Designing a Contractual Choreography Language for Automating Cross-Organizational Business Collaboration

Alex Norta*, Lixin Ma†, Yucong Duan‡

*Tallinn University of Technology, Akadeemia Tee 15A, 12618 Tallinn, Estonia
†Hainan University, Haikou, China
‡University of Shanghai for Science and Technology, Shanghai, China
Email: alex.norta@gmail.com, duanyucong@hotmail.com, malixin.usst@gmail.com

Abstract—Automating business collaboration promises efficiency and effectiveness increases with means of service-oriented cloud computing. For automating cross-organizationally connected business processes operation, existing choreography languages are typically technology driven and focus less on their socio-technical suitability and expressiveness that recognizes the interaction between people in organizations and technology in workplaces. The eSourcing Markup Language (eSML) addresses the gap, extracted from our studies of suitability and expressiveness for supporting the automation of business collaboration. eSML has been applied in case studies and its development approach is replicable for exploring strengths and weaknesses of other choreography languages.

Keywords-Choreography language, eSourcing, suitability, expressiveness, cross-organizational, B2B, business process, sociotechnical

I. INTRODUCTION

With the emergence of new automation paradigms such as service-oriented computing (SOC) and cloud computing (CC), the way companies collaborate with each other experiences significant changes. SOC [7] comprises the creation of automation logic in the form of web services. In CC [32], the Internet is used to access web-based applications, web services, and IT infrastructure as a service. Web services [8] are an important vehicle for enabling organizations to cooperate with each other by cross-organizationally linking business processes [22], [23], [34], [35], [18] with choreography languages for the purpose of electronic outsourcing.

With respect to existing choreography languages, the most notable are versions of the Business Process Execution Language such as AbstractBPEL [29] and BPEL4Chor [14], Web Services Choreography Description Language (WS-CDL) [21], Business Process Modeling Notation (BPMN) [13], [45] Let’s Dance [46], ebXML BPSS [16] and more recently, the Business Choreography Language (BCL) [31]. However, not only existing choreography languages but also other XML-based languages for SOC have not been adopted by industry as expected. A reason is the approach for language development that does not take into account socio-technical suitability and expressiveness deficiencies that recognizes the interaction between people in organizations and technology in workplaces. Suitability means that choreography languages comprise concepts and properties to allow the formulation of real-world business-collaboration scenarios in many perspectives. Expressiveness means the constructs of a choreography language have semantic clarity for ensuring uniform enactment behaviour by different business process engines. Additionally, business collaboration is based on contractual agreements. This paper fills the gap by answering the research question how to systematically develop a language for cross-organization and contract-based collaboration specifications. From there we deduce several sub-questions. What is the collaboration context and model the specification language must cater for? What are the main suitability- and expressiveness concepts and properties?

The structure of the paper follows the design-science method [20] for the development of eSML and is as follows. Section II presents a business-collaboration model that evolves from case studies in the research project called CrossWork [19], [30], namely eSourcing [34], [35], [36]. In Section III, we further explore the collaboration model in a pattern-based way [34], [36] with the objective of generating the essential concepts for eSML to gain business-collaboration suitability. Next, assuming the control-flow perspective is dominant for enacting business collaborations, we present in Section IV the expressiveness-assurance in eSML. Section V presents the resulting structure in eSML. Section VI discusses a "proof-of-construction" application system. Finally, Section VIII concludes this paper and discusses future work.

II. BUSINESS COLLABORATION MODEL

In the EU research project CrossWork [19], observing business collaborations of industry partners reveals characteristic features. An original equipment manufacturer (OEM) develops the creation of value chains
in an inhouse business process according to different perspectives, e.g., control flow of tasks, information flow, personnel management, allocation of production resources, and so on. The CrossWork case studies reveal that the basis for business collaboration between organizations are contracts. The basis has implications for the suitability exploration in the sequel.

Figure 1. eSourcing in three-level business-process.

As an explanation of vertical business collaboration, Figure 1 depicts a three-level model as part of an eSourcing example [34], [35], [36], [18]. The three-level model is instrumental for not forcing collaborating parties into connecting their information infrastructure directly. The processes in Figure 1 depict the control-flow perspective of the eSourcing concept that focuses on structurally harmonizing on an external level the intra-organizational business processes of a service consuming and one or many service providing organizations into a business collaboration. Important elements of eSourcing are the support of different visibility levels of corporate process details for the collaborating counterparts and flexible mechanisms for service monitoring and information exchange. Recently, leading IT-enterprises launched eSourcing application systems1,2 to enable business collaboration.

The very top and bottom of Figure 1, show the internal levels of the service consumer and -provider respectively where processes are directly enactable by legacy systems caters towards a heterogeneous system environment, e.g., by workflow management systems. Furthermore, processes of the OEM and service providers on a conceptual level are independent from infrastructure and collaboration specifics. In the center of Figure 1, the external level stretches across the respective domains of eSourcing parties where structural process matching takes place and for which eSML is applicable. Either collaborating counterparties project only interfaces, or parts, or all of the respective conceptual-level processes to the external level for performing business-process matching [35], [18]. A contractual consensus between collaborating parties comes into existence when the projected processes are matched externally, i.e., when they are equal. Not projected process parts remain opaque to the collaborating counterparts.

The eSourcing model in Figure 1 shows we use Petri-net formalism for exploring structural properties [35]. The dashed monitoring arcs in Figure 1 connect the conceptual business processes via the external level into a configuration. More recently, the citation [18] demonstrates with BPMN and BPEL the feasibility of this approach with industry standards.

III. Suitability Exploration

There are two angles for approaching the suitability exploration. First, in Section III-A we explore cross-organizational collaboration from the paradigm that contracts are the foundation. In Section III-B, a pattern-based exploration further details the contractual collaboration paradigm.

A. E-Contract-Based Exploration

For ensuring that eSML comprises socio-technical concepts to allow the formulation of real-world business-collaboration in relevant perspectives like control-flow, data-flow, resources and so on, the case-study findings culminating in the eSourcing model of Figure 1 require more exploration. Taking pre-existing work about contract automation [9] into account for the eSML suitability exploration, we pursue a pattern-based exploration method.

The foundation of eSML is the XML-based language ECML (Electronic Contracting Markup Language) [9]. A contract is a legally enforceable agreement, in which

---

two or more parties commit to certain obligations in return for certain rights [37], [15]. Contracts are instruments for organizing business collaborations. Electronic contracting aims at using information technologies to significantly improve the efficiency and effectiveness of paper contracting, allowing companies to support newly emerging business paradigms, while still being legally protected.

Although ECML permits business-process definitions, it lacks a clear collaboration-model support as proposed by eSourcing. Inheriting concepts from ECML, at the highest abstraction, a contract in eSML answers three conceptual questions i.e., the Who, Where, and What question for which we refer the reader to [9] for further details.

1) The Who Concept: This concept legally clearly identifies the contracting parties by including the class party. Parties are actors that have rights and obligations that are listed in the eSourcing configuration. Concerning the relationship cardinalities, it is defined that at least two parties must be stated in a contract. It is also possible to have more than two parties defined. For example, an original manufacturer can agree with several suppliers to be part of one contract.

In a contract, several third parties may optionally be involved that are termed mediators in the case of eSML. Mediators represented by class mediator participate in the enactment of an eSourcing configuration but their rights/obligations are not stated. Consequently, mediators do not have to sign the e-contract. If their relations with the parties must be defined as legally binding, they can become a party in the same or in a separate contract where their rights and obligations are stated. For example, an original manufacturer can agree with several suppliers to be part of one contract.

Contracting parties and optional numbers of mediators are in a relationship with several other classes. The class company_data comprises information like the name of a contracting party or mediator, the type of legal organization, and so on, while the class company_contact_data refers to information related to the geographic location of the eSourcing party. That way, a contracting party or a mediator is uniquely identified according to legal requirements.

The class resource_section is the root class of the resource perspective. A contracting party, or a mediator may have attached resource definitions. However, in most cases it might, for example, be superfluous to define resources of mediators, unless a mediator is also primarily involved in commercial exchanges. Resources may comprise of actors and non-actors where the latter can be of a consumable or non-consumable nature.

The classes company_data and company_contact_data are subclasses of class only_vars_section that contains variables and so-called process snippets. The class var_section is a docking class belonging to the data-flow perspective that includes company description, trade registration number, VAT registration number, address of registration, etc. The class snippet_section references so-called contract snippets that can be attached to particular contract definitions, for example, to attach general terms and conditions.

2) The Where Concept: The Where model of concepts in Figure 2 contains provisions related to the context of the e-contract. We distinguish two basic aspects of the e-contract context, i.e., the business context and the legal context. Thus, two subsections are proposed in the Where section. In addition, a third subsection can be defined to include other e-contract provisions that are not related to the legal and business context.

All three classes named business_context_provisions, legal_context_provisions, and other_context_provisions are subclasses of the grouping class named all_section. It references the classes process_section, var_section, rule_section, and snippet_section.

3) The What Concept: As depicted in Figure 3, the What model contains concepts related to the exchanged values and their related conditions. Two main subsections are distinguished in the What concept, namely the exchanged_value and the corresponding exchange_provisions for the value exchange. These classes are defined separately for every respective contractual party.
involved in contracting.

In a case of product exchange, the product is described by means of data constructs. In a case of service exchange, the service is described through a combination of data that flows through and process constructs. The corresponding financial reward for the received value (in non-barter exchanges) uses the same constructs as a service description subsection. The value exchange provisions subsection requires the use of rule and process specification constructs. Examples for exchange provisions are rules for determining how late payment needs to be handled, how cancellations are dealt with, or definitions for calculating interest adjustments in payments.

B. Pattern-Based Exploration

The chosen method for continued suitability exploration of additional business-collaboration concepts, is as follows. To translate the eSourcing model of Figure 1 into a suitable choreography language, we deduce several feature dimensions in the form of axes that create a multi-dimensional, logical space. On every axis, dimension values detail the eSourcing feature an axis represents. By taking a subset of axes, we create a logical space that represents a particular eSourcing perspective. Consequently, the axes and their contained values serve as a taxonomy for ordering and relating to each other a set of perspective-relevant patterns. Note, we present a high-level overview of the pattern space and refer to [36] for the actual pattern specifications.

The three axes in Figure 4 represent different eSourcing dimensions with values. The created multi-dimensional space is instrumental for deducing eSourcing-construction elements for protecting internal business details, ensuring data exchange that adheres to correct control-flow, and for permitting the service consumer a controlled observation of the service provider’s enactment progress. Correspondingly, the axes of the multi-dimensional space of Figure 4, represent the conceptual dimensions called contractual visibility, conjoinment, and monitorability [36]. The first conceptual dimension permits deducing interaction patterns [34] that occur during the setup phase of an electronic business collaboration. The interaction patterns are input for the proof-of-construction prototype described in Section VI. The latter two conceptual dimensions of Figure 4 turn into eSML language constructs.

The cube dimensions and values of Figure 4, are as follows. Contractual visibility focuses on the amount of business-process nodes a collaborating party projects to an external level to be visible for the counterparty. First, a white-box value means all nodes of a process part to be sourced are externalized. In case of a black-box value, only the interfaces of that process part are projected. Finally, the gray-box value means, the interfaces and a subset of the nodes and arcs of the externally sourced process part are projected.

Conjoinment focuses on the exchange of business information between the domains of the collaborating parties. Consequently, the business processes within the domains contain equal conjoinment constructs. One-directional conjoining implies that there is one out or in-directed information exchange between the domains of a service consumer and provider. Bi-directional conjoining is initiated by an out-directed information exchange to the domain of the collaborating counterpart who returns the communication exchange immediately to the initiating party.
Monitorability covers the way how nodes in the consumer’s and provider’s conceptual-level business processes are linked with each other via constructs of the values termed *messaging* and *polling*. The nodes of the externalized process part are connected to nodes in the corresponding service-provider process. The degree of monitorability of service provisioning for a service consumer increases by the amount of linked nodes. At a minimum, all interface nodes of both domain processes need to be linked with each other. Additional nodes may be linked that belong to the respective business processes of service consumers and service providers. We refer to [36] for detailed pattern specifications and to [34] for collaborating counterparty-interaction patterns during setup.

### IV. Expressiveness Exploration

Semantic clarity for choreography-language constructs ensures that different application systems create uniform enactment behavior. For eSML, we consider control-flow in business collaborations as the best explored perspective based on which the expressiveness may be expanded for other perspectives. We adopt XRL (eXchangable Routing Language) [5] for realizing semantically clear control-flow, which is contained in the schema definition [34] of eSML for the purpose of specifying the contractual spheres of a service consumer and one or many service providers.

XRL is an instance-based workflow language that uses XML for the representation of process definitions and Petri nets [38], [39] for its semantics. The definition of XRL [5] contains as routing elements a catalog of control-flow patterns [3], [4], [24], [25] that result in strong control-flow expressiveness. These routing elements are equipped with Petri-net semantics [6], namely, every routing element stands for an equivalent workflow net (WF-net) [1], [2], [17] that can be connected with other routing elements into a bigger WF-net. Informally, a WF-net is a special Petri-net type with one unique start state and one unique end state and connecting nodes that always lead from the start state to the end state. When the unique output state is reached, all other states are empty. Using WF-nets for XRL-semantics permits the use of theoretical results and standard tools such as Woflan [2], [43] for checking the notion of soundness.

The syntax of XRL is completely specified in a DTD and schema definition [5], [34]. An XRL route is a consistent XML document, that is, a well-formed and valid XML file with top element route (see the Appendix of [34]). The structure of any XML document forms a tree. In case of XRL, the root element of that tree is the route that contains exactly one so-called routing element. A routing element is an important building block of XRL and can either be simple (no child routing elements) or complex (one or more child routing elements). A complex routing element specifies whether, when and in which order the child routing elements are carried out.

### V. eSourcing Markup Language

After explaining the approach for developing eSML through exploring methodologically the suitability in Section III and expressiveness prerequisites in Section IV, we show the high-level structure of the business-collaboration language. As explained earlier, eSML uses parts of the ECML [9] schema as a foundation. Figure 5 reflects this fact by considering an entire eSML instance as a contract between collaborating parties and by structuring the eSML content into the blocks Who, Where, and What, as explained in Section III. The ECML-based parts of eSML in Figure 5, we can not explain here in detail, namely, the definition of company data and company-contact data, the Where block, and the XRL-based process definition within the exchanged-value definition. Instead, we refer to [34] for elaborate code examples and detailed models.

The bold typed eSML-definitions in Figure 5 are extensions and modifications that are not part of the ECML foundation. In the Who block, extensions for eSML are the resource definition and the data definition. In the What block, the XRL adoption permits the use of control-flow patterns for business-process definitions.
that have semantic clarity. Extensions adopt the conjoin-
tment nodes described in Section III-B and for linking to
the resource- and data-definition sections of eSML that
are both based on respective pattern collections [41],
[42]. The lifecycle definitions [34] are for the business
processes and contained tasks.

The lifecycle-mapping block addresses establishing se-
manic equivalence between, firstly, the lifecycles of the
cross-organizationally harmonized business processes,
and secondly, for the lifecycles of tasks from the opposing
domains. Different labels of tasks belonging to processes
of opposing domains may be considered semantically
equal while they actually deviate from each other. To
establish a semantic connection, the second part of the
mapping block focuses on the mapping of task labels
in the active_node_label_mapping tag. Such mapping
is relevant for establishing a contractual consensus be-
tween collaborating parties. The monitorability section
of Figure 5 influences how much of the enactment phase
is perceived by the service consumer. We refer to [34]
for further details of the full eSML schema, models and code examples.

VI. eSML APPLICATION

The contractual-visibility dimension of Figure 4 and
the related pattern specifications [36], permit deducing
interaction patterns [34] between collaborating business
during the setup phase. The pattern exploration
reveals functionality for eSRA (eSourcing Reference Ar-
chitecture) that we refine on three levels and that is
implementable into the required system-application en-
vironment for supporting the establishment a business
collaboration using eSML on the external level. The
developed CrossWork prototype represents a subset of
eSRA and [19], [30] explain case-study details that show
eSML meets the objective of being suitable for the
external formulation and overall enactment of business
collaborations. In the case studies, the external eSML
processes we map to BPEL processes on the conceptual
levels of the collaborating counterparties. Since the suit-
ability and expressiveness of eSML [30] is higher than
for BPEL, this limitation also affects the external-level
specification abilities correspondingly.

To evaluate the expressiveness of eSML, the control-
flow specification realizes the WF-net semantics of XRL
by mapping to PNML [26], [27], [44], an XML-based
interchange format that permits the definition of Petri-
net types. A style-sheet translator contains mapping

Due to page limitation, in Figure 6, we can only
explain the lifecycle of a business process as it is car-
ried out by the enactment application XRL/flower [33]
that is adoptable for an eSRA-based implementation.
Woflan [2], [43] for checking control-flow soundness, is
part of XRL/flower. Note that new control-flow elements
adopted in XRL merely require an additional mapping
rule in the stylesheet translator while enactment engine
remains unchanged. We refer to [34] for further details.

VII. RELATED WORK

Contracting is part of Web-service choreography in
some research work that only takes a technical position.
In [10], contracts are descriptions of the observable
behaviour of multiple services to tackle the problem
of composition. Moreover, the authors show that a
compliant group of contracts is still compliant after
replacement by one of its subcontract. In [11], the same
authors relate the theory of contracts with the notion
of choreography conformance, used to check whether an
aggregation of services correctly behaves according to a
high level specification of their possible conversations.
Projection and contract refinement achieve composition
of choreography.

The authors in [40] recognize support for automated
service contracting and enactment is crucial for any large
scale service environment, where large numbers of clients
and service providers interact. Concurrent Transaction
Logic is instrumental to model and reason about service
contracts to allow iterative processes in the specification
of service contracts and enable reasoning about such
contracts.

In [12], the authors consider contracts to be finite
labelled transition system and study the foundational
aspects of contract compliance in a language indepen-
dent way. The language independent representation of
contracts allows for choreography projection in struc-
tured operational semantics. The evaluation applies the
theory of contract compliance with industry-standard
choreography specifications such as WS-CDL.

The related work has in common that no socio-
technical approach for the concept of contracting in
service choreography recognizes the interaction between

![Figure 6. Lifecycle of eSML-business-process instantiation and enactment.](image-url)
people and technology in cross-organizational collaborations. We address this lack by choosing a legal notion of a contract that states a consensus between collaborating parties must be present. This consensus in eSML represents the matching process views of service consumer and service provider, which is fundamentally different to the listed related work of purely technical focus.

VIII. Conclusions

This paper presents eSML, a choreography language for cross-organizational business collaboration. eSML results from a sociotechnical methodical, case study-based suitability and expressiveness exploration that ensures the language comprises essential collaboration concepts with a foundation for semantic clarity.

As eSML adopts a real-life contracting foundation, collaborating parties use process views they project externally for cross-organizational matching. Once a matching occurs, a consensus exists that is the essential criteria for establishing a contract. The process views are subsets of larger business processes inside the domains of collaborating parties. We use a subset of ECML as a base language for eSML that we enhance with additional schemas for process-view matching, and cross-organizational conjunction and monitorability for achieving suitability. Furthermore, assuming that the control-flow perspective in a business collaboration is best explored, the expressiveness in eSML we address by adopting WF-net based semantics that is verifiable with tool support. Employing setup-interaction patterns of collaborating parties, we evaluated eSML in a proof-of-construction prototype for the setup and enactment of business collaborations in the CrossWork research project.

For future work, we plan to carry out further case studies with eSML in research projects about designing cloud-computing infrastructures for mobile business collaboration. In those studies, we want to address expressiveness extensions into more business-collaboration perspectives than control-flow. Furthermore, an open research issue is the safeguarding of business collaborations with transactionality concepts that need to go further than traditional transactions from the database- and workflow domains. Thus, future work will explore such electronic business transactions with the objective of understanding how to extend eSML for ensuring a safeguarding of electronic business collaborations.

Acknowledgment

This paper was supported in part by CNSF grant 61363007 and by HNU grant KYQD1242 and HDSF201310; the Dawn Program of Shanghai Education Commission (11SG44); the Research Fund for the Doctoral Program of Higher Education of China (20123120130001).

References


