipment notification, and, the as-built inspection information. The components of BPMs are classified using a reference classification framework APQC PCF [5]. Multiple context-classification schemes are managed in the BPCCS to help catalog these BPMs, including ISO 10314 classification [6], ebXML-adopted Porter classification [7], and OAGIS functional classification [5]. When outsourcing the enterprise activities associated with parts of a BPM, the BPCCS provides BP-based context information to help discover available manufacturing services. Once services are discovered, the original BPM may be modified. The original BPM is kept for traceability purposes, where the APQC PCF is employed to allow cross-industry reference.

Requirements Analysis. Fig. 3 illustrates the two main functional parts of our approach to support the required BPM LCM functions: Classification Schemes (on the right) and Catalog (on the left).

The Classification Schemes allow the BPCCS users to classify their process models (and their parts) using multiple contextual dimensions, thereby defining a context for the intended use of the model. Several contextual dimensions have been proposed previously for inclusion into systems like BPCCS. Those dimensions include industry, product, business process, role, and function, among others [8]. The

assumption is that it should be possible to find a process model with the needed (or similar) semantics by providing the context in which the model will be used.

The BPCCS Catalog stores and inspects business process models to extract relevant metadata. Both reference and specific models are stored and described by the context in which they are derived using Classification Schemes' context dimensions. Derived process models are regarded as variants of the reference process model. Variants are needed because the context of a specific model can differ from the context of the original reference model, as illustrated in Fig. 3. Reference-model elements are used as shared terminology to which elements of specific process models are mapped and thus semantically aligned.

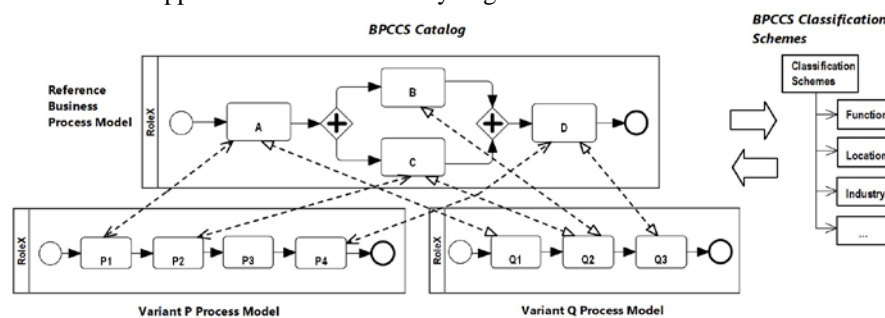


Fig. 3. Overview of Approach

Conceptual Design. Following the requirements analysis, we identified the needed BPCCS capabilities, chief among which was the ability to capture precisely context and its semantics for BPM LCM. We analyzed research results in four related areas regarding these capabilities [9]: Modeling BP Variability; Modeling BP Context; BP Catalog Approaches; and Industry Approaches. Our analysis showed that effective support for the needed capabilities still does not exist. (Please see [9] for details of the analysis.) To address this situation, we proposed a BPCCS metamodel that builds on the ebRIM specification [10].

BPCCS allows its users to classify their process models, and parts of the models, using multiple contextual dimensions. The dimensions can be grouped by their purposes and organized into 'aspects', represented by the ContextAspect element. Currently, we are leveraging Zachman's framework [11] and its 5WH maxims¹ to organize context dimensions into context aspects. For example, geographical-location and organization-unit context dimensions is related to the 'Where maxim'; while, the industry context dimension is related to the 'What maxim.' The previous Introspection functionality can help pre-populate the results of the 5WH maxims.

Fig. 4 illustrates the metamodel concepts on a generic example. Here, the context of a business process is described by associating the process (the far right of the Fig. 4) with various (multiple) classification nodes (OY, US, 44, 10279, etc.), belonging to different classification schemes (ISO 3166, NAICS, etc.) used to describe different

¹ Who, What, Where, When, Why, and How

dimensions of the context (Role, Geo Location, etc.) through giving the answers to different context aspects (Why?, When?, etc.).

To define the range of values for a context dimension, different taxonomies or controlled vocabularies can be used. Beside classification schemes, stored catalog objects of a particular type can also be used as values of context dimensions.

Finally, the context of a particular catalog object is defined as the set of all the associations the object has. Included among the associations are classification nodes of appropriate schemes and other catalog objects of appropriate types that can be used for particular context dimensions.

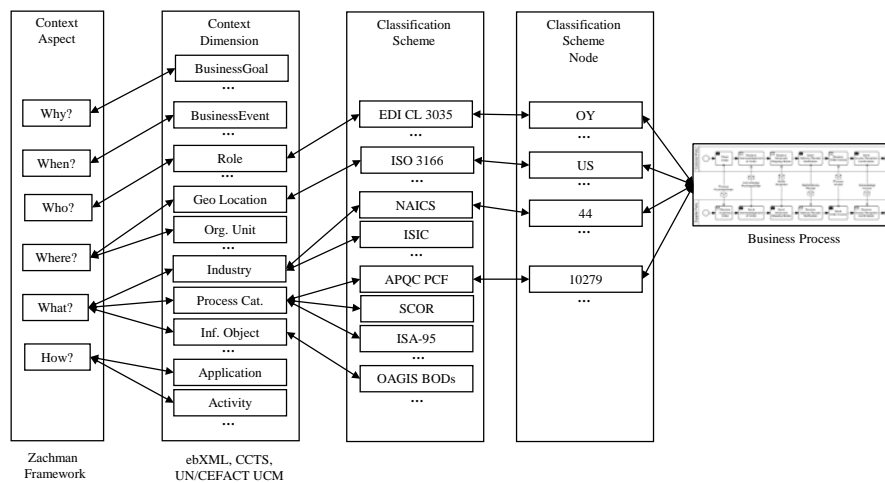


Fig. 4. An Illustration of Context for a Business Process Model

Verification. We have developed proof-of-concept prototype based on the BPCS metamodel. Also, we collected various business processes together with their models and context definitions, which we used to populate the prototype implementation. Most of these processes were obtained from members of the OAGi consortium where integration use cases are collected for enhanced reuse of information and communication artifacts. In this example, the goal is to describe business processes and their components for their subsequent retrieval and reuse.

We focus on *Retrieve Electronic Control Unit (ECU) Information* sub-process which encapsulates activities for exchanging information about the product and its parts (in this case, ECU) between various systems [12]. The goal is to design this sub-process by reusing an existing process model.

First, we needed to search the collection of populated, business-process models for the ones applicable to the context of the *Retrieve ECU Information* sub-process. To use this context, values for all context aspects should be provided. As suggested before, this can be done by answering Zachman's interrogatives. Context for *Retrieve ECU Information* sub-process, defined in this way, is given in Fig. 5.

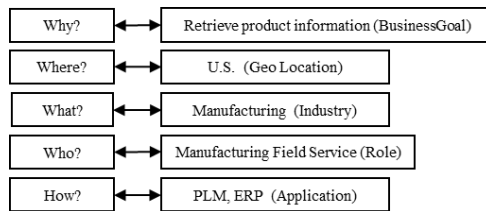


Fig. 5. Example Context for the *Retrieve ECU Information* Sub-process

Once the context is identified, we can browse BPCCS to find applicable business processes. For this purpose, a complete or partial context from Fig. 5 can be used. For example, we can start by searching processes that achieve the goal (Retrieve product information). This search yielded three business processes models – three different BP variants of the Exchange Product Information goal.

Although the outcome of all variants is the same – exchanging full product information – each variant uses a different sequence of activities and different message exchanges. Messages are defined using OAGIS Business Object Documents (BODs) specification [1]. BODs are defined using a verb-noun structure, where verb defines the desired action that should be applied to the exchanged business information, which is represented by a noun. For example, ProcessBOM BOD (verb: Process; noun: BOM – Bill of Materials) specifies that the receiving system shall execute a certain process on the BOM contained in the BOD.

Variant 1 uses BODs based on a single noun – ExhaustiveBOM noun, which defines BOM for full product information, including structure and child item details. Variant 2, instead of exchanging full product information with a single message, uses a BOD based on StructureBOM noun to exchange structure of BOM first. Variant 3 is same as Variant 2, but with one subtle difference. After receiving the structure of the product, the user decides which detailed parts information should be retrieved. This was not possible in Variant 2, where details for all product’s parts were exchanged by default.

These differences between variants are related to their different, although similar, contexts, as shown in Table 1. Columns represent variants, while rows define context dimension/aspect combinations. Table 1 also shows used classifications.

Analyzing contexts of the business process variants provide information for determining which variant should be reused. For example, there is a difference in the Industry context dimension, where Variant 1 is designed for Wireless Telecommunications Carriers, and Variant 2 and Variant 3 are designed for Electronic Computer Manufacturing. This context difference had a crucial impact on variant design and is correlated with the messages (BODs) used in the variant. Namely, telecommunication products and services do not have overly complex structure; and, it is computationally feasible to exchange their full information using a single message. Hence, Variant 1 uses BODs based on ExhaustiveBOM noun. However, this cannot be expected for manufacturing domain as products are typically much more complex. Thus, Variant 2 and Variant 3 separate product

information using BODs based on StructureBOM and ItemMaster nouns to make potentially very large product information exchange feasible.

Table 1. Comparison of three contexts for retrieved business process variants.

	Exch. product info. v1	Exch. product info. v2	Exch. product info. v3
Business Goal (Why?)	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> Retrieve product information, Synchronize product information	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> Retrieve product information, Synchronize product information	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> Retrieve product information
Process Category (What?)	<i>Class.Scheme:</i> APQC Cross-industry PCF (7.0.4) <i>Node/Value:</i> Manage product and service master data	<i>Class.Scheme:</i> APQC Cross-industry PCF (7.0.4) <i>Node/Value:</i> Manage product and service master data	<i>Class.Scheme:</i> APQC Cross-industry PCF (7.0.4) <i>Node/Value:</i> Manage product and service master data
Industry (What?)	<i>Class.Scheme:</i> NAICS 2012 <i>Node/Value:</i> 517210 Wireless Telecommunications Carriers	<i>Class.Scheme:</i> NAICS 2012 <i>Node/Value:</i> 334111 Electronic Computer Manufacturing	<i>Class.Scheme:</i> NAICS 2012 <i>Node/Value:</i> 334111 Electronic Computer Manufacturing
App. (How?)	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> CRM, ERP, API	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> CRM, ERP, API	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> CRM, ERP, API
Role (Who?)	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> Customer Relationship Management, Integration Specialist, Order Management	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> Customer Relationship Management, Integration Specialist, Order Management	<i>Class.Scheme:</i> Custom <i>Node/Value:</i> Customer Relationship Management, Integration Specialist, Order Management
Location (Where?)	<i>Class.Scheme:</i> ISO 3166 <i>Node/Value:</i> US	<i>Class.Scheme:</i> ISO 3166 <i>Node/Value:</i> US	<i>Class.Scheme:</i> ISO 3166 <i>Node/Value:</i> US

4 Discussion and Next Steps

Our assessment showed that context specification enabled by the BPCCS metamodel supports desired behaviors. First, it was possible to specify the goal context intuitively using Zachman’s 5WH interrogatives. Second, it was possible to search for business processes applicable to a given context, as described by context dimensions. Third, manual comparison of business processes by their context was supported. Fourth, process-model designs could be analyzed in correlation with their contexts. Fifth, it was possible to find, and manually reuse, the most appropriate business process, together with message profiles (i.e., BOD subsets).

Other types of assessments are planned including support for greater automation in business process model reuse and refinement, which requires further development of context management apparatus. This is planned to be done using semantic technologies, developed in parallel with a needed ontological basis, for explicit and shared conceptualization of the context elements. For the automation to be realized, BPMN [3] conformance testing of process modeling tool is anticipated, which is necessary for the business process models introspection functionality.

Finally, we are planning for user and organizational adoption of MSSRT and BPCCS tools, which is a particularly challenging in the light of likely disruptions to the current practices of message standards development and use.

5 Conclusion

The paper presents a new approach to manage business process-based context definition that describes the intent for usage of message standards. Central to the approach is business process model life-cycle management capability. A preliminary assessment shows that the approach provides desired support to the end user in search, comparison, analysis, and reuse of business process models. Next steps include integration of the approach within the overall message standards life-cycle management support, further validation of the approach, and work on its adoption within standards development organizations.

Disclaimer

Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.

References

1. Open Applications Group Integration Specification (OAGIS) . <http://www.oagi.org>. Accessed 12 Jul 2017.
2. NIST (1993) Draft Federal Information Processing Standards, Standard for Integration Definition for Function Modeling (IDEF0).
3. BPMN (2011) Business Process Model and Notation (BPMN), Version 2.0.
4. ISO 15000-5:2014 Electronic Business Extensible Markup Language (ebXML) - Part 5: Core Components Specification (CCS). <https://www.iso.org/standard/61433.html>. Accessed 12 Jul 2017
5. APQC Process Classification Framework. <https://www.apqc.org/>. Accessed 12 Jul 2017
6. ISO/TR 10314-1:1990 Industrial automation - Shop floor production - Part 1: Reference model for standardization and a methodology for identification of requirements. <https://www.iso.org/standard/18360.html>. Accessed 12 Jul 2017
7. ebXML (2001) ebXML Catalog of Common Business Processes v1. 0. UN/CEFACT and OASIS
8. ebXML (2001) The role of context in the re-usability of Core Components and Business Processes. UN/CEFACT and OASIS
9. Ljubic M, Ivezic N, Kulvatunyou B, et al. (2017) Business Process Model Life-Cycle Management in Cloud Manufacturing. Proceedings of the ASME 2017 International Manufacturing Science and Engineering Conference
10. ebXML RIM (2012) OASIS ebXML RegRep Version 4.0 Part 1: Registry Information Model (eBRIM). OASIS
11. Zachman J (2002) The Zachman Framework for Enterprise Architecture.
12. Kulvatunyou, B., Ivezic, N., and Srinivasan V. "On architecting and composing engineering information services to enable smart manufacturing." *Journal of computing and information science in engineering* 16.3 (2016).