Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations

Conference Paper · August 2015

DOI: 10.1007/978-3-319-21915-8_1

1 author:

Alex Norta
Tallinn University of Technology

Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations

Alex Norta

Department of Informatics, Tallinn University of Technology,
Akadeemia tee 15A, 12816, Tallinn, Estonia
alex.norta.phd@ieee.org

Abstract. Electronic communities of decentralized autonomous organizations (DAO) that engage in agile business-network collaborations, are enabled by recent blockchain-technology related innovations using smart contracting. DAOs utilize service-oriented cloud computing in a loosely-coupled collaboration lifecycle that commences with the setup phase. The latter supports the selection of services provided and used by DAOs in combination with smart contract negotiations. Such setup phases for DAO-communities use blueprints of business-network models that DAOs populate with tentative service offers. The negotiation phase may result either in a consensual agreement, a counteroffer, or a disagreement. In the latter case, the smart contract negotiation collapses and the lifecycle returns to the beginning of the selected collaboration blueprints. To the best of our knowledge, such a smart-contracting setup lifecycle has not been formalized so far. The paper fills the gap and evaluates the model with means of model-checking methods.

Key words: decentralized autonomous organization, smart contract, open cloud ecosystem, Governance-as-a-Service

1 Introduction

A trend emerges for so-called decentralized autonomous organizations (DAO)[3] to engage in the formation of electronic communities that smart contracts [33] cast together. A smart contract is a computerized transaction protocol [32] to execute contract terms. Consequently, for achieving non-repudiation and fact-tracking of a consensual smart-contract agreement, blockchain technology [18, 28] is suitable. The blockchain is a distributed database for independently verifying the chain of ownership of artefacts [27] in hash values that result from cryptographic digests. As a further means to realize electronic communities of DAOs, the emergence of service-oriented cloud computing (SOCC) [35] promises for companies an acceleration of seamless, ad-hoc integration and coordination of information- and business-process flows [10] to orchestrate and choreograph [22] heterogeneous legacy-system infrastructures.

While research results emerge for cross-organizational business collaboration, a gap exists with respect to a formalized exploration of DAO-collaboration setup lifecycles. This paper fills the gap by investigating the research question how to set up in a dependable way electronic communities of business collaborating DAOs to the point of a consensually agreed upon smart contract? Here, dependable [2] means the components that are part of the setup lifecycle are relied upon to perform exclusively and correctly the system task(s) under defined operational and environmental conditions over a defined period of time. Based on this main research question, we deduce the following sub-questions to establish a separation of concerns. What is the top-level setup-lifecycle and which business data flows along it? When exceptional scenarios occur, what mechanisms exist for an orderly compensation-rollback and partial-, or complete lifecycle termination? What are the relevant system properties for successfully realizing the startup-lifecycle with transactionality platform as Governance-as-a-Service (GaaS) in a Cloud?

The remainder of the paper is structured as follows. Section 2 provides additional information relevant for understanding the business-collaboration context. Section 3 shows the top-level of the formalized startup-lifecycle in which service protocols are visible with their data-exchanges. Furthermore, in Section 4 we show the successful rollbacks of smart-contracting semantics within a startup-lifecycle. Section 5 lists the results from model checking that are aiding the application-system implementation of a sound startup-lifecycle. Section 6 discusses related work and finally, Section 7 concludes this manuscript by summarizing the research work, giving the contributions achieved and showing directions for future work.

2 Conceptual Collaboration Context

For comprehending the setup-lifecycle in the sequel, the following frameworks are important to comprehend. We explain a peer-to-peer (P2P) collaboration model for DAOs in Section 2.1. Furthermore, as contracts are the foundation of business collaboration, we also show in Section 2.2 concepts and properties for smart contracting.

2.1 P2P-collaboration model

Pertaining to DAO-collaboration, Figure 1(a) conceptually depicts a configuration. The blueprint for an electronic-community formation is a so-called business-network model (BNM) [29]. The latter captures choreographies that are relevant for a business scenario and it contains legally valid template contracts that are service types with affiliated organizational roles. The BNMs are available in a collaboration hub that houses business processes as a service (BPaaS-HUB) [23] in the form of subset process views [10]. The latter enable a fast and semi-automatic discovery of collaboration parties for learning about their identity, services, and reputation.

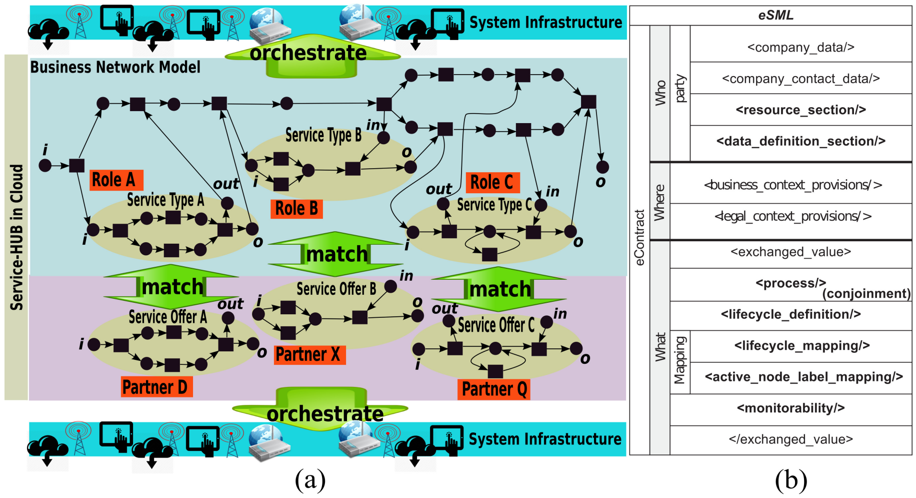

Fig. 1. P2P-collaboration using the eSourcing framework.

On the external layer of Figure 1(a), service offers identically match with service types contained in the BNM with the contractual sphere of collaborating parties. Additionally, a collaborating partner must match into the specific partner roles associated with a respective service type. We refer the reader to [10] for details about the the tree-based process-view matching to establish a DAO-configuration into a contract-based collaborations.

2.2 Smart contract

The top-level structure shows a smart contracting language termed eSourcing Markup Language (eSML) [25] in Figure 1(b). The bold typed definitions in the eSML-structure are extensions and modifications that are not part of the Electronic Contracting Markup Language (ECML) [1] foundation.

The core structure of a smart contract we organize according to the interrogatives *Who* for defining the contracting parties together with their resources and data definitions, *Where* to specify the business- and legal context, and *What* for specifying the exchanged business values. For achieving a consensus, we assume the *What*-interrogative employs matching process views that require cross-organizational alignment for monitorability. We refer to [25] for more information about the smart-contracting ontology.

Next, we discuss first the DAO-lifecycles for the top-level collaboration-setup stages. Note, from here on we use eCommunity for a community of collaborating DAOs and eContract for an electronic contract.

3 Top-Level Setup-Lifecycle

First, Section 3.1 shows the formalized top-level of the setup lifecycle[1] using Coloured Petri Nets (CPN) [14]. The latter is a graphical oriented language for the design, specification, simulation and verification of systems such as communication protocols, distributed systems, automated production systems. Informally, the CPN-notation comprises states, denoted as circles, transitions, denoted as rectangles, arcs that connect states and transitions but never states with other states or transitions with other transitions, and tokens with color, i.e., attributes with values. Arcs carry inscriptions in CPN-ML expressions that evaluate to a multiset or a single element. We use CPNtools[2] for designing, evaluating and verifying the models in this paper. Modules in CPN are non-atomic place-holder nodes for hierarchic refinements that correspond to respective services in a system-implementation. Furthermore, Section 3.2 describes the token colours that all CPN-models of this paper use.

3.1 Formalized setup top-level

The lifecycle of Figure 2 shows that a nested module labelled *BNM selection* is an ecosystem for breeding service types that become part of so-called business-network models (BNM). For establishing such an eContracting preliminary, the so-called eSourcing Markup Language eSML [25] is a candidate for BNM-specifications. The latter is a cross-enterprise collaboration blueprint to insert service types and roles for the next step in the lifecycle, namely the population with service offers and eCommunity-partners. The BNMs that emerge from the breeding ecosystem exist permanently for repeated use in the subsequent populating stage, i.e., conformance-validated service offers and BNMs. The *populate*-module validates the inserted service offers against the service types of the BNM as it emerges from the breeding ecosystem.

Such a breeding ecosystem in the form of service brokers citation [11] investigates in a state-of-the-art comparison since this is identified as a key concern for future cloud technology development and research. In [6, 7], the authors introduce service value brokerage between service consumers and service providers that crosscuts business modeling, knowledge management and economic analysis. The research in [24] describes a lifecycle of finding and matching service requests with service offers for collaboration enactment in a Business-Process-as-a-Service (BPaaS) HUB.

At the end of the *populate*-phase in Figure 2, a proto-contract exists for a *negotiate* step that is carried out by the eCommunity-partners. The negotiation of the proto-contract has three different outcome options. An agreement of all eCommunity-partners establishes the eContract for subsequent rollout and enactment; a counter-offer from only one eCommunity-partner triggers a business-semantics rollback to the inception of the *negotiate*-module; finally, a disagreement of only one eCommunity-partner results in a complete termination of not

[1] Full CPN-model: http://tinyurl.com/ofae8gn
[2] http://cpntools.org/

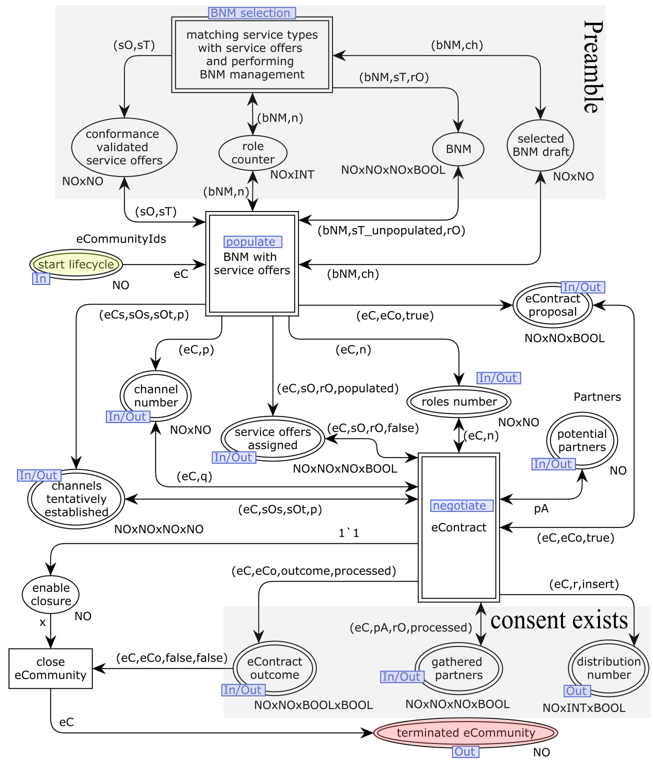

Fig. 2. Top-level formalized setup-lifeycle for electronic community establishment.

only the contract negotiation but additionally, the startup-lifecycle also suddenly terminates with the identification ending in the state labelled *terminated eCommunity*.

According to [16], agent-based negotiation is progressing fast and enables semi- to fully automated negotiation. In [5], an agent-based coordinated-negotiation architecture ensures adherence to service-level agreements and the stateful coordination of complex cross-organizational services.

3.2 Related business-collaboration data

In the left column of Table 1 showing 1 is the top-level and 4 the lowest refinement of the setp-lifecycle components. Data-flow properties listed for availability in a certain refinement-hierarchy level are present for all lower- but not for any higher hierarchy levels. The fourth column of Table 1 textually explains the purpose of a data-flow property while the types are either integer, string or boolean. In the first case, the integer mostly represents a token identification number, a string is either an eContract-negotiation outcome or an eContract-proposal extracted from a business-network model and boolean represent decision points. The data-flow properties of Table 1 either represent in concrete implementations more elaborate database tables, or XML-schemata with more refining properties.

level	CPN module	data property	description	type
1	eCommunity lifecycle	sO	service offer that fits a service type	integer
		sOs	service offer source for communication channel establishment	
		sOt	service offer target for communication channel establishment	
		pA	partner of an eCommunity	
		rO	role a partner can fill	
		eC	eCommunity identification	
		eCo	eContract based on which partners of an eCommunity transact	
		n,r,k,p,l,q,s	counter variables	
		assigned	service offer assigned to a service type	boolean
		processed	partner prepared for eContract counteroffer re-distribution	
		decision	for negotiated contract proposal (agree|disagree|counter)	string
		outcome	like decision, but input for eCommunity continuation or termination	
2	create	bNM	business network model that get populated with service types and roles	integer
		m	counter variable	
		sT	service type that populates a bNM	
3	populate	ch	channel of communication between services	integer
4	interoperability checking	rOt	role source for communication channel establishment	integer
		rOs	role target for communication channel establishment	
4	contract extraction	spec	specification of extracted eContract	string
4	agreement finalizing	result	whether all eCommunity partners agree on an eContract proposal or not	boolean
		distributed	contract distributed to partner	
4	disagreeing	z	counter variable	integer
		eCo_new	new eContract from a counteroffer to be negotiated	

Table 1. Concepts and properties used during the setup phase of an electronic community of DAO (full details in [21]).

Next, we show what rollback mechanism are in place on a refined setup-lifecycle level for flexibly responding to exceptional situations.

4 Compensation Rollbacks

To study the rollback types in detail, the following CPN-models are all refinements of Figure 2 and we commence with Section 4.1 for discussing the establishment of an eContract-preamble. Next, Section 4.2 shows rollbacks that occur during the BNM-population phase and finally, Section 4.3 comprises detailed rollbacks furing the negotiation-lifecycle.

4.1 BNM selection

This module functions as an ecosystem to breed BNMs. For that, a repository exists in the *BNM selection* for which we assume users insert service types over time that they specify themselves. The same assumption holds for the repository of service offers in *BNM selection*that is correspondingly a state in the CPN-module labelled *repository service offers*.

To be considered as a service offer for populating a BNM, beforehand the passing of a conformance validation is necessary. The actual BNM-selection involves choosing a BNM-draft for adding validated service offers and roles to be filled subsequently with respective eCommunity-partners.

Research in [17] focuses on finding service providers that minimizes the total execution time of a business process based on cost and execution time constraints. Work in [8] gives a Web service selection algorithm that satisfies not only user preferences and functional requirements but also takes into account transactional properties and service-level-agreement characteristics.

4.2 The BNM-population

In this service depicted in Figure 3, an unique eCommunity number identifies the entire transaction lifecycle. The service types in the chosen BNM draft specification are populated with conformance-validated service offers. Next, an interoperability check of those service offers ensures the channels are capable of exchanging data that matches semantic expectations. When the proto-contract with tentatively established channels is ready, the population stage is ready for the next BNM population.

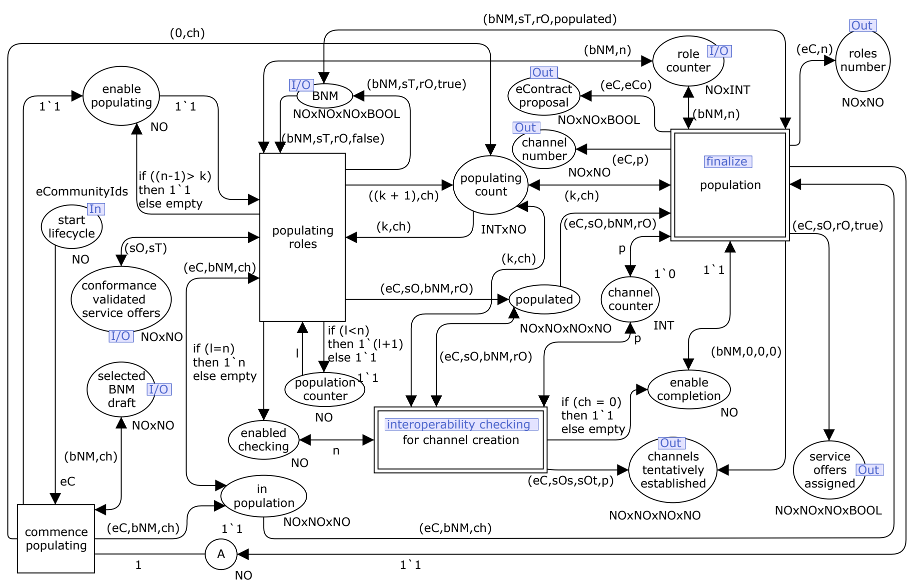

Fig. 3. Populating a BNM with service offer, roles, and checking interoperability (module *populate* in Figure 2).

The service offers chosen for populating a BNM are in the state labelled *populated*. For the tentatively established channels between service offers, a dedicated token color contained in the BNM indicates the required amount of channels. Consequently, a chosen service offer becomes a channel source and another one a channel target that must adhere to given channel requirements and data-semantics matching. The interoperability checking ends when the number of required tentative channels existsthat enables the completion stage of BNM-population.

Populating the BNM with roles touches on many open issues in the state of the art [4]. The open issues pertain to hiding business details in the process that, e.g., constitute a competitive advantage; the lacking expressiveness of existing resource-assignment languages to meaningfully automate role population; remaining in control over the allocation of outsourced activities; different vocabulary in the respective domains of the parties involved; and so on. For interoperability checking in Figure 3, citation [36] gives many aspects. When checking the interoperability abilities within a populated BNM, the categories involved [9] comprise collaboration knowledge, models and standards, business processes, services, information and data.

To complete the population of BNMs, the *finalize*-module empties the *populate*-module of remaining tokens. In an application-system implementation, this emptying corresponds to emptying an instantiation of not needed database entries and abandoned computing processes that lead to a collapse of the overall system. Such a collapse and restart is undesirable if many eCommunities are active to transact at the same time.

Finalizing the BNM-population commences with an enabling of completion, continues with a reset and extracts important values for the subsequent proto-contract negotiation. The extracted values are the amount of roles, the proto-contract with tentative channels, a channel counter and assigned service offers. Eventually, the *finalize*-module concludes the BNM-populating stage.

4.3 The eContract negotiation

Once the *negotiate*-module enables from a completed BNM population, an eContract-proposal is extracted from that input from the *populate*-module. All eCommunity-partners receive an eContract copy who vote with three possible options. The *negotiate*-module we split into two depictions, Figure 4 for extracting a proto-contract and Figure 5 for consensually establishing an eContract.

The first part of negotiation in Figure 4, comprises the extraction of a proto-contract. After choosing an eCommunity for a prepared business-network model (BNM), a pool of potential partners serves to fill the contained roles with concrete partners. Next, in the *contract extraction* module, the eContract-proposal emerges from the *population*-module, extending specifications of the tentatively created service-offer channels. The created eContract-proposal is ready for the subsequent steps of negotiation.

The *negotiate*-module in Figure 5, caters in an ideal case for an *agree* of all eCommunity partnersthat establishes a consensus to make an eContract come

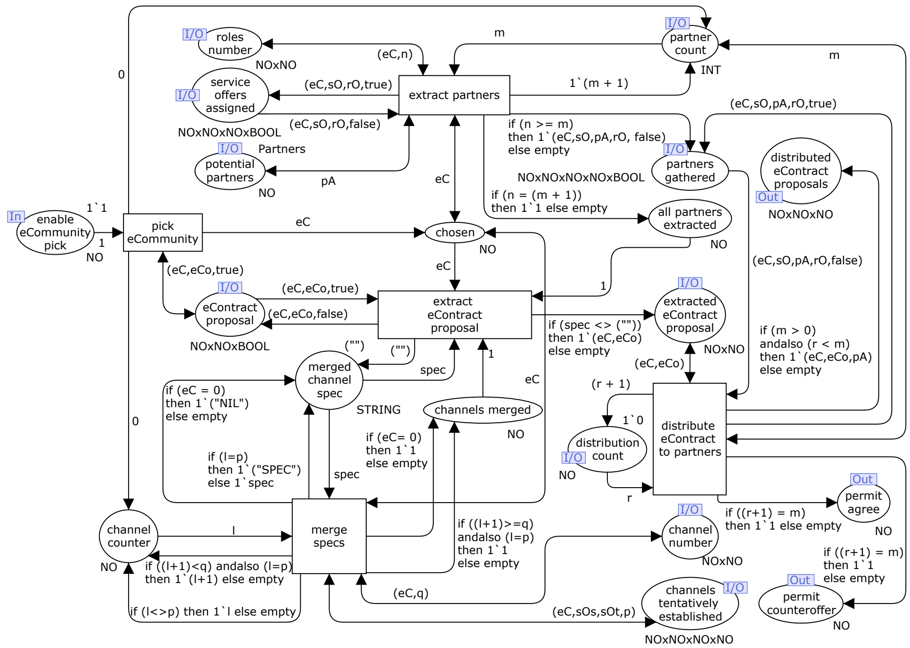

Fig. 4. Extracting a proto-contract (in module *negotiate* of Figure 2).

into existence. Secondly, one eCommunity-partner decides on a modification and proposes a *make counteroffer* for an eContract. The counteroffer instantaneously disables all voting options, removes the already casted votes and redistributes a copy of the modified contract for every respective eCommunity-partner. Thirdly, in the case of only one *disagree*, the voting process halts, again, it triggers the removal of already casted votes and terminates the entire CEC-lifecycle. Subsequently, the eCommunity identification comes to rest in the state labelled *terminated eCommunity*.

When all eCommunity-partners vote on an eContract-proposal, the objective is to achieve a consensus by all agreeing. In that case, the nested module *agreement finalizing* empties the *negotiation* of eCommunity-partner instances and the offered services are reset for the next negotiation phase. Finally, the negotiation outcome is set to *true*. For both cases of an eContract-proposal disagreement and also for the issuance of a counteroffer by an eCommunity-partner, the calculated number of eCommunity-partners serves as input to empty the *negotiate*-service.

A disagreement leads to the removal of all eCommunity-related tokens, the lifecycle termination and the identification token moves to its final state labelled *terminated eCommunity*. However, with a new counteroffer, the existing eCommunity-partners must cast votes again and as such, each partner must receive a contract-proposal copy of the counteroffer. All other eCommunity-related tokens remain in their respective states.

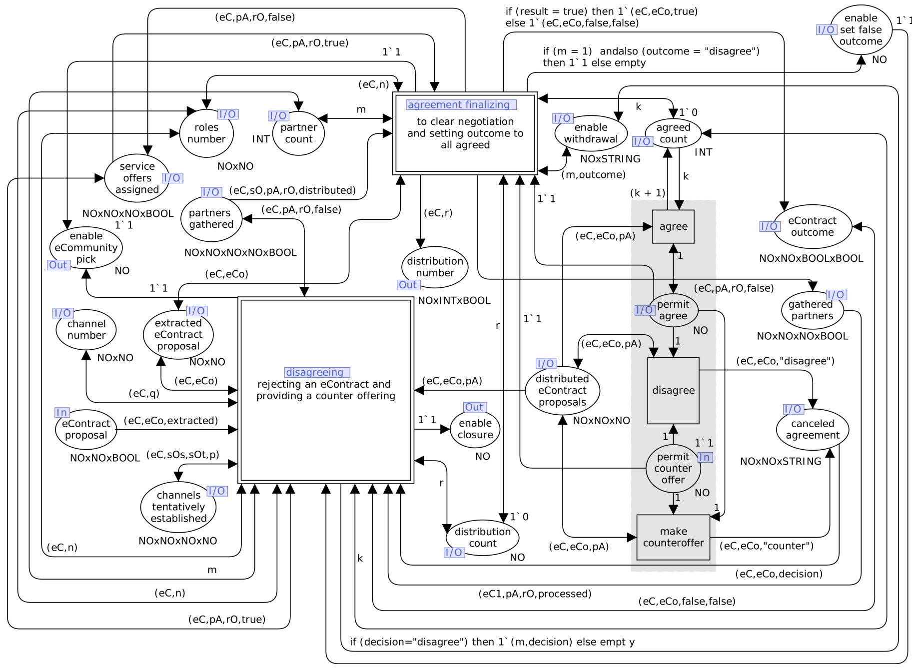

Fig. 5. Deciding on an eContract to complete a negotiation (in module *negotiate* of Figure 2).

When an eCommunity-partner casts a vote in a negotiation phase indicating a disagreement, this module performs the final clearance preparation for the subsequent enactmentthat is *disagreement finalizing*. Setting the negotiation outcome to $false$ starts the removal of the eCommunity-partners, followed by a removal of service offers and channelsthat completes the disagreement finalization.

For eContract negotiation, the citation [15] defines a knowledge-based negotiation procedure where specific separate protocols and strategies enable agent participation in interaction processes between collaborating parties. In [30], contributions are a discussion of core elements in an eContract negotiation such as role, properties of electronic services and contract models; the definition of the actual negotiation process; and a conceptual model to support the negotiation of web service. In [34], the authors give features of an eContract metamodel they connect with quality-of-service (QoS) for being taken into account during a negotiation and renegotiation process a feature-based toolkit supports as a proof-of-concept prototype. Finally, the contribution [19] propose a lifecycle to support the management of eContracts comprising proposal, configuration, publication, negotiation, operation, and closure. A broker enables automatic configuration and deployment of web services.

Next, we perform a formal- and also proof-of-feasibility evaluation of the startup lifecycle.

5 Model Properties

We employ CPN Tools [14] for correctness and performance checking, especially on aspects relevant for system developers: reachability of CPN-modules end states in manual, or fully automated simulation token games as state explosion means full computational verification is challenging for this size of models; detection of loops as a potential source of livelocks that prevent desired termination reachability; loops require specific attention with respect to effectiveness of exit conditions, such as elements of business-level policy control; performance peaks during runtime either for the design of sufficient resources or for restricting the load with business-level policy control; full system utilisation for ensuring that each part of the modelled system actually is used in some scenario; and consistent termination, i.e., consistent home markings that ensure simple testing of a real system.

		module property					
	module	loops	performance peaks	liveness	Utilization	home marking	dead mark-ing
	BNM selection	no	evenly balanced	ND/NL	yes	no	multiple
	populate	no	populating roles, interoperability checking	ND/NL	yes	no	no
negotiate	negotiate with forced agreement	yes	contract extraction, agreement finalizing	D*/NL	yes*	no	multiple
negotiate	negotiate with forced counteroffer	yes	contract extraction, distribute eContract to partners	D*/NL	yes*	no	no
negotiate	negotiate with forced disagree	yes	contract extraction, agreement finalizing	D*/NL	yes*	no	multiple

Table 2. Model-checking results for the setup-lifecycle.

The model-checking results in Table 2 focus on CPN-modules where the generated state-space is computationally feasible. For the *negotiate*-module there are three outcomes. Either all collaboration partners agree on a contract proposal, or a counteroffer negotiation unfolds, or one partner disagrees and collapses the proposal.

Loops exist in all three outcomes of the *negotiate*-module. The contract negotiation loop is self-restricting as it only processes the collaboration partners respectively. However, finalising the negotiation comprises loops because of a counteroffer issuance. The test results for remaining modules in Table 2 show they do not contain loops.

Performance peaks in Table 2 represent places in the startup-lifecycle that are potential performance bottlenecks. Peaks exist in all but the module *BNM selection*. For the *populate*-module, peaks occur for populating roles and also for checking if channel requirements and data-semantics match. Pertaining to the latter modules we refer the reader to [21] for details. For all three *negotiate*-run cases, peaks exist for counteroffers and negotiation finalisation. All these elements are effectively dependent on the amount of populations and negotiations performed, and therefore, stay limited.

While no module listed in Table 2 have any home marking, the model-checking results for dead markings differ. Multiple dead markings and no home markings means for practitioners the testing of implementations is more demanding as many test cases are required. $D*$ means the model-checking results show the dead markings result from intentional disabling of marking paths for the purpose of focusing in specific marking paths under investigation. The latter means for practitioners the testing of implementations is more demanding as many test cases are required. Finally, pertaining to utilisation tests, Table 2 shows no unused subsets exist in the models. We refer to [21] for full details about the model-checking results. Pertaining to utilisation tests, Table 2 shows no unused subsets exist in the models. Entries yes* state a subset of the CPN-module is intentionally disabled as an available runtime path to manage state-space explosion issues, or for allowing an analysis to focus on specific enactment paths.

Finally, due to page limitations, we refer the reader to [25] for details about the feasibility of an application-system implementation.

6 Related Work

As noted earlier, it is necessary to cross-organizationally establish collaboration frameworks in a way that does not force companies into disclosing an undesirable amount of business internals [10]. Different research efforts address this issue. In [13], the authors investigate tool support for cross-organizational collaboration design. Similarly in [31], research results present an integrated specification language and a user interface for collaborating government organizations to specify events of common interest, policies, constraints and regulations in the form of different types of knowledge rules, manual and automated services, and sharable workflow processes.

In [20], a framework facilitates the understanding of major cross-organizational collaboration challenges. For example, supporting process-level collaboration, and protection of shared IP and data with various enterprise-level and regulatory policies, including flexible and policy-aware process collaboration among people from different enterprises. Also [26] points out the need for agility of business operations in a collaborative services ecosystem of partners and providers by proposing Work-as-a-Service that a collaboration hub facilitates. A cross-organizational architecture in [12] specifies the features and their composition at a higher level that abstracts the internal implementation mechanisms of the organizations involved.

7 Conclusion

We investigate the setup-lifecycle of cross-organizational business-process aware collaboration for decentralised autonomous organizations. During the collaboration setup, the parties must consent to establish a smart contract. The setup-lifecycle commences with equipping a business network model with service types and assigned partner roles. Candidate partners with their service offers populate the service types and a negotiation follows that results either in a dissent, counter-offer issuance, or a consent. We model-check the setup lifecycle to reveal formal properties that have practical implications for application-system implementations.

For exploring the setup-lifecycle in a dependable way, we choose CPN Tools that has a graphical modelling notation backed with formal semantics. Exceptional scenarios that may occur are mismatches during the service-endpoint creation during the population stage. Additionally, exceptions may occur in the negotiation state of a smart contract when either parties disagree, or when a counteroffer issuance occurs. In the first case a loop back to the population stage occurs while a counteroffer issuance remains contained within the negotiation loop. The relevant model properties we check for are loops, performance peaks, liveness, utilization, home markings and whether dead markings exist in the setup-lifecycle.

For future work, we plan to apply the setup-lifecycle in projects for cyber-physical system governance in the realm if Internet-of-Things. Consequently, we plan to explore how blockchain-technology can realize non-repudiation in process-aware collaboration missions. Additionally, we plan to investigate how blockchain technology can enable novel approaches for effective management of trust, reputation, privacy and security in cross-organizational cyber-physical system collaboration scenarios.

Acknowledgments

This work was funded by the Estonian IT Akadeemia and also partly the research project SF0140013s10 "Model-based Creation and Management of Evolutionary Information Systems" by the Estonian Ministry of Education and Research.

References

1. S. Angelov. *Foundations of B2B Electronic Contracting.* Dissertation, Technology University Eindhoven, Faculty of Technology Management, Information Systems Department, 2006.
2. A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):11–33, 2004.
3. V. Butterin. A next-generation smart contract and decentralized application platform, 2014.
4. C. Cabanillas, A. Norta, M. Resinas, J. Mendling, and A. Ruiz-Cortes. Towards process-aware cross-organizational human resource management. In I. Bider, K. Gaaloul, J. Krogstie, S. Nurcan, H.A. Proper, R. Schmidt, and P. Soffer, editors, *Enterprise, Business-Process and Information Systems Modeling*, volume 175 of *Lecture Notes in Business Information Processing*, pages 79–93. Springer Berlin Heidelberg, 2014.
5. M.B. Chhetri, J. Lin, S.K. Goh, Jun Yan, Jian Ying Zhang, and R. Kowalczyk. A coordinated architecture for the agent-based service level agreement negotiation of

web service composition. In *Software Engineering Conference, 2006. Australian*, pages 10 pp.–, April 2006.
6. Y. Duan, K. Huang, A. Kattepur, and W. Du. Towards value-driven business modelling based on service brokerage. In A.R. Lomuscio, S. Nepal, F. Patrizi, B. Benatallah, and I. Brandic, editors, *Service-Oriented Computing âĂŞ ICSOC 2013 Workshops*, volume 8377 of *Lecture Notes in Computer Science*, pages 163–176. Springer International Publishing, 2014.
7. Y. Duan, Y. Wang, J. Wei, A. Kattepur, and W. Du. Value-added modelling and analysis in service value brokerage. In A.R. Lomuscio, S. Nepal, F. Patrizi, B. Benatallah, and I. BrandiÄĞ, editors, *Service-Oriented Computing âĂŞ ICSOC 2013 Workshops*, volume 8377 of *Lecture Notes in Computer Science*, pages 209–222. Springer International Publishing, 2014.
8. J. El Hadad, M. Manouvrier, and M. Rukoz. Tqos: Transactional and qos-aware selection algorithm for automatic web service composition. *Services Computing, IEEE Transactions on*, 3(1):73–85, Jan 2010.
9. B. Elvesæter, F. Taglino, E. Del Grosso, G.B. Elguezabal, and A. Capellini. Towards enterprise interoperability service utilities. In *EDOCW*, pages 224–229, 2008.
10. R. Eshuis, A. Norta, O. Kopp, and E. Pitkanen. Service outsourcing with process views. *IEEE Transactions on Services Computing*, 99(PrePrints):1, 2013.
11. F. Fowley, C. Pahl, and L. Zhang. A comparison framework and review of service brokerage solutions for cloud architectures. In A.R. Lomuscio, S. Nepal, F. Patrizi, B. Benatallah, and I. BrandiÄĞ, editors, *Service-Oriented Computing âĂŞ ICSOC 2013 Workshops*, volume 8377 of *Lecture Notes in Computer Science*, pages 137–149. Springer International Publishing, 2014.
12. C. Hahn, J. Recker, and J. Mendling. An exploratory study of it-enabled collaborative process modeling. In M. zur Muehlen and J. Su, editors, *Business Process Management Workshops*, volume 66 of *Lecture Notes in Business Information Processing*, pages 61–72. Springer Berlin Heidelberg, 2011.
13. Christopher Hahn, Jan Recker, and Jan Mendling. An exploratory study of it-enabled collaborative process modeling. In M. zur Muehlen and J. Su, editors, *Business Process Management Workshops*, volume 66 of *Lecture Notes in Business Information Processing*, pages 61–72. Springer Berlin Heidelberg, 2011.
14. Kurt Jensen, Lars Michael, Kristensen Lisa Wells, K. Jensen, and L. M. Kristensen. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. In *International Journal on Software Tools for Technology Transfer*, page 2007, 2007.
15. K. Kravari, C. Papavasileiou, and N. Bassiliades. Knowledge-based e-contract negotiation among agents using semantic web technologies. In C. Badica, N.T. Nguyen, and M. Brezovan, editors, *Computational Collective Intelligence. Technologies and Applications*, volume 8083 of *Lecture Notes in Computer Science*, pages 215–224. Springer Berlin Heidelberg, 2013.
16. R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *Commun. ACM*, 53(1):78–88, January 2010.
17. D.A. Menasce, E. Casalicchio, and V. Dubey. On optimal service selection in service oriented architectures. *Performance Evaluation*, 67(8):659 – 675, 2010. Special Issue on Software and Performance.
18. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.
19. J. B Neto and C.M. Hirata. Lifecycle for management of e-contracts based on web service. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, 2013.

20. H.R.M. Nezhad, C. Bartolini, J. Erbes, and S. Graupner. A process- and policy-aware cross enterprise collaboration framework for multisourced services. In *SRII Global Conference (SRII), 2012 Annual*, pages 488–493, July 2012.
21. A. Norta. *Safeguarding Trusted eBusiness Transactions of Lifecycles for Cross-Enterprise Collaboration*. http://www.cs.helsinki.fi/u/anorta/publications/TBT_SOCC.pdf, 2012.
22. A. Norta, P. Grefen, and N.C Narendra. A reference architecture for managing dynamic inter-organizational business processes. *Data & Knowledge Engineering*, 91(0):52 – 89, 2014.
23. A. Norta and L. Kutvonen. A cloud hub for brokering business processes as a service: A "rendezvous" platform that supports semi-automated background checked partner discovery for cross-enterprise collaboration. In *SRII Global Conference (SRII), 2012 Annual*, pages 293–302, July 2012.
24. A. Norta and L. Kutvonen. A cloud hub for brokering business processes as a service: A "rendezvous" platform that supports semi-automated background checked partner discovery for cross-enterprise collaboration. *Annual SRII Global Conference*, 0:293–302, 2012.
25. A. Norta, L. Ma, Y. Duan, A. Rull, M. Kõlvart, and K. Taveter. eContractual choreography-language properties towards cross-organizational business collaboration. *Journal of Internet Services and Applications*, 6(1):1–23, 2015.
26. D. Oppenheim, S. Bagheri, K. Ratakonda, and Y.M. Chee. Agility of enterprise operations across distributed organizations: A model of cross enterprise collaboration. In *SRII Global Conference (SRII), 2011 Annual*, pages 154–162, March 2011.
27. B.S. Panikkar, S. Nair, P. Brody, and V. Pureswaran. Adept: An iot practitioner perspective, 2014.
28. T. Patron. *The Bitcoin Revolution: An Internet of Money*. Travis Patron.
29. T. Ruokolainen, S. Ruohomaa, and L. Kutvonen. Solving service ecosystem governance. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2011 15th IEEE International*, pages 18–25. IEEE, 2011.
30. G.C. Silva, I.M. de Souza Gimenes, M. Fantinato, and M.B.F. de Toledo. Inter-organizational negotiation of web-services. *International Journal of U-& E-Service, Science & Technology*, 6(5), 2013.
31. S.Y.W. Su, Xuelian Xiao, J. DePree, H.W. Beck, C. Thomas, A. Coggeshall, and R. Bostock. Interoperation of organizational data, rules, processes and services for achieving inter-organizational coordination and collaboration. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–10, Jan 2011.
32. M. Swan. Blockchain thinking: The brain as a dac (decentralized autonomous organization). In *Texas Bitcoin Conference*, pages 27–29, 2015.
33. N. Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.
34. D.A. Vecchiato, M.B.F. Toledo, M. Fantinato, and I.M.S. Gimenes. A feature-based toolkit for electronic contract negotiation and renegotiation. In *Proceedings of the IADIS International Conference WWW/Internet*, volume 2010, pages 3–10, 2010.
35. Y. Wei and M.B. Blake. Service-oriented computing and cloud computing: Challenges and opportunities. *Internet Computing, IEEE*, 14(6):72–75, 2010.
36. Martin Zelm, Raquel Sanchis, Raul Poler, and Guy Doumeingts. *Enterprise Interoperability: I-ESA'12 Proceedings*. John Wiley & Sons, 2012.