A Business-Context-Based Approach for Message Standards Use - A Validation Study

Elena Jelisic[1(✉)], Nenad Ivezic[2], Boonserm Kulvatunyou[2], Nenad Anicic[1], and Zoran Marjanovic[1]

[1] Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia
{elena.jelisic,nenad.anicic,zoran.marjanovic}@fon.bg.ac.rs
[2] National Institute of Standards and Technology, Gaithersburg, MD, USA
{nivezic,serm}@nist.gov

Abstract. While necessary for a successful integration of enterprise services and business-to-business (B2B) applications, message standards can be difficult to use. This paper proposes an innovative, business-context approach and a software tool that we believe can overcome those difficulties. To accomplish both, the paper shows how the business-context approach and the new tool address difficult issues common to traditional approaches of using message standards. The paper also identifies research questions that need to be addressed to increase scalability of the new approach and tool.

Keywords: Enterprise applications · B2B · Services · Integration · Business context · Software

1 Introduction

Message standards are key to the integration of many applications and services, especially in this age of numerous, emerging, ecosystems of services [1,2]. Yet, message standards usage is complex because the standards are very large superset specifications. Moreover, those specifications were developed to address integration requirements from several business processes and multiple industry domains. Consequently, to use the standards, it is necessary to (1) know the implementation-specific language for the standard and (2) have capability to create a ***profile*** (relevant subset of) the standard for the specific integration situation. Recently, a new software tool [3], based on a UN/CEFACT international standard, has been developed to address both necessities. The tool creates an implementation-neutral representation based on the business context in which the messages are used. The neutral specification facilitates integration; the business context facilitates profiling. In other industries where message standards are used, however, there is currently lack of knowledge of the tool, profiling method, and the impact on different integration scenarios.

T. Welzer et al. (Eds.): ADBIS 2019, CCIS 1064, pp. 337–349, 2019.
https://doi.org/10.1007/978-3-030-30278-8_35

This paper documents an initial validation of the tool and method based on a small but realistic integration use case from the Electric Power industry. The validation is based on the ISO 15000-5 (Core Component Technical Specification) [4] and the Score tool [3]. The validation compares the traditional method of using and integrating message standards with our proposed method. The comparisons focus on the functions associated with profiling including (1) removing the need to know implementation language for the integration; (2) enhancing the ability to capture intent for use of the message standard; (3) decreasing the likelihood of generating new, superfluous, and redundant variations of message standards and their components; and (4) increasing the coherence of the standard by reducing the need for ad-hoc re-use of the its components.

The rest of the paper is organized as follows. Section 2 includes background information about concepts used in the paper. Section 3 describes both a traditional and the newly proposed approach and provides a use case-based analysis of the two on the task of message profiling. Section 4 provides an analysis of approaches on the task of message profiling. Section 5 provides discussion of the analysis results and proposes next research steps. Section 6 completes the paper with conclusions.

2 Background

Score [3] is a novel tool supporting message-standards development and use. It was created in cooperation between Open Applications Group Inc. (OAGi) [5] and National Institute of Standards and Technology (NIST) [6]. The tool, which is based on the Core Component Technical Specification (CCTS) [4], has been used in the development of the latest version of the Open Applications Group Integration Specification (OAGIS) [7]. CCTS is an implementation-neutral, standardization approach that offers two types of data modeling components: Core Components (CCs) and Business Information Entities (BIEs). Together, these components capture both the structure and contents of information-exchange models [8]. CCs are context-independent, conceptual, data-model components. BIEs are logical, data-model components that restrict the underlying CCs to specific business context. Business context (BC) is used to capture intent of a created BIE. UN/CEFACT defines a BC by a set of the context values associated with their corresponding context categories [8]. UN/CEFACT proposed eight BC categories. In this paper we will use four of those categories: Business process role, Geo-political, Activity (Business process), and Industry.

OAGIS uses XML schemas to normatively define the structure of its message specifications in what are called Business Object Documents (BODs). The specifications leave the usage intent of these BODs largely open. BODs follow a standard architecture, developed by OAGi, that contains two main areas: an application area and a data area. The application area conveys integration context information, while data area carries the business and engineering contents. The data area is described using a verb and a noun. A verb is used to define the type of operation that should be conducted using the exchanged business content. A noun is defined using a set of components, which can either be a simple field or a compound field that envelops a set of fields [9].

The Score tool implements an innovative approach that brings new efficiencies to the size of OAGIS. This novel approach increases the reusability of OAGIS's library of standard CCs both at the modeling and XML-Schema levels [10]. This is accomplished using the business context that describes the intent of a profiled message standard. Such an enhanced version of OAGIS is referred to as "business-context-aware OAGIS".

3 Traditional vs. New Approach to Enterprise Applications Integration: A Use Case-Based Comparison

Guided by our intent to validate the use of the new approach, we position the validation activities within accepted practices in business-to-business integration (B2B). This section starts with a simple use case utilized for comparison of two approaches. The first one is a traditional approach, where OAGIS is used as a message standard for defining interoperable structures of an interchanged business documents (i.e. messages). The second one is the proposed approach, where business-context-aware OAGIS will be used for the same purpose. Then, we describe steps that are common to both the traditional and the new approach, which occur prior to the profiling step. Next, we discuss the profiling step in the traditional approach and identify issues in this approach. Finally, we provide details of the profiling step in the proposed approach.

3.1 A Use Case: Complaints Processing

An existing enterprise in Serbia, denoted as Company A, designs new functionality to handle complaints. This enterprise operates in the Electric power generation, transmission, and distribution industry. Besides its power-servicing activities, this enterprise also manages public procurements. The subject of a public procurement can be anything, but this paper will focus on integration with enterprises that supply products to the company. This supplier is denoted as Company B. For ease of understanding, we focus only on the Complaint business document that is created because of the processing of complaints upon receiving a shipment. Figure 1 shows a complete, public-procurement process using a System Context Diagram (SCD). The rectangles in the figure are the actors who share information - the edges in the figure - that is needed to create the public-procurement, complaint document. That document is represented by the thick edge, **Complaint**, from the pubic-procurement function and the business partner, **Supplier** actor. The message, the one that contains the compliant documents, will be the focus of our validation.

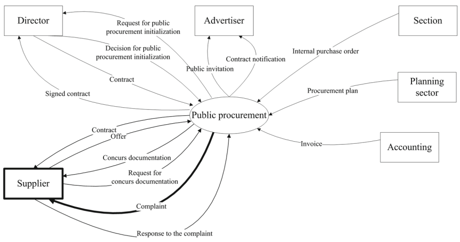

Fig. 1. Public procurement System Context Diagram (SCD).

3.2 Traditional or New Approach: Pre-profiling Steps

Within the validation study, there are seven pre-profiling steps that are common to both the traditional and the new approaches.

1. *Identify business partner(s).*
2. *Identify messages (documents) to exchange with business partner(s).*
3. *Identify the part of a business system that is subject of integration process.*
4. *Identify public processes.*
5. *Agree on business document structure.*
6. *Agree on a message standard usage.*
7. *Create a source serialization format – an XML Schema.*

For realization of the first five steps, the FonLabis [11] information-system-design methodology (a best-practice system-identification approach) was used. For identifying business partners and messages, the FonLabis approach was used to create the above SCD. Steps three and four are identified using Data Flow Diagrams (DFDs). In the fifth step, the FonLabis approach was used to access the data models and data dictionaries needed to recognize the structure of the business document that should be interchanged. In the sixth step, review of existing message (or document) standards against both business and technical requirements was conducted. In the seventh step, required database tables, and specific columns inside those tables, are chosen and transformed into XML Schema. Following are some details of the use case-specific realization.

Step 1. Company A manages procurements and collaborates with partners. Some of these partners are internal while others, such as suppliers, are external. Since B2B is the focus of this paper, only external partners will be considered. The supplier in our example is Company B from USA.

Step 2. Since the scope of business partners is narrowed, the only messages considered are those exchanged with Company B. This paper focuses on interchange of the Complaint business document.

Step 3. We have used DFDs to decompose the public procurement process into three sub-processes: Contracting, Realization, and Processing of Complaints.

Since we focused on suppliers and the Complaint business document, the only process of interest is Processing of Complaints.

Step 4. Processing of Complaints is further decomposed into three sub-processes: Receipt Processing, Complaint Sending, and Response Processing. Only the former two processes are communicating with a supplier by exchanging messages. These processes are called public [11]. At this point we define integration domain as *Sending a complaint to Supplier.*

Step 5. Data Dictionary is used to describe the business document structure. The complaint business document structure is presented in Fig. 2. Each field is described by its domain and constraints.

```
Complaint:<Complaint_Number, Customer, Supplier, Receipt_Number, Contract_Number,
Invoice_Number, Date, Complaint_Explanation, Employee>

Customer: <Customer_Name, Customer_Tax_ID, Customer_Address>
Supplier:<Supplier_Name,  Supplier_Tax_ID, Supplier_Address>
Customer_Address: <Country, ZIP_Code, Town, Street_Name, Number, [Appartment_Number]>
SupplierAddress: < Number, Street_Name, [Appartment_Number], Town, State, ZIP_Code>
Employee :<Employee_Name, Employee_Surname>
```

Fig. 2. Complaint message data dictionary.

Step 6. OAGIS is a widely used message standard for B2B integration and it is adopted for creating an interoperable Complaint business message structure.

Step 7. By analyzing Company A's database design, we have identified tables and columns inside those tables that are necessary in this use case realization. Thereafter, an SQL view is created over all identified tables and columns. Finally, XML Schema is generated using a user-defined function that takes an existing view and creates a corresponding XML Schema document.

3.3 Traditional Approach: Profiling Step

These activities take place within the Profiling step of the traditional approach:

1. *Understand the selected message standard use.*
2. *Define the message standard intent.*
3. *Select an adequate message standard schema.*
4. *Profile the message standard schema.*

The first activity needs to answer questions such as (a) What is the structure of a business message? (b) How the message can be customized? (c) What common XML Schemas are used? (d) What integration scenarios exist? (e) What messages are used in the scenarios? Next activity chooses an integration scenario in which the message will be used. Based on the scenario, the user selects an appropriate, standard, message schema. Finally, that selected message schema needs to be profiled meaning that for each element in the source XML Schema, a counterpart must be selected in the standard message schema. Following are details of the use case-specific realization along with issues discovered in the traditional approach.

Step 1. As discussed in Sect. 2, OAGIS BODs cover many functional areas of enterprise. Consequently, OAGIS messages have a very modular and elaborate structure that requires reuse of existing components or fields, as shown in Fig. 3. The graph shows a part of the ***ItemNonconformance*** noun. Nodes colored black are defined in the ***ItemNonconformance*** schema. White and light gray nodes respectively reflect reuses of ***Components*** and ***Fields*** defined in the corresponding modular schemas. The dark gray node ***IdentificationType*** is a reusable node defined in the ***Meta*** schema[1].

Issue 1. In Fig. 3 there is a referenced element ***Party***, defined in ***Components*** schema. This element is described through a set of complex types in an inheritance hierarchy starting from the ***PartyIdentification***, ***PartyBaseType***, and then ***PartyType***. This pattern also repeats in its sub-elements such as ***Location***. As we can see, a long list of complex types and their extensions have to be navigated to get to the whole structure of only one element, ***Party***. Even in this very simple example, it is evident that the structure of messages (1) can be far from simple AND (2) hard to trace through either automatically or manually.

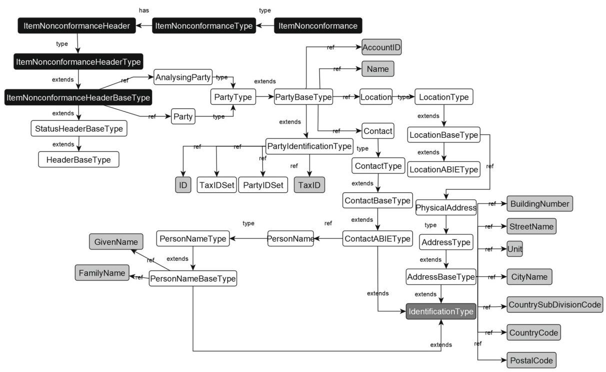

Fig. 3. An illustration of the ItemNonconformance noun and the Party component structures.

Step 2. OAGIS offers several, generic, integration scenarios as starting points for designing use case-specific integration solutions [7]. Each such scenario proposes a set of generic messages to as the basis for integration. By analyzing the OAGIS-provided list of proposed integration scenarios, we have concluded that, for our use case, the most fitting one is Scenario #64 Item Nonconformance.

[1] The XML-Schema structure shown in Fig. 3 is for this paper only. It is but one possible structure for ***ItemNonconformance***. It does not conform to any known standard notation.

This scenario hints at several possible integration flows for reporting item or product non-conformance [7]. These flows represent the two possible reasons for a complaint. Two nouns are selected from the scenario including ***ItemNonconformance*** and ***EngineeringChangeOrder***. These nouns are meant to be used with three, selected, use case verbs: ***Process***, ***Notify*** and ***Show***.

Issue 2. Scenarios that exist in the OAGIS standard, as stated, are used only as a guide providing starting points for defining one's own specific integration flow. This means that messages identified in the scenario can be used in several, different, integration use cases. Hence, there is a problem of finding and selecting the use case, and its existing messages, that is most relevant to our use case.

Step 3. All identified messages from the chosen scenario (Scenario #64) were inspected and after analyzing their structures, we have selected ***NotifyItemNonconformance*** BOD for our example.

Issue 3. This type of analysis needs to be done for every new integration use case. Because of the ad-hoc nature of the step, existing message profiles might not be found; even if there were a similar use case and intent from such an existing profile.

Step 4. For illustration, we focus only on profiling the ***Party*** component from ***ItemNonconformance*** schema. This element is referenced, and its first-level type extends ***PartyBaseType***. For our mapping we needed only ***TaxID***, ***Name***, ***Location*** and ***Contact*** elements. For ***Contact*** we needed ***GivenName*** and ***FamilyName*** elements, and for ***Location*** all elements that are presented in Fig. 3 are needed.

Issue 4. Figure 3 shows that complex types can be directly restricted or extended (through XML Schema extension/restriction); but, such restrictions or extensions do not apply to complex types of the referenced elements. These types must be restricted (or extended) through new elements with new complex types. So, even though all the needed elements already exist in the original OAGIS schema, we still must introduce a new element ***Party*** with a newly defined complex type. Similarly, a new element ***Contact*** with new complex type would also need to be created to restrict ***PersonNameType*** to ***FamilyName*** and ***GivenName*** elements. The same procedure needs to be applied for all reused elements and underlying types that need restrictions or extensions. Consequently, the connections between the newly added elements and their original versions, which are used to give them their intended semantics, are lost.

3.4 New Approach: Profiling Step

Within the *Profiling* step of the new approach, the following activities take place:

1. *Define Business Context.*
2. *Profile message standard component or noun for the defined business context.*
3. *Export profiled message or noun using XML expression.*
4. *Reference profiled components from the profiled BOD.*

The first activity needs to (a) define business categories, (b) define lists of values for each category, and (c) define requested BCs by choosing values for BC categories. In the second activity, one needs to create new profiles by using the existing components and messages. In addition, we reference the BC in which the components and messages are intended to be used. Next, all these profiled objects are exported using XML expression. When we finally get to XML Schemas, all, and only, the needed profiled components are referenced from the target BOD profile in the final step. Following are details of the use case-specific realization.

Step 1. In this use case, four relevant context categories have been identified, including Business Process Role, Geo-political, Activity, and Industry. Relevant roles in the use case that become context values are Procurement director from Company A, and Sales manager from Company B. Since our enterprises are from Serbia and USA, these two values were defined in the list of values for the Geo-political BC category. Other category values could be countries from all around the world. Using International Standard Industrial Classification of All Economic Activities (ISIC) [12] we have defined the list of values for Industry BC category. Lastly, the list of values for the Activity BC category are obtained from the business processes modeled in the SCD and DFDs. Once all the BC category values are known, the needed BCs are created as shown in Table 1.

Table 1. Business contexts A and B.

BC category	Business contexts	
	BC_A	BC_B
Business process role	Procurement director	Sales manager
Geo-political	Serbia	USA
Activity	Sending a complaint	Receiving a complaint
Industry	Electric power generation, transmission and distribution	Manufacturing

Step 2. A logical data model of BIEs is created using UML class diagram and is presented in Fig. 4. The prefix to a BIE name indicates the BC in which the BIE is intended to be used. All BIEs are obtained by restriction of the underlying CCs. The CC library contains all terms used in the OAGIS messages (whether the type of term is component, field, BOD, or data type). Since the Score tool normalizes the inheritance hierarchy of each CC when a corresponding BIE is created, it was easy to restrict all needed CCs through the corresponding BIEs. There are no new standard components introduced.

Step 3. All created profiles are exported as separate XML Schemas. The Score tool uses a specialization of XML Schema, Naming and Design Rules Technical Specification for transforming BIEs into XML schema elements [10, 13].

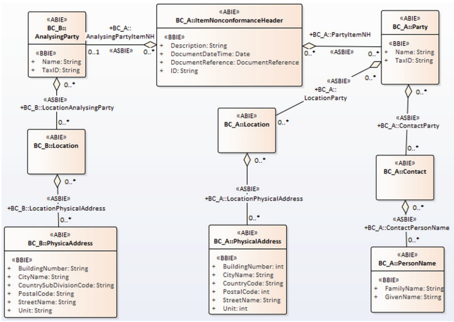

Fig. 4. Logical data model of the ItemNonconformance BIE.

Step 4. In the exported ***ItemNonconformance*** profiled schema, we have imported the other two created schemas for profiled components. Finally, ***ItemNonconformance*** schema elements ***Party*** and ***AnalysingParty*** are changed into referenced elements, thus referencing appropriate elements from the imported schemas.

4 An Analysis of Traditional and New Approaches

This section presents a detailed analysis of the two presented approaches applied to the profiling task. Each use case-specific issue identified in the profiling step of the traditional approach is generalized. In this section we comment on these generalized issues and then show how the proposed approach resolves the issue.

1. Understand the selected, message-standard use.

Generalized Issue: The OAGIS message standard is designed to maximize reusability and extensibility of its components. It is an attractive solution for applications and services integration across functional domains within or between enterprises as well as within and across various industry verticals. This causes the OAGIS structure to be highly compositional. The complex structure demands significant investment to learn to extend and correctly use OAGIS when developing and profiling messages. Also, like virtually all message standards today, OAGIS is an implementation-specific standard and requires extensive XML or JSON knowledge.

New Approach Resolution: The new approach adopts a modular, syntax-independent, CCTS-based representation to manage the OAGIS standard in a model-based manner. Consequently, the OAGIS structure is simplified and made more comprehensive by using the CCTS conceptual and logical components instead of the XML-specific concepts. These components can be presented using UML class diagrams. A UML profile is available to enrich diagrams with CCTS stereotypes.

2. Define the message standard intent.

Generalized Issue: In OAGIS, there is no possibility to record the intent of message standards or their profiles. While OAGIS provides integration scenarios, they are not intended to be a standard specification but only a starting point to describe a similar integration situation for which one designs one's own solution based on the content offered by OAGIS. Therefore, scenarios only give guidelines for OAGIS message selection. This design decision can negatively affect the message standard profile reusability because the integration situation is described imprecisely. This, in turn, leads to issues in finding and selecting existing relevant profiles.

New Approach Resolution: The new approach addresses these issues by enabling a BC definition. BC is an innovative way to define a message standard intent, based on the international CCTS standard. The standard suggests a combination of eight BC categories for representation of a BC. BC can be defined both at the message and the component level. Consequently, BC enables precise description of the integration intent and the component and message profile reusability in related integration scenarios.

3. Select an adequate standard message schema.

Generalized Issue: OAGIS provides a list of integration scenarios that can help users find an appropriate message schema. Using XML tools, every message identified in the scenarios should be analyzed to find the most suitable one. The selected message schema must have a counterpart for each element in the relevant business document. These steps are repeated in every integration use case. This means that redundant and superfluous standard messages are likely to be generated.

New Approach Resolution: The new approach bypasses this issue by using the CCTS representation of OAGIS standard. An appropriate message is chosen by analyzing conceptual, data-model components. Needed components from the conceptual model will be used for creating a logical data model for a specified BC. When the corresponding logical data model is created, it can be transformed into XML Schema or JSON Schema. This logical model can be reused with similar integration scenarios.

4. Profile the standard message schema.

Generalized Issue: Because of the complex structure of the OAGIS canonical message standards in XML Schema, profiling them in that syntax is a demanding undertaking. OAGIS messages are created using inheritance and reference methods. Inherited types can either be restricted or extended, but referenced elements need to be treated separately. The user may use one of the following methods to customize referenced elements (a) Introduce new schema elements with new defined types, or (b) Reference needed elements directly and not through components that contain them. These two resolutions lead to bloated message standards that contain new, superfluous and redundant components. Also, those new standards can cause loss of semantics by ad-hoc combination of existing components.

New Approach Resolution: The user is not working at the XML Schema level; instead he is working at the syntax-independent, conceptual, and logical levels (see Fig. 4). Since a BIE normalizes the inheritance hierarchy of each CC, it is as if each CC exists independently. Moreover, it is easy to restrict all needed CCs through corresponding BIEs, without introducing new ones. The BIEs' names are prefixed with the BC in which they are intended to be used. This increases reusability of existing components. When a logical model is created, it can be transformed into an implementation-specific model. At this time, the tool supports XML Schema and JSON schema serializations. In other words, all the XML Schema-specific requirements, where users had to introduce new elements and types in order to customize original OAGIS Schema, are avoided. This is accomplished by building an implementation-neutral, context-aware model of OAGIS.

5 Discussion and Next Steps

5.1 Profiling Task - Discussion of Analysis Results

We have seen how the traditional approach to message standards use has several issues. Message standards are implementation-specific and require knowledge of an implementation-specific language. In addition, their structure can be very complex, causing the task of message standards profiling to be an exceptionally demanding undertaking. Above all, the most significant shortcoming is that message standards alone do not provide a mechanism to track message standards intent; and there is no existing tool to do so either. As a result, message standards become bloated, grow very large, and become difficult to maintain and use. The newly proposed approach, which is implemented in the Score tool, resolves these issues by providing a simplified and implementation-neutral model for the OAGIS message structure. This model is realized using the CCTS components to build both conceptual and logical data models. Such a representation of the OAGIS messages removes all XML-specific issues and makes message standards more accessible to the integration developer, architect, or the enterprise end user. The new approach provides benefits over a pure, type-based approach. Most significantly, it increases reusability of the Core Components standard's library both at the modeling and schema levels [10].

However, we have also identified research questions that come with the newly proposed approach. These questions need to be addressed for the scalable use of the new tool. Although BC is a valuable concept, it brings new concerns. The lists of categories that describe a BC must be chosen carefully since they should enable the unique identification of the BC. Special attention needs to be paid to given values for each category. Combinations of these values should enable reuse of messages and their components, but only in the right situations. Using many abstract BCs could result in higher chances of inappropriate BIE reuses. Creating rigorous a BC requires in depth analysis of enterprise operating environment and could be time consuming. In addition, BC, at least in parts, should not be defined manually or ad-hoc, but should be defined automatically in some way.

The mapping problem in the traditional approach, however, exists in the proposed approach as well. The ambiguous situations where some elements from the actual business document can be mapped to multiple OAGIS Schema elements still remain and need to be addressed through future research.

5.2 Next Steps

Considering the issues identified in the proposed approach, future research may examine data mining techniques to accomplish BC definition. Namely, such techniques would help identify combinations of context categories expected to adequately define a specific BC. For example, one could envision probabilistic models that take context categories and check the possibility that a given specific BC is plausible. This could allow a user to classify message instances automatically (i.e., assign BC and serve as an outlier detection). For example, one could be alerted if (1) the probability of occurrence of a BC is very low for the given context categories, or (2) that there are additional context categories which are commonly used but were not specified.

There have been some attempts to automate BC definition using business process models. Notably, the collaboration between NIST and OAGi, has led to the development of the Business Process Cataloging and Classification System (BPCCS). The BPCCS tool was developed to create and manage Context Model and to provide a user interface to the Business Process Analyst. The resulting context model is specified along with additional semantic constraints on the process model [14].

The mapping problem could be attacked by introducing the CCTS in the database-design process. This would be achievable only for new, information-system designs where conceptual data models would be created using Core Components. Also, a set of logical data models for different BCs would be provided. This would enable automatic translation of the underlying data structures into OAGIS messages without any risk that incorrect mappings could occur.

6 Conclusion

This paper validates a newly proposed context-based approach for message standards use. A simple, yet realistic integration use case is used as a foundation of the validation. In the traditional approach, the OAGIS message standard is used for defining the structure of exchanged business documents. A number of issues arise in the traditional approach. The paper shows that the new approach can address all the issues by using the novel Score tool to bring context-awareness to the OAGIS standard. Also, although BC is a valuable concept, the paper points at new research challenges that come with the proposed approach. Definition and management of BCs need to be done carefully, not manually or ad-hoc, hence the processes should be automated in some way. The proposed approach still needs to address the mapping problems. The important conclusion is that existence of standards is simply not enough to ensure a cost-efficient, successful application integration. The new approach shows to be a promising avenue to meet these goals.

Disclaimer

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

References

1. Barros, A.P., Dumas, M.: The rise of web service ecosystems. IT Prof. **5**(8), 170–177 (2006). https://doi.org/10.1109/MITP.2006.123
2. Cardoso, J., Voigt, K., Winkler, M.: Service engineering for the internet of services. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2008. LNBIP, vol. 19, pp. 15–27. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00670-8_2
3. Context-Driven Message Profiling. https://oagi.org/. Accessed 10 June 2019
4. UN/CEFACT Core Components Technical Specification CCTS, version 3.0. https://www.unece.org. Accessed 10 May 2019
5. The Open Applications Group Inc. https://oagi.org
6. The National Institute of Standards and Technology (NIST). https://www.nist.gov
7. OAGIS 10.5 Enterprise Edition documentation. https://oagi.org/
8. Novakovic, D.: Business Context Aware Core Components Modeling. Publikationsdatenbank der Technischen Universität Wien (2014). https://publik.tuwien.ac.at/
9. Kulvatunyou, B., Ivezic, N., Srinivasan, V.: On architecting and composing engineering information services to enable smart manufacturing. J. Comput. Inf. Sci. Eng. **16**(3), 15–27 (2016)
10. UN/CEFACT XML Naming and Design Rules Technical Specification. https://www.unece.org/. Accessed 12 May 2019
11. Jankovic, M.: Specifikacija aspekata interoperabilnosti u metodološkim pristupima razvoju IS. Univerzitet u Beogradu (2016). http://nardus.mpn.gov.rs
12. International Standard Industrial Classification of All Economic Activities (ISIC), Rev. 4. https://oagi.org/, https://doi.org/10.18356/8722852c-en
13. OAGi Specification for Serializing BIE to XML Schema. https://oagi.org
14. Ivezic, N., Ljubicic, M., Jankovic, M., Kulvatunyou, B., Nieman, S., Minakawa, G.: Business process context for message standards. In: CEUR Workshop Proceedings, pp. 100–111 (2017)