A Novel Data Standards Platform using the ISO Core Components Technical Specification

Conference Paper · August 2021

6 authors, including:

Boonserm Kulvatunyou
National Institute of Standards and Technology
102 PUBLICATIONS **946** CITATIONS
SEE PROFILE

Elena Jelisic
University of Belgrade
9 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Hakju Oh
University of Maryland, Baltimore County
4 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Simon Frechette
National Institute of Standards and Technology
36 PUBLICATIONS **684** CITATIONS
SEE PROFILE

Proceedings of the ASME 2021
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE 2021
August 17-20, 2021, Virtual, Online

DETC2021-68067

A Novel Data Standards Platform using the ISO Core Components Technical Specification

Nenad Ivezic[1], Boonserm Kulvatunyou[1], Elena Jelisic[2,4], Hakju Oh[3,4], Simon Frechette[1], Vijay Srinivasan[1]

[1]Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA
[2]Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia
[3]Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD
[4]Associate, Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA
Email: {nenad.ivezic, boonserm.kulvatunyou, elena.jelisic, hakju.oh, simon.frechette, vijay.srinivasan}@nist.gov

Abstract

It is generally observed in inter-organizational communication that present-day data exchange standards are too costly, large, slow to respond to industry demands, and complex to develop and use. These problems in data exchange are felt keenly by the manufacturing industry and its vast supply chains. In addressing these challenges, a successful attempt was recently made that involves two major developments. The first is the Core Components Technical Specification (CCTS), which is an ISO-approved meta-model for data exchange standards. CCTS introduces common data types, uniform structure for data models, and data usage semantics. The second is Score, which is a novel open-source software tool. Score was developed by National Institute of Standards and Technology researchers as a platform to take advantage of the CCTS for data exchange standards development and usage. This paper describes the potential of the CCTS and the Score platform, and describes the current status of the Score platform and Score-enabled industry interactions.

Keywords: data exchange, standards, engineering, platform, manufacturing, supply chain

1 INTRODUCTION

Data exchange standards (DESs), including X12, OAGIS, and UBL, enable inter-organizational communication that support many industries [1, 2, 3]. A DES (i.e., exchange or interchange format) is an agreed-upon format (i.e., encoding of information) described formally to enable sharing of information between software systems designed by different organizations under different requirements.

DESs consist of *message schemas* and their *components*, which define valid data exchanges and help achieve an intended interpretation of data by the receiving systems. To support the data exchanges for required variations in business processes, the message schemas often include a very large number of components and relaxed constraints. Within any specific one of these business processes, only certain components will apply to certain situations and will have more restricted constraints on data formats, cardinalities, etc. The DES development is the construction of the message schemas and their components. The DES usage, on the other hand, is the development of usage specification of the message schemas and their components for specific integration use cases. Result of the DES usage specification development, or *profiling*, process is message schema profiles and component profiles ready to be implemented for the integration use case whose requirements governed the profiling process. Profiling is the process of refinement of a message schema or its component type either (i) by removing parts of the schema not relevant to the intended use or (ii) by assembly (or insertion) of components relevant to the intended use or (iii) by a combination of the two. Hereinafter, both profiling outcomes, message schema profiles and component profiles, will be referred as *profiles*.

Best practices today in many industries—including biopharmaceutical manufacturing, agriculture, food manufacturing, aircraft manufacturing, and finance–recognize the need to improve the development and use processes of DESs. The need becomes obvious when the enterprises attempt to improve the agility and resiliency of their production processes and supply chains through pervasive digitalization. Significant system integrations are necessary since outdated development and usage methods for DESs render DES-based integrations costly and time-intensive.

Motivated by the challenges of DES development and use, Core Components Technical Specification (CCTS) [4] has been developed and approved as an ISO meta-model standard with the goal to radically decrease the cost and complexity of DESs development and usage processes. CCTS is based on the ideas of common data types, a uniform structure for data models, and data usage semantics.

While CCTS had the potential to advance the DESs discipline, its actual realization in industry has proven to be elusive since its original release in early 2000. This is changing thanks to a recent development in the form of a novel, open-source software tool called *Score*, which can be viewed as a platform on which multiple industrial applications can be built to support integration. Score makes use of CCTS and is already seeing significant adoption and impact in multiple industries. In fact, the latest release, 10.7, of the popular DES, OAGIS[1], is developed entirely on the Score platform [5].

This paper describes the potential of the CCTS meta-model standard and its capabilities, current state, and on-going industry activities enabled by the novel, open-source Score platform. The paper is structured as follows. Section 2 compares a traditional approach and the new, CCTS-based and Score-enabled approach to using DESs for systems integration. Section 3 introduces key CCTS concepts. Section 4 describes the new Score platform for CCTS-based DESs, enabling a new generation of model-based, syntax-neutral DESs, and their impact on industry. Section 5 lists on-going activities. Section 6 provides some closing remarks.

2 TRADITIONAL AND NEW APPROACHES TO DES USAGE FOR SYSTEMS INTEGRATION: A COMPARISON

This section provides a comparison between a traditional and the new approach to using DESs for systems integration. The traditional approach adopted here uses the OAGIS 10.6 version, which was developed and delivered as a collection of DES message schemas expressed using the XML Schema language. The new approach, on the other hand, is the CCTS-based, Score-enabled approach that allows complete management of the OAGIS 10.7 version without any need for direct manipulation of the DES message schemas using XML Schema or another implementation technology language. Instead, the new approach offers higher-level design concepts concerned with the structure and component contents of the message schemas.

The comparison described below is accomplished by comparing two DES usage processes, in which a systems integration team engages by following the corresponding two approaches for an industrial integration scenario. Through the comparison, the limitations of the traditional approach and the advancements of the new approach, which resolve the limitations, are identified [6].

In comparing the two DES usage processes, the paper follows a *pharmaceutical material replenishment* business-to-business (B2B) integration scenario. In particular, for this illustrative example, the OAGIS *ItemCertificateOfAnalysis* message schema is used to communicate results of a material analysis to get shipment authorization[2]. This type of communication may often be seen, for example, in information exchanges supported by Laboratory Information Management Systems (LIMS) and Enterprise Resource Planning (ERP) systems.

For the purposes of the comparison, we introduce a reference, generalized DES usage process. Fig. 1 shows the generalized process (middle part) that an integration team works through in either of the approaches. This generalized process is then specialized to explain the differences in the two processes enabled by the traditional and the new approach to DES usage, as shown in the Fig. 1.

Following this generalized process for an integration scenario, a systems integration team needs to (1) identify the integration requirements, (2) study the DES design principles of the approach, (3) develop derived integration requirements, following adoption of a DES, (4) identify the target DES message schema, and (5) create a profile of the target message schema. The following subsections discuss each of the generalized process's tasks, and their specializations, which are results of using the traditional and the new approaches.

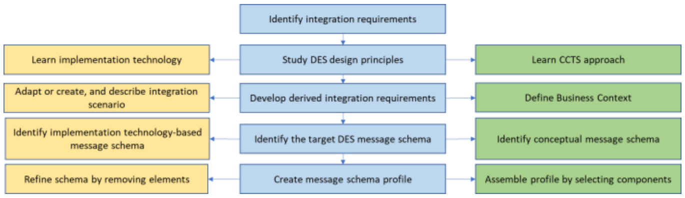

Identify integration requirements
Learn implementation technology
Study DES design principles
Learn CCTS approach
Adapt or create, and describe integration scenario
Develop derived integration requirements
Define Business Context
Identify implementation technology-based message schema
Identify the target DES message schema
Identify conceptual message schema
Refine schema by removing elements
Create message schema profile
Assemble profile by selecting components

Fig. 1. DES Usage Processes: (1) Generalized (middle); (2) Traditional (left); and (3) Newly Proposed (right).

2.1 Task 1: Identify integration requirements

Inter-organizational business processes are coordinated by the exchange of information. The purpose of this task is to (1) identify the information that needs to be communicated as part of the process and (2) where multi-step exchange is required, identify the individual exchanges occurring in process. For example, certain information may be communicated from one party to another in one step and in the next step the receiving party acknowledges receipt. Step 1 is invariant across the traditional and new approaches.

[1] Open Applications Group Inc (OAGi) is a not-for-profit open standards development organization (SDO) that promotes business process interoperability so that software can talk to each other. Its major product is the Open Applications Group Integration Specification (OAGIS) standard.

[2] Certificate of Analysis (COA) includes important technical and business information exchanged along the supply chain, especially in regulated industries such as food and pharmaceutical manufacturing.

2.2 Task 2 Study DES design principles

The goal of this task is to develop the enterprise- and integration situation-specific procedures for DES usage specification, which ultimately lead to creation of the needed DES message schema profiles. Inputs to this task include (1) the enterprise-adopted methodology and architecture for systems integration, driving the selection of an appropriate DES; and (2) systems integration use cases (from the previous task) for which DES message schemas are being adopted and then profiled.

Traditional approach: Following the traditional approach using OAGIS 10.6, the integration team *studies DES design principles*. Specifically, they must become knowledgeable in the underlying implementation technology (e.g., XML, JSON, or Open Application Specification (OAS)). The team needs to answer questions about internals of the standard, such as: (a) What is the internal structure of a message schema? and (b) What does it entail to use the message schema to convey the desired information?

The integration team needs to deal with the complicated structures of message schemas that may have hundreds or thousands of elements that can be hard to trace through. For example, the team needs to deal directly with OAGIS 10.6 message schemas, which have elaborate structure. Within the ItemCertificateOfAnalysis message schema, for example, there are chains of references to generic elements, such as the one from Party, to SupplierParty, which, in turn, involves complex types in an inheritance hierarchy: PartyIdentificationType, PartyBaseType, SupplierPartyBaseType and SupplierPartyType. Such lists of complex types and their inheritances need to be traversed to understand the structure of each element, such as SupplierParty, as illustrated in Fig. 2.

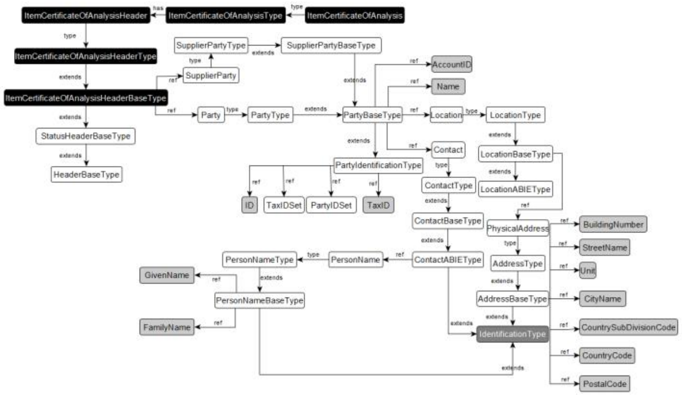


Fig. 2. Illustration of a complex structure in traditional DES.

The traditional DESs are difficult to understand and use owing to the technical complexity of their modeling constructs (e.g., extension/restriction hierarchy, grouping of elements) in implementation-specific languages. While some uses of these modeling constructs aid interpretation of the model, many uses are just modeling shorthand. Moreover, different implementation-specific languages may use different modeling constructs. If the users of a DES require multiple implementation-specific languages of that DES because, for example, different business units use different middleware technologies, they have to learn to navigate and manipulate all of them. Consequently, significant investments and efforts are required to deal with heterogeneity, complexity, and arcane implementation-specific constructs rather than to focus on reasoning about business and engineering concepts relevant to the integration objectives at hand.

New approach: Alternatively, following the new Score-based approach, the integration team *studies DES design principles* by acquiring and applying knowledge of the CCTS-based approach, which is focused on design-decision-making and, therefore, abstracted from, and unburdened by, the implementation technology considerations.

The team may defer learning about, and making decisions on, implementation technology, making the task conceptually simpler and focused. In a CCTS-based approach, the team focuses on the higher-level design concepts that mirror the requirements established in Step 1, which is primarily concerned with determining structure and components of the message schemas suitable to carry the data that successfully encode the information. Fig. 3 illustrates the types of the higher-level design concepts that the integration team is concerned with, which may be treated as parts of a meaningful logical model, reflecting domain concerns. Such logical models are types of constructs that the CCTS-based approach supports.

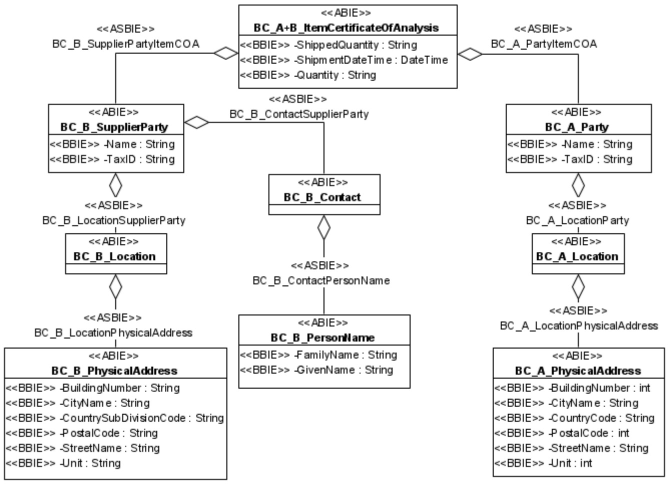

<<ABIE>>
BC_A+B_ItemCertificateOfAnalysis
<<BBIE>> -ShippedQuantity : String
<<BBIE>> -ShipmentDateTime : DateTime
<<BBIE>> -Quantity : String
<<ASBIE>>
BC_B_SupplierPartyItemCOA
<<ASBIE>>
BC_A_PartyItemCOA
<<ABIE>>
BC_B_SupplierParty
<<BBIE>> -Name : String
<<BBIE>> -TaxID : String
<<ASBIE>>
BC_B_ContactSupplierParty
<<ABIE>>
BC_A_Party
<<BBIE>> -Name : String
<<BBIE>> -TaxID : String
<<ABIE>>
BC_B_Contact
<<ASBIE>>
BC_B_LocationSupplierParty
<<ABIE>>
BC_B_Location
<<ASBIE>>
BC_B_ContactPersonName
<<ASBIE>>
BC_A_LocationParty
<<ABIE>>
BC_A_Location
<<ASBIE>>
BC_B_LocationPhysicalAddress
<<ABIE>>
BC_B_PersonName
<<BBIE>> -FamilyName : String
<<BBIE>> -GivenName : String
<<ASBIE>>
BC_A_LocationPhysicalAddress
<<ABIE>>
BC_B_PhysicalAddress
<<BBIE>> -BuildingNumber : String
<<BBIE>> -CityName : String
<<BBIE>> -CountrySubDivisionCode : String
<<BBIE>> -PostalCode : String
<<BBIE>> -StreetName : String
<<BBIE>> -Unit : String
<<ABIE>>
BC_A_PhysicalAddress
<<BBIE>> -BuildingNumber : int
<<BBIE>> -CityName : String
<<BBIE>> -CountryCode : String
<<BBIE>> -PostalCode : int
<<BBIE>> -StreetName : String
<<BBIE>> -Unit : int

Fig. 3. ItemCertificateOfAnalysis - a portion of logical data model.

The Score-enabled, CCTS-based approach uses a modular, implementation-independent representation to develop and use the OAGIS. OAGIS 10.7, which was completely developed and managed in Score, is simplified and made more comprehensive relative to 10.6. This was achieved by using CCTS model-based components instead of the XML or other implementation-specific concepts. These model-based components can be presented, say, using UML class diagrams stereotyped for CCTS, as shown in Fig. 3.

2.2 Task 3: Develop derived integration requirements

The goal of this task is to develop derived requirements expressions following from the selected DES. The task uses the enterprise- and integration situation-specific procedures for DES usage specification, which were developed in the prior task. In addition, the input includes the systems integration use cases for which DES message schemas are being adopted and profiled, as identified by the enterprise.

Traditional approach: Following the traditional approach, such as the OAGIS 10.6, the integration team *develops derived integration requirements* (using the adopted DES approach design principles) by reviewing, selecting, and adapting one of the proposed integration scenarios in the DES approach, which are offered as a non-normative starting point for the requirements description.

To select and adapt an integration scenario, the team needs to go through a collection of generic integration scenarios provided by OAGIS. These scenarios are expressed as UML sequence diagrams and need to be adapted in order to meet the identified integration requirements. Each scenario involves a number of generic message schemas. By manually reviewing and analyzing some 80 existing OAGIS-provided integration scenarios, one would hopefully find the most appropriate one. Alternatively, if the DES provides no suitable scenario, the team creates one, using natural language to describe it. In either case (existing or bespoke scenario) they use the DES message types in detailed specification of the scenario.

For expository purposes, suppose the team adopts *Scenario #70 Supplier Provided Certificate of Analysis* shown in Figure 4. This scenario provides a starting point for several alternative integration flows for reporting the results of a material analysis in order to get authorization for the material shipment. Four business objects–Shipment, InspectionOrder, TestResults and ItemCertificateOfAnalysis–may be paired up with various verbs/actions, such as Process or Acknowledge, relevant in the scenario. A verb and a business object pair compose a message schema, for example, ProcessItemCertificateOfAnalysis.

In the case of OAGIS-provided integration scenario, the same message schemas may appear in various integration use cases and in different orders. Additionally, some message schemas identified may not be used or message schemas not identified in the scenario may be needed in one's own specific integration use case. The integration team needs to solve the problem of finding and selecting an integration scenario and its existing message schemas, and, then, adapting them to the integration use case at hand, which is a non-trivial, heuristic exercise that requires prior expertise in applying the specific DES. Often, integration teams would spend days or weeks combing through a message schema to identify elements suitable for an integration use-case objective.

The present-day DESs largely do not capture precisely the business's intended use. The "out-of-the-box" DES schema types are aggregates of many elements and are largely agnostic with respect to specific integration use cases. They provide little guidance for their usage in specific use cases. They aggregate many elements that are needed to enable similar uses, yet each of which has specific integration requirements, (e.g., data formats, mandatory elements, etc.) and intended use. For example, in OAGIS prior to version 10.7, there is no possibility to record the intent of the standard user-defined profiles in a way that would enable search of these profiles that are possibly relevant for a new integration use case.

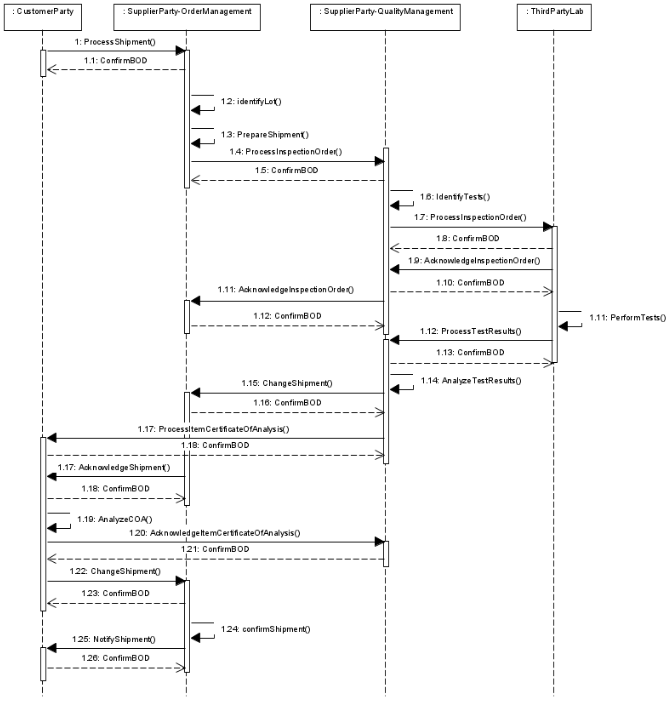


Fig. 4. OAGIS Scenario #70: Supplier Provided Certificate of Analysis.

New approach: On the other hand, following the new approach, the integration team *develops derived integration requirements* (using the adopted DES approach design principles) by applying the CCTS-based Business Context approach to encoding the integration scenario specifics, capturing the intended use of the required message schema type or its components.

To define a specific business context (BC) for a systems integration use case, the systems integration team would (a) reference (if provided) or define a collection of business categories for the integration domain, (b) reference or define lists of values for each of the categories from the collection, and (c) define a BC by choosing values from the BC categories' lists of values. In the example integration use case, four significant context categories are identified: Business Process Role, Geo-political, Activity, and Industry. Table 1 illustrates a possible BC for the example integration use case. The roles in the use case become context values for the Business process role category: Receiving manager from Company A, and Quality manager from Company B. The enterprises are from Serbia and USA, and these are values for the Geo-political category. Other category values could be countries from all around the world. The Industry category is assigned the Pharmaceutical value (short for the Manufacture of Pharmaceuticals, Medicinal Chemical and Botanical Products) from the International Standard Industrial Classification of All

Economic Activities (ISIC) [17]. Lastly, the Activity category uses the values from the actual business processes model for the use case.

Table 1: Business Contexts for supplier and customer parties of the integration use case

	Business context	
Context Category	BC_A	BC_B
Business process role	Receiving manager	Quality manager
Geo-political	Serbia	USA
Industry	Pharmaceutical	Pharmaceutical
Activity	Validate COA	Request COA Validation

COA: Certificate of Analysis

In the CCTS-based approach, the limitation of the traditional approaches is addressed by specifying the DES profile's business context (BC). With BC, the integration team has a way to define precisely intent of a message schema or component, as defined in the CCTS standard. CCTS suggests use of eight BC categories to define uniformly a BC. A BC, and a conceptual component, together, define DES profiles and document their usages both at the DES schema and at the component level. In such a way, the BC notion introduces precise descriptions of the usage intent for specific integration use cases, enhancing the capacity for their reuse.

2.3 Task 3: Identify the target DES message schema

The goal of this task is to develop or identify an existing message schema (from the selected DES), capable of meeting data exchange requirements. This message schema, then, provides design space where (in the next task) the team, through a profiling process, determines the target data exchange solution: a profiled message schema. The task also uses the enterprise- and integration situation-specific procedures for DES usage specification, developed in the first task. In addition, the input includes the DES-specific derived requirements expressions from the previous task.

Traditional approach: Following the traditional approach, the integration team *identifies the target DES message schema* by identifying a DES-provided implementation technology-based message schema.

To select an adequate message schema, the team needs to inspect all identified message schemas from the chosen integration scenario (Scenario #70, in this example) and other parts of the OAGIS standard and, after analyzing their structure, select the most appropriate among them (the ProcessItemCertificateOfAnalysis, in this example). The integration team would do such manual analysis, potentially, for each integration use case. The manual nature of this step impedes identification of existing profiles (discussed next), even when those profiles are developed for a similar integration use case. This has an impact on achieving interoperability with existing or future systems that use this DES but cannot easily re-use the existing profiles.

The current message schemas are difficult to interpret and use consistently across their use cases because their definitions and usage guidelines are expressed imprecisely and ambiguously using natural languages for human interpretation. This state of DESs, also negatively effects integration and interoperability of systems using the traditional DESs.

New approach: On the other hand, following the new approach, the integration team *identifies the target DES message schema* by identifying the CCTS-based conceptual message schema.

In the CCTS-based Score approach, the lacking ability of the traditional approach to identify relevant existing profiles is bypassed by using a CCTS-based OAGIS standard, such as the OAGIS version 10.7. By analyzing conceptual data-model components (CCTS core components), an appropriate DES schema is chosen. Then, using the specified business context (BC), the components from the conceptual model are used to create a logical data model, which represents a usage-specific DES schema model. At that time, existing relevant profiles, or BIEs, are identified for their reuse. Once this BC-enabled logical data model is created, it is possible for it to be transformed into XML Schema or JSON Schema (or any other schema specification language). Such logical models can then be readily reused in any similar integration use case, as specified by the BC definitions.

2.4 Task 4: Create message schema profile

The goal of this task is to create a profile of the message schema type created or identified in the previous task, meeting the data exchange requirements. The task also uses the enterprise- and integration situation-specific procedures for DES usage specification, which were developed in the second task. In addition, the input also includes the DES-specific requirements expression and the DES-specific message schema type defined in the previous tasks.

Traditional approach: Following the traditional approach, the integration team *creates message schema profile* by refining the implementation technology-specific schema. This is accomplished by removing existing, irrelevant elements from the schema and, possibly, adding new, missing elements.

The team profiles the identified message schema and its components for the specific integration use case. To create the profiles, the team needs to deal with implementation constraints inherited in the adopted language. For example, in XML Schema, there are no provisions for replacing the type of an existing information element, therefore new elements must be added rather than old type adapted when a new type is needed. For example, *SupplierParty* may exist, but special requirements in *BioPharma* may force a definition of *BioPharma_SupplierParty*. In a similar manner, another element, *BioPharma_Contact* also may need to be created to indicate that *PersonNameType* uses the *FamilyName* and *GivenName* elements, for a specific integration use case. One needs to follow this procedure for each reused element and underlying type that require restrictions or extensions. The integration team needs to deal with complex maintenance of element qualifications and connections between the added elements and their original versions.

Message schemas often contain many data elements useful only in infrequently occurring business situations. Time-consuming profiling, may be needed to eliminate the unused elements. Also, significant time and effort is required owing to the need to create profiles independently in each required implementation technology-specific language.

New approach: On the other hand, following the new approach, the integration team *creates message schema profile* by assembling the profile, which is accomplished by selecting the relevant components from a component library.

The team creates component profiles (i.e., BIEs) for the defined business context by re-using existing ones from a library of Core Components (CCs) and BIEs. (Please refer to Section 3 for description of the CC and BIE terms.) In this task, the team references the BC in which the BIEs are intended to be used. Logical data models of the BIEs may be created using UML class diagram or similar; a part of a UML diagram showing the ItemCertificateOfAnalysis BIE is presented in Fig. 3. This is an important step because it separates the implementation-specific details from the conceptual and logical model. All BIEs are created by restricting the corresponding CCs. The CC library is created specifically for OAGIS and contains all the required concepts. Therefore, there is no need to introduce new ad-hoc components. BC prefixes are added to these names in the diagram to indicate the BIEs' intended use.

Following the CCTS-based approach, the integration team does not work at the implementation-specific XML Schema level; instead, the team works at the implementation-independent, conceptual, and logical levels (as illustrated in Fig. 3). The ability to create implementation-neutral, context-aware model of OAGIS is responsible for this advancement. BIE structures normalize the inheritance hierarchy of each Core Component (CC), and the integrator can manage each CC as it exists independently. Also, all of the needed CCs may be restricted through corresponding BIEs, circumventing any unnecessary introduction of new CCs. In this way, reusability of existing BIEs is increased. When a logical model is completed, it can be transformed into an implementation-specific model; Score currently supports XML, JSON schema and Open API 3.0 serializations. In this way, the XML Schema-specific requirements, which forced the introductions of new elements and types in order to customize original OAGIS Schema, are avoided.

3 ISO CORE COMPONENTS TECHNICAL SPECIFICATION

e-Commerce technologies developed at the end of the 20th century, including XML, HTTP protocol, Web client-server architectures, enabled the development of ebXML, a cross-industry initiative that addressed the increasingly complex state of e-Commerce practice. One impactful part of ebXML is the Core Components Technical Specification (CCTS), which provides a new foundation for DESs [4].

CCTS is an advanced ISO-standardized meta-model that allows implementation technology-neutral standardization. CCTS introduces two fundamental types of data modeling components. First, *Core Components* (CCs) are abstract DES building blocks. CCTS offers a set of CC types to support the CC use at different granularity levels (i.e., message schemas and components) of a DES. Second, *Business Information Entities* (BIEs) are the profiles of corresponding CCs devised for an integration use case. BIEs restrict related CCs for a specific *business context* (BC) in which they are intended for use. A use-specific collection of CCs is denoted as *conceptual data model*, while a corresponding collection of BIEs is denoted as *logical data model*. BC is established as a model-based specification of usage semantics for a BIE. UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) defines a BC specification to consist of context values assigned to corresponding context categories [8]. There are eight BC categories proposed by UN/CEFACT, including business process, geo-political region, and industry. These categories capture important characteristics of an integration use case to be expressed succinctly by a BC expression.

CCTS is a significant advancement that offers new model-based concepts that greatly improve DESs. First, the CCTS-based CC and BIE components are represented in a syntax that does not require any commitment to a particular implementation solution. With additional specifications between a CCTS-based data model and its representation in an implementation technology (often called the Naming and Design Rules or NDR), BIEs can be readily transformed into XML Schema, JSON Schema, or another implementation solution. Second, the CCTS concepts allow innovative tools for the life-cycle management (LCM) of DESs. Score is one such tool that builds on CCTS advancements and it will be discussed next using the same *pharmaceutical material replenishment* B2B scenario.

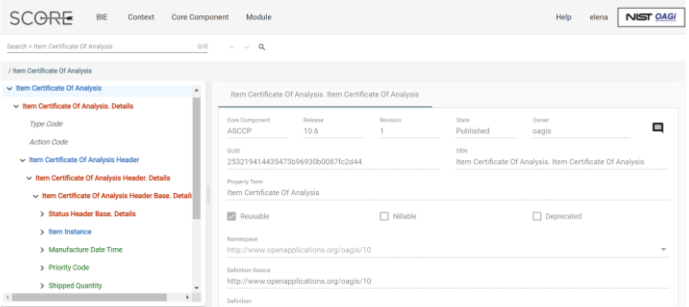
4 THE SCORE PLATFORM – CURRENT CAPABILITIES

This section describes the new Score platform, which enables the new generation of CCTS-based DESs and processes described in Section 2. Score was created by the National Institute of Standards and Technology (NIST) Engineering Laboratory working with industry representatives as part of the Open Applications Group, Inc. (OAGi) Semantic Refinement Working Group. Score can be viewed as a DES platform for two reasons. First, numerous industrial applications can be built on the platform, making use of its resources. Second, data exchange standards can be built entirely on the platform, as demonstrated by the latest release of OAGIS.

Score provides several capabilities that remove the limitation of current DESs. It streamlines the CCTS-based DES development as described earlier in this paper via web-based, DES-focused graphical user interface (GUI). The DES-focused interface (i.e., GUI providing CC or BIE views, as specified in CCTS and managed in Score) is an important capability, allowing easier manipulation of DES components than is possible using generic XML or JSON editors. Through engagements with industry users, Score was shown to provide required operational support for enterprise integration architects, software developers, and standards architects. The following are key new capabilities of the Score platform.

- *Create and manage (i.e., copy, edit, update, etc.) CCs.* Score enables life-cycle management of the DES, DES libraries, and canonical DES models that are the basic data exchange artifacts. Figure 5 displays a screenshot of the Score browser-based GUI allowing revision management of CCs, showing the *ItemCertificateOfAnalysis* CC. This capability is specifically important because it enables development and potential extensions of the Score Core Component library.

- *Create and manage Business Contexts (BCs), including the BC categories, category schemas, and schema values*. Score enables the creation of a reference design contexts and expression of intended usage of BIEs. Figure 6 shows the Score GUI allowing life-cycle management of BCs, using the BC example from Table 1.



/ Item Certificate Of Analysis

- Item Certificate Of Analysis
  - Item Certificate Of Analysis. Details
    - *Type Code*
    - *Action Code*
    - Item Certificate Of Analysis Header
      - Item Certificate Of Analysis Header. Details
        - Item Certificate Of Analysis Header Base. Detail
          - Status Header Base. Details
          - Item Instance
          - Manufacture Date Time
          - Priority Code
          - Shipped Quantity

Item Certificate Of Analysis. Item Certificate Of Analysis

Core Component	Release	Revision	State	Owner
ASCCP	10.6	1	Published	oagis

GUID: 253219414435473b96930b0087fc2d44

DEN: Item Certificate Of Analysis. Item Certificate Of Analysis

Property Term: Item Certificate Of Analysis

☑ Reusable ☐ Nillable ☐ Deprecated

Namespace: http://www.openapplications.org/oagis/10

Definition Source: http://www.openapplications.org/oagis/10

Definition

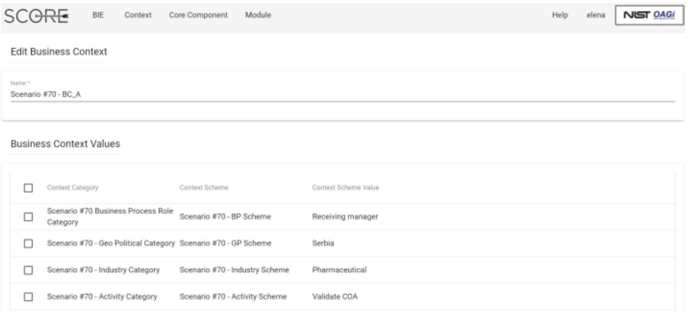
Fig. 5. A screenshot of the Score user interface allowing CC life-cycle management, showing the ItemCertificateOfAnalysis CC.

- *Create and manage (i.e., copy, edit, update, etc.) BIEs.* Score enables creating BIEs that are based on CCs. Figure 7 displays a screenshot of the Score GUI allowing life-cycle management of BIEs, showing the *ItemCertificateOfAnalysis* BIE example from Figure 3.

- *Assemble CCs and BIEs.* Score enables the creation of assemblies of complex data exchange artifacts (including standard business documents or DES message schemas).

- *Collaborate on CC & BIE specifications.* Score enables teams of integration architects and developers to work collaboratively on designing data exchange artifacts.

- *Life-cycle management (LCM) of CCs & BIEs*. This enables multiple versions of BIEs that are based on various CCs releases to be hosted and managed in a single repository as a reliable source of truth for DES artifacts. Additionally, it provides automations that allow enterprises to efficiently manage changes in CCs and the transitions in BIEs between the underlying CC releases.



Edit Business Context

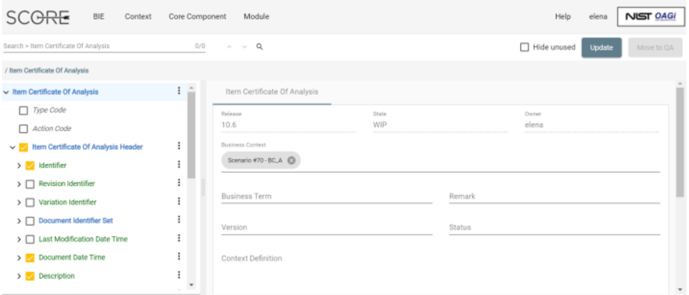
Name *: Scenario #70 - BC_A

Business Context Values

☐	Context Category	Context Scheme	Context Scheme Value
☐	Scenario #70 Business Process Role Category	Scenario #70 - BP Scheme	Receiving manager
☐	Scenario #70 - Geo Political Category	Scenario #70 - GP Scheme	Serbia
☐	Scenario #70 - Industry Category	Scenario #70 - Industry Scheme	Pharmaceutical
☐	Scenario #70 - Activity Category	Scenario #70 - Activity Scheme	Validate COA

Fig. 6. A screenshot of the Score user interface allowing the life-cycle management of BCs.

- *Generate an implementation-specific BIE.* This enables the automated production of BIEs in an implementation technology of choice—currently XML, JSON, or OAS. Figure 8 shows an *ItemCertificateOfAnalysis* BIE that was automatically expressed by Score as an XML Schema.



/ Item Certificate Of Analysis

- Item Certificate Of Analysis
  - ☐ *Type Code*
  - ☐ *Action Code*
  - ☑ Item Certificate Of Analysis Header
    - ☑ Identifier
    - ☐ Revision Identifier
    - ☐ Variation Identifier
    - ☐ Document Identifier Set
    - ☐ Last Modification Date Time
    - ☑ Document Date Time
    - ☑ Description

Item Certificate Of Analysis

Release	State	Owner
10.6	WIP	elena

Business Context: Scenario #70 - BC_A

Business Term

Remark

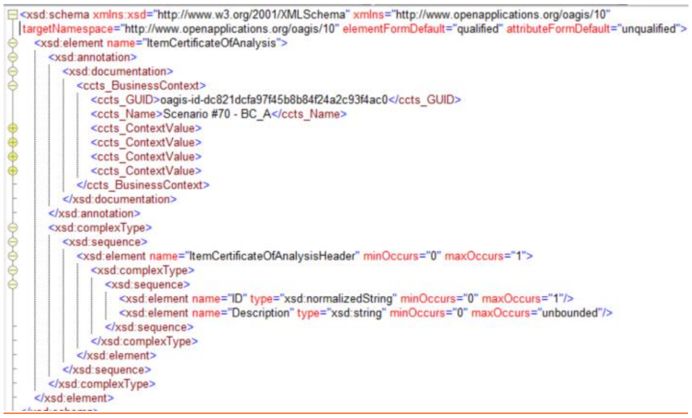
Version

Status

Context Definition

Fig. 7. A screenshot of the Score user interface allowing life-cycle management of BIEs, showing the ItemCertificateOfAnalysis.

Figure 9 shows a snippet of the Score MySQL database schema. In Score 2.0, the database schema consists of 75 tables and 337 relations. It represents the realization of the CCTS meta-model with additional enhancements in a relational database. These additional tables and relationships are required to support user access and the life-cycle management (LCM) of CCs and BIEs from DES developers and users while allowing them to achieve the cost effectiveness and efficiency promised by DES-based integration. The figure illustrates tables that map directly to the CCTS meta-model entities like ACC (Aggregate CC) and BBIE (Basic BIE), tables that support release management such as the RELEASE and ACC_MANIFEST, and tables that support serializations into implementation-specific syntaxes such as MODULE (file) and CDT_AWD_PRI_XPS_TYPE_MAP (implementation-specific data type mappings).



```xml
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.openapplications.org/oagis/10"
targetNamespace="http://www.openapplications.org/oagis/10" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="ItemCertificateOfAnalysis">
        <xsd:annotation>
            <xsd:documentation>
                <ccts_BusinessContext>
                    <ccts_GUID>oagis-id-dc821dcfa97f45b8b84f24a2c93f4ac0</ccts_GUID>
                    <ccts_Name>Scenario #70 - BC_A</ccts_Name>
                    <ccts_ContextValue>
                    <ccts_ContextValue>
                    <ccts_ContextValue>
                    <ccts_ContextValue>
                </ccts_BusinessContext>
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ItemCertificateOfAnalysisHeader" minOccurs="0" maxOccurs="1">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="ID" type="xsd:normalizedString" minOccurs="0" maxOccurs="1"/>
                            <xsd:element name="Description" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Fig. 8. An ItemCertificateOfAnalysis BIE expressed automatically by Score as an XML Schema.

One of the key advancements that Score brings to the traditional standard development and use processes is the linking between a DES (e.g., the CC entities) and the usage specification (e.g., the BIE entities), which is now also structurally represented (as opposed to using a word or a spreadsheet processors). The other capability enabled by the CCTS-based representation of a DES in a relational database is that it offers finer-grain changes in tracking, profiling, and management at the component level compared to the file level used in the past. This allows the user to more efficiently and effectively manage the life-cycle of the DES and integration artifacts including their serializations in various implementation-specific platforms.

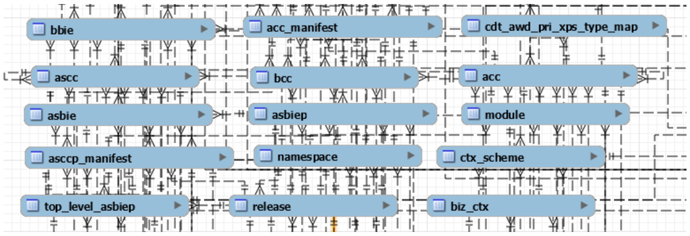


Fig. 9. A graphical view of a snippet of the Score 2.0 database schema. Lines represent relationships between tables.

To support enterprise applications, Score provides new DESs development and usage capabilities within existing workflows and in collaboration with other tools. For example, Score provides DES artifacts for applications programming interface (API) development, allowing integration teams to design and profile message schemas and components and generate target schemas that are ready to be implemented within a collection of existing API development tools. Implementation schemas may be generated in a variety of implementation technologies, including XML Schema, JSON Schema, and Open API Specification (OAS). For example, Score was shown to generate valid OAS 3.0.2 BIEs that were uploaded in API editors for use.

5 THE SCORE PLATFORM – ON-GOING WORK

5.1 On-going Score-supported industry activities

This section accounts for the support provided by the Score platform in a number of diverse, on-going industry activities.

OAGIS model-based, implementation-independent, software platform for the standard life-cycle management. At the June 2019 "NIST Open Industrial Digital Ecosystem Summit and OAGi Symposium," Jim Wilson, the President and CEO of OAGi, remarked: "NIST Score is the future of OAGIS and the OAGIS + Score system is currently used by large OAGIS users such as Boeing, Land O' Lakes, and Lockheed Martin. Additionally, the NIST researchers are working to enable Score to work on other DESs such as supply chain standards for chemical and agricultural industries, and oil and gas asset management standards" [9].

The OAGIS + Score is widely available through the MIT Open Source License agreement. The open source license decision was made to support both NIST and OAGi's goal of broadening the adoption of the OAGIS + Score DES to maximize the data exchange efficiencies in industrial organizations' systems.

Most recently, OAGIS has advanced to the new version 10.7, which, for the first time, is managed and updated entirely using Score. OAGi stated that "OAGIS 10.7 marks a major milestone in OAGi's history with OAGIS improvements made exclusively in Score, our tool for semantic library management" [5]. The latest Score 2.0 is deployed as SaaS by OAGi and can be accessed at https://oagiscore.org/. In addition, the tool is deployed on-premise by a number of OAGi member companies.

Biopharmaceutical manufacturing. The Score team is engaged with NIIMBL, the National Institute for Innovation in Manufacturing Biopharmaceuticals [10], to help develop DESs for data exchange requirements in the NIIMBL Big Data program. Score is of interest as a tool that can help accelerate the development of DESs by both efficiently reusing and further adapting existing BIEs from other industries, such as food manufacturing, developed already in Score. Current focus is in the area of ingredient testing and reporting testing results, where standards such as the Certificate of Analysis (CoA) play a significant role. There is also interest in additional DESs that support biopharmaceutical production processes, such as formulation, recipe, production order, work center operations, and work-in-process status [11].

Food manufacturing. The Score team has been working with AgGateway, a global organization developing the resources and relationships that drive digital connectivity in global agriculture and related industries [12], on a number of aspects of DESs. As an example, Score was leveraged in the AgGateway Product Catalog Working Group to "achieve their goal to assure online stores have complete, accurate and up-to-date information in their Product. The first phase – providing the minimum data set required to populate a retail-facing eCommerce catalog – has been completed. Specifically, AgGateway's Standards & Guidelines Committee recently approved the OAGIS Catalog JSON

Schema v1.0 profile using the NIST/OAGi Score Tool" [13]. The Catalog BIE designed and generated in Score was successfully used to demonstrate its role in an enterprise systems integration workflow.

Small and Medium Enterprises (SMEs). The Score team participates in the OAGi SME Working Group to support development of a well-defined, useful, and extensible collection of message schemas, called OAGIS Express Pack, for use by Small and Medium Enterprises [14]. Score is playing two key roles in this initiative. First, Score is the vehicle to efficiently define Minimal Viable Product (MVP)of the OAGIS DES which would continue to be in-synch with OAGIS, and then generate easy-to-use schemas in implementation technologies of choice. The BIEs would reflect the requirements gathered from SMEs over many years in an 80–20 principle (i.e., 80% of the users need only 20% of the product) approach. Second, Score is a vehicle to assure that the 80–20 principle-based approach is meaningful and allows rapid adaptation and addition of new data elements to the Express Pack schemas from the comprehensive OAGIS standard. A large portion of OAGIS message schemas have been profiled for the Express Pack and used to generate on demand XML, JSON, or OAS Schemas, for typically needed exchanges in procure-to-pay integration scenarios (including Purchase Order and Shipment Notice). A first release of the Express Pack is planned for March 2021.

Financial industry. The Score team works with the Business Payments Coalition- (BPC) led e-Invoice Semantics Workgroup [15] to support development of the semantic model for the electronic invoicing capability for the North American markets, including support for the Small and Medium Enterprises. Score is being used to support mapping of the BPC-adopted UBL (Universal Business Language) DES onto the OAGIS standard. This is expected to help the SME-focused OAGIS Express Pack address the BPC e-Invoice requirements so the latter can be used more immediately by SMEs. In addition, Score allows OAGIS to be immediately usable in the future e-Invoice network as the DES in supplier and customer connection links and gateways to the network. The BPC will be undertaking an assessment of the e-Invoice Exchange Framework Standards in 2021 and the Score platform will be explored as a potential part of the Standards Oversight Assessment capability, in addition to the helping SMEs adopt e-Invoicing.

5.2 New challenges and on-going research in Score

Score is being used to provide industries with innovative capabilities, and many research and development activities are being pursued to enhance these capabilities.

Business Process Cataloging and Classification System (BPCCS) is a prototype system developed at NIST to manage business process models underlying integration activities. Business process models may provide significant part of Business Context information that describes the design intent, or intended usage, of a BIE. Central to the system is its business process model life-cycle management capability.

BPCCS integration with Score is explored to facilitate robust, efficient, and precise definition of DESs by direct ingestion and use of business process model information describing the integration situation at hand. Today, Business Context definitions are entered manually in Score. By connecting BPCCS with Score, and minimizing human intervention, a greater precision in DESs development will be possible. In addition, greater reuse and collaboration will be possible by providing a referential basis for domain-specific terminology and a knowledge base that can be established using the business processes models of integration scenarios.

The multi-standard Score platform is an effort that enables additional DESs being moved from their implementation-specific forms to the CCTS-based implementation-independent form. First in this group was the Chem eStandards. Other standards, such as AgXML, are also being reviewed for their re-representation using CCTS and inclusion in Score.

One benefit of the multi-standards inclusion in Score is that it may allow components from different DESs to be used and combined as needed, which actually would address a frequent industrial requirement [16]. Another benefit of having overlapped or complementary standards in Score is that a standard message or document may be expressed in any number of implementation technologies, per specific data exchange requirements. In addition, the OAGi Mapping Specification Working Group is currently investigating the potential of making mapping specifications and artifacts easier to use and reuse when they are done in CCTS-based DESs, as opposed to DESs with different base representations, such as XML Schema/XSLT and JSON Schema/JOQL.

Advanced Business Context (BC) and Reuse investigates novel approaches to representing and reasoning with Business Context to efficiently reuse BIEs. BC captures the intent of using a BIE. Capturing that intent is essential to finding and reusing the right BIEs, which enables interoperability of corresponding data and services. Following CCTS, a specific business context is represented as a set of context values that are associated with their corresponding context categories (e.g., business process, geo-political location, business process activity, actor, resources available, and industry).

While BC can indeed identify situations in which some BIEs will be reused, one must also consider what happens in the background (e.g., transformation of the data into the database). Hence, BC must provide meaningful definitions and also help identify validation rules that must be triggered when choosing the BIEs in the usage context.

This research area investigates recently proposed logic- and ontology-based techniques [16]. Business-context knowledge is viewed as a collection of decentralized, acyclic graphs. Each business-context category has at least one such graph that represents a list or taxonomy of possible values in the corresponding category. Then, a specific business context for a specific information exchange situation associates a collection of nodes belonging to specific categories in the graph. Formal representations of Business Context knowledge and algorithms to reason over the knowledge carry promise of greater usability and precision of the CCTS-based standards.

Agile standards life-cycle management is a research area seeking to enable frequent releases of the standard with better quality and faster, more reliable adoption cycles. In the past, each OAGIS minor release, for example, could take six months to two years to finish. In addition, some integration environments stayed with the same release of OAGIS for ten years. The reasons are 1) DESs were centrally developed by a part-time standard architect, 2) DESs were developed with standalone file-based tools, 3) DESs were developed and delivered in an implementation/technology-specific form, and 4) the implementation guide and message schema profiles and component profiles were developed using office-suite tools and were totally disconnected from the DESs themselves. While the state of practice enabled by the current Score capabilities outlined in Section 4.2 has already addressed some of these issues, that effort is just the tip of the iceberg. Research in this area may include 1) integrating Score with tools such as GitHub and Continuous Integration/ Continuous Development (CI/CD) tools; 2) employing semantic search and schema-matching methods to help users in discover data elements, upgrade BIEs between OAGIS releases, and even automate some mappings; 3) providing platform and service capabilities so that analytics and extensions can be easily added to the tool as BIE quality metrics; and 4) link CCs and BIEs to enterprise data dictionaries to facility business analytics.

6 CLOSING REMARKS

The central claim in this paper is that the data exchange standards discipline is on a precipice of a major transformation that will usher in more advanced data exchange standards. The claim was supported by a description of the ISO CCTS meta-model standard and the novel, NIST-developed, open-source Score platform with potential for major industrial impact.

Based on CCTS, the novel, open source Score platform provides advanced capabilities by offering rich data exchange standards life-cycle management functions at the conceptual and the logical levels, as well as at the implementation level of the data exchange standards. Its industrial impacts, which are already being seen in the OAGi member community, has the potential to grow as the open-source Score community picks up steam, and additional standards development organizations contribute and participate in the Score platform. Early feedback and intensive engagement from industry participants has been very encouraging, motivating further the Score development team.

DISCLAIMER AND ACKNOWLEDGEMENT

Certain commercial systems and applications identified in this paper are not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that they are necessarily the best available for the purpose.

REFERENCES

[1] X12, https://x12.org

[2] Open Applications Group, Inc., https://oagi.org

[3] OASIS Universal Business Language (UBL) TC, https://oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

[4] UN/CEFACT, Core Components Technical Specification CCTS, version 3.0.,2009

[5] NIST/OAGi Score, https://oagiscore.net

[6] Jelisic, E., Ivezic, N., Kulvatunyou, B., Anicic, N., & Marjanovic, Z. (2019, September). A Business-Context-Based Approach for Message Standards Use-A Validation Study. In European Conference on Advances in Databases and Information Systems (pp. 337-349). Springer, Cham.

[7] OAGIS 10.7 Standard and Enteprise Editions, Last accessed 9 February 2021, https://oagi.org/Resources/tabid/143/Default.aspx

[8] UN/CEFACT Unified Context Methodology Technical Specification, https://www.oasis-open.org/. Last accessed 05 November 2020

[9] NIST Open Industrial Digital Ecosystem Summit and OAGi Symposium agenda, https://www.nist.gov/system/files/documents/2019/05/31/digitalized_industry_summit_agenda.pdf

[10] The National Institute for Innovation in Manufacturing Biopharmaceuticals, https://niimbl.force.com/s/

[11] Fermier, A., McKenzie, P., Murphy, T., Poulsen, L., & Schaefer, G. (2012). Bringing New products to Market Faster. Pharmaceutical Engineering

[12] AgGateway, https://aggateway.org

[13] An Overview of Current and Planned AgGateway Work, January 2021. https://www.aggateway.org/Portals/1010/WebSite/CURRENT%20WORK/Activity%20Summary%20-%20Jan%202021.pdf?ver=2021-01-04-191838-807

[14] Small & Medium Enterprise Working Group, https://oagi.org/OAGISWorkingGroups/WorkingGroups/tabid/149/Default.aspx

[15] Business Payments Coalition e-Invoicing, https://businesspaymentscoalition.org/e-invoicing/

[13] IIoT Message Modeling for Enterprise Integration and Interoperability – A White Paper from OAGi and MIMOSA, https://www.mimosa.org/wp-content/uploads/white-papers/2020/06/iiot-message-modeling-for-enterprise-integration-and-interoperability/IIoT-Message-Modeling-for-Enterprise-Integration-and-Interoperability-EN-US-Letter.pdf

[16] Novakovic, D., & Huemer, C. (2016). Applying business context to calculate subsets of business document standards. Information Technology and Management, 17(3), 203-227.

[17] Department of Economic and Social Affairs Statistics Division. International Standard Industrial Classification of All Economic Activities (ISIC), Revision 4. Last accessed 25 December 2020