ARTICLE IN PRESS

Information Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Shared Ledger Accounting — Implementing the Economic Exchange pattern

Hans Weigand [a,*], Ivars Blums [b], Joost de Kruijff [a]

[a] Tilburg University, P.O. Box 90153 5000 LE Tilburg, The Netherlands
[b] SIA ODO, Draudzības iela 11-2, Ogre, LV-5001, Latvia

ARTICLE INFO

Article history:
Received 30 January 2019
Received in revised form 31 May 2019
Accepted 10 September 2019
Available online xxxx

Keywords:
Accounting ontology
Smart contracts
UFO ontology
Blockchain

ABSTRACT

Distributed Ledger Technology (DLT) suggests a new way to implement Accounting Information Systems, but an ontologically sound consensus-based design is missing to date. Against this research gap, the paper introduces a DLT-based shared ledger solution in a formal way and compliant with Financial Reporting Standards. We build on the COFRIS accounting ontology (grounded on UFO) and the blockchain ontology developed by De Kruijff & Weigand that distinguishes between a Datalogical level, an Infological and an Essential (conceptual) level. It is shown how both consensual and enterprise-specific parts of the business exchange transaction can be represented in a concise way, and how this pattern can be implemented using Smart Contracts. It is argued that the proposed Shared Ledger Accounting system increases the quality of the contents from an accounting perspective as well as the quality of the system in terms of auditability and interoperability.

1. Introduction

"For the last 40 years, we have had the Internet of Information.; now, with Blockchain, we are getting to Internet of Value". Tapscott & Tapscott [1] talk about blockchain as the second era of the Internet. The showcase of blockchain is Bitcoin, but especially since 2018, its scope of application has widened enormously. Still, the common denominator in most foreseen applications is the economic exchange. This includes the standard order fulfillment, but transfer of public records, like real-estate, stocks, and patents, or logistics, peer-to-peer lending systems and crowdfunding are also special cases or parts of the economic exchange. According to [2], emerging blockchain technology has the potential to drastically change the environment in which inter-organizational processes are able to operate, as these processes can be executed in a trustworthy manner "even in a network without any mutual trust between nodes".

Blockchain and Smart Contract technology also suggests a new way to implement an Accounting Information System (AIS). How exactly this can be done and what the limitations are is still very much an open question [3]. A bit more has been written already about the possible benefits. Based on the literature so far, these AIS benefits are the following:

- **Immutability.** The public blockchain as the one underlying Bitcoin claims to provide an immutable tamper-proof storage for transactions that is completely under the control of the technology. This immutability contrasts with the traditional situation where data is under control of the IT center or cloud provider, always with the possibility of management overriding or third-party manipulation. The immutability greatly improves the integrity and verifiability of AIS data and diminishes the need for many of the administrative checks, although the need for a proper design of the control infrastructure is not taken away. The immutability claim is still under discussion [4] and has to be made differently for different blockchain implementations, but we take it as an interface assertion in this paper.
- **Actor-independence.** AISs are traditionally kept inside an enterprise and represent the company perspective on economic exchanges. Evidence from the environment, e.g. invoices from suppliers, is used by the auditor and considered important, but there is no systematic connection between the invoices sent from company A with the invoices recorded in company B. Triple-entry accounting [5] has been proposed earlier as an independent and secure mechanism to improve the reliability of financial statements based on a neutral intermediary, however, this requires dependence on a third party. A blockchain-based shared ledger (SL) can solve this problem. An actor-independent mechanism may not only drastically reduce the need for multiple copies of the same data, but also contributes to the validity of the transactional data because it is based on consensus.
- **Smart control.** Recording is one thing. Quality control of the recording process is another. Smart contracts encoded

* Corresponding author.
E-mail address: h.weigand@uvt.nl (H. Weigand).

https://doi.org/10.1016/j.is.2019.101437

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

with accounting and business rules can enable not only efficient control of the recording process [3], e.g. authorization checks, and error-detection, but also increase its effectiveness. For traditional internal control measures, auditors must check the design, implementation and operation. Implemented controls could have been switched-off. Building these controls into Smart Contracts that are accessible for auditors (or the parties they represent themselves) makes the design transparent, ensures a 1–1 implementation, and provides a transparent operation (preventive or detective).
- **Tight integration.** The AIS offers a representation of the (economic) reality of an enterprise, but so far relies on human interfaces with this reality. The "reality" consist of social and physical processes. Purchasing or invoicing are such social processes. With SL, the purchase order can be put into the blockchain or be tightly connected to it, so that the relationship between order and the AIS representation of it becomes 1–1. In terms of Grigg [5]: "the [SL] entry *is* the transaction". For physical processes, such as the delivery of physical goods, the blockchain combined with IoT infrastructure can achieve a close 1–1 correspondence by setting up the SL as the register of enforceable property rights. This register can be used by an IoT application (for instance, when granting access to a rented room or car). The Smart Contract itself can neither force the service provider performance nor enforce the physical transfer of a purchased good to a receiver, but it may generate a value liability token that is enforced if the obligations are not fulfilled. We also mention here the integration with and selective disclosure to other parties, such as tax and customs (real-time taxing), regulatory bodies, financial/integrated reporting and assurance services.

Other advantages mentioned in the literature are continuous assurance and real-time reporting, but in our view, these are not specifically bound to the blockchain technology. In spite of the potential advantages, only a few explorative papers have explored the design of a blockchain-based Distributed Ledger technology (DLT) See Section 5 for more details. The conclusion of our literature research is that an ontologically sound and truly consensus-based design is missing to date.

Against this research gap, the goal of this paper is to introduce an original DLT solution for AISs in a formal way, grounded in accounting ontology. The key research questions are: (1) what is an economic exchange with respect to the Financial Reporting Standards? (2) To what extent can economic exchanges be stored in Shared Ledger? (3) What are advantages of a Shared Ledger? (4) How can the Shared Ledger be implemented in DLT, using Smart Contracts? We build on the blockchain ontology developed in [6], which drawing on Enterprise Ontology [7] distinguishes between a Datalogical level, an Infological (platform-independent) and an Essential (conceptual) level (Fig. 1). The transaction as a unit of information is located at the infological level. The essential level represents what the actors effectuate when performing this transaction and how the social reality is changed, whereas the datalogical level describes how the information is stored.

In line with [8], we extend the REA business ontology [9] used in [6] for the essential layer to the core COFRIS accounting ontology [10,11] that is based on current Accounting and Financial Reporting Standards (IFRS) [12,13]. An innovative characteristic of COFRIS with respect to REA is that it does not put the economic event in the center, but the evolving economic relationship in which the economic exchange takes place. Hence events are not viewed in isolation, but as contributing to the development of the exchange. Because of this choice, COFRIS includes an ontological grounding of the obligation concept and provides a good

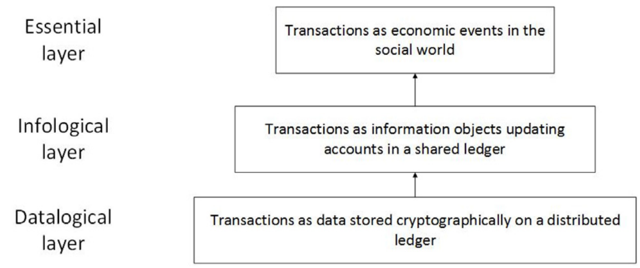


Fig. 1. Blockchain layers based on enterprise ontology. *Source:* Adapted from [6].

basis for a consensus view. From a Design Research perspective [14] the paper *builds* a Shared Ledger framework, *evaluates* the Shared Ledger with respect to accounting standards requirements by means of illustrative examples, and *demonstrates* its implementability in Smart Contracts.

In the following, we will use the term Shared Ledger rather than Distributed Ledger. The ledger is distributed at the datalogical level, but the key feature at the infological level is that the ledger is shared and provides a consensus view.

The paper has relevance for the IS field and for accounting. It is relevant for the IS field because (a) the AIS is traditionally at the core of enterprise IS (b) blockchain/SL is expected to be a crucial component of the future IS technology. Moreover, the value of ontological modeling in the IS field has been widely recognized, as witnessed for instance by the Formal Ontologies in IS (FOIS) conference series that had its 10th edition in 2018, and the development and use of languages like OntoUML [15]. The results of the paper should also be of interest to accounting researchers. Not only the recording and the related auditing process can gain substantially in efficiency, but also the accounting quality can be improved when the norms of consensuality and correlativity are adhered to. While SL Accounting could be beneficial for all branches of accounting, this article mainly regards Enterprise Financial Reporting (FR), as the most regulated and shared foundation. We do not consider the SL to be autonomous in the sense that regulations are not needed anymore. On the contrary, in order to gain the most from Shared Ledger approach, new regulations must be introduced, LEI[1] in particular.

The structure of this article is as follows. Section 2 summarizes the current research on SL accounting. Section 3 (theory) is a brief overview of the Economic Exchange pattern. Section 4 proposes and illustrates a framework for Shared Ledger accounting based on the Economic Exchange pattern, while in Section 5 the implementability is evaluated by providing a mapping to Smart Contracts at the infological level. Section 6 is the conclusion.

2. Related research

There are a few explorative papers on SL accounting. Dai & Vasarhelyi [3] sketch a system based on triple-entry accounting [5]. In this framework, each company keeps it double-entry bookkeeping system, but the blockchain ledger glues the two together, by (a) having a copy of each account of the local system in the DLT, and (b) adding "obligation" tokens and their transfer from one company account to the other that should match – perhaps enforced by Smart Contract – the Payables or Receivables, and (c) having aggregating accounts of total assets, liabilities and equities whose correspondence with the individual accounts can be monitored by a Smart Contract. Appelbaum & Nehmer [16] discuss the design requirements for a blockchain-based DLT system

1 Legal Entity Identification: see https://www.gleif.org.

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

and its repercussions for auditing tasks, giving special attention to cloud-based DLT solutions. When reviewing the triple-entry solution of [3] we wonder why still so much duplication of accounting entries is needed, given the DLT robustness. Grigg's original motivation was performance, but performance is a factor on the datalogical, not the essential level (Fig. 1). Furthermore, the status of "obligation" in the model of [3] needs more explanation. Although [3] already presents the main ingredients of an SL solution, an ontological definition is missing. Both referred papers are exploratory in nature. Wang & Kogan [17] introduce a blockchain-based AIS, including a prototype implementation. The main concern addressed in their paper is the tension between protection of private data and the desirable public blockchain transparency. The authors solve the tension using Zero-Knowledge Proof encryption. Apart from the encryption solution, the description of the AIS is sketchy. The paper defines a blockchain-based AIS as "a neutral and independent infrastructure that underpins business event recording" However, whether (or how) such a neutral representation – consensus view – is possible within current accounting standards, is not discussed. Another early paper [18] summarizes the developments in blockchain technology till 2017 and industry adoption for accounting with the Big4 companies.

Finally, we mention [2] that discusses challenges and opportunities of blockchain for interorganizational BPM. The paper does not touch AIS. This is regrettable, as most interorganizational processes are about economic exchanges, and then accounting and legal effects must be addressed somehow. As smart contracts allow for multiple levels, we suggest that, if for the surrounding coordination processes [7], including negotiation and failure handling, the DLT is indeed a good solution, the BPM contract is set up in such a way that the AIS smart contract is embedded in it and is called each time that an action is performed with accounting and legal effect.

It can be concluded that although the potential benefits of an SL AIS are recognized in several explorative papers, a formally worked out design is lacking. It can also be observed that the first works on blockchain and BPM do not make the connection between business processes and AIS, although this seems very relevant especially for interorganizational processes.

3. Theory of economic exchange

In this Section, we first define the concepts of economic exchange and valuation. Then, after an overview of the foundational ontology UFO (3.2), we present the COFRIS economic exchange pattern (3.3). As economic resources, as well as obligations and their fulfillment, play a central role in the economic exchange, these concepts are defined in 3.4 and 2.5, respectively, grounded in the UFO ontology.

3.1. The economic exchange and its valuation

In the 13th century, Fibonacci wrote: "When you will wish to exchange some merchandise for another merchandise, that is barter, you recall the price of each merchandise, which prices must always be in the same currency" [19]. Since then, not much has changed for economic interactions. For mutual benefit, market participants exchange economic resources and claims, whose value is expressed in monetary terms. It is important to note that the economic exchange can be a lengthy activity over time of which delivery and payment are parts.

Valuation is an indispensable component of economic exchanges. First it serves as a mean to mediate the exchangeables, secondly, it allows to partition exchange contract fulfillment in time and to facilitate the protection of resources, and thirdly it helps to assess the gain or loss for performed exchanges. However, valuation can be done in an enterprise-specific way, as is the current practice, or enterprise-independent way.

To illustrate the latter, consider a non-conventional example. Bitcoin mining is a production and exchange activity led by the Bitcoin network smart contract, in which new Bitcoins are added to the money supply. Some (eurozone) miner in the role of an offeror is expending computing power hardware and electricity resources (valued in EUR at direct cost) for verifying past transactions on the Bitcoin network. The network plays the role of both offeree and market, in exchange for the opportunity to compete to be rewarded with Bitcoins (valued at a market price in EUR).

In this example, competitive exchange activities occur and are recorded on a network and shared. In contrast, in conventional accounting "we do not evaluate activities as such but evaluate them by their effects on assets" [20]. Those effects are recorded separately at the enterprise, thus taking an enterprise centric perspective.

Enterprise ontologies depict the main objects and relationships of an organization, and functions and actions of a business. Traditional enterprise ontologies tend to take an enterprise centric perspective. In contrast, UFO social subontologies are based on UFO Social relator, its dispositions and manifestations, which support consensual and correlative relationships and interactions among social agents, allowing for contract, group or market perspectives with agent specific views. REA Enterprise ontology [9] conceptualizes economic exchanges, resources, events, and agents, and supports both an actor-dependent and an independent view, but is not grounded in upper ontology, nor has concepts to sufficiently cover the financial reporting domain.

In [10] an Economic exchange reference ontology and pattern was introduced in the context of enterprise-specific Accounting Information Systems, that later evolved into Core Ontology for Financial Reporting Information Systems (COFRIS) [11,21,22]. This exchange ontology is grounded on UFO — Unified Foundational Ontology [23]. The social, service and legal aspects of the exchange ontology were further developed grounded on UFO social, service and legal subontologies. Therefore, we will first briefly introduce the UFO subontologies used.

3.2. UFO social, service and legal subontologies

Unified Foundational Ontology [23] is presented using OntoUML diagrams and language [24,25]. The OntoUML is a UML profile whose metamodel reflects ontological distinctions and axioms of UFO. We will use OntoUML for our conceptualizations as well.

According to UFO-C, the Sub-ontology of Intentional and Social Entities [15], the exchange of *Communicative acts* creates *Social moments* such as *Commitments* and *Claims* that inhere in the Social agents involved in these communicative acts. *Social agents* are roles played by *Human agents and Institutional agents* (such as Enterprises). Two or more pairs of mutually dependent Commitments and Claims form a kind of social relationship between involved social individuals and is termed a *Social relator*. Social relators are important for our consideration because they are grounding Legal relators that in turn are grounding Economic relators. The latter underlie all relationships required for our ontology.

A commitment (internal or social) is *fulfilled* by an agent A if this agent performs an *action x* such that the post-state of that action is a situation that satisfies that commitment's goal. We assume that the social relator is:

- *Correlative* – meaning that one party's commitment and its fulfillment is a counterparty's claim and its fulfillment and vice versa,

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

- *Consensual* – meaning that the commitment and the claim and their fulfillment are agreed and immutable among parties.

Consensual does not imply that the commitment needs to be agreed upon at the very beginning. A social relator can first be *Offered* – meaning that the commitment (e.g. to exchange) is offered by one party but not yet agreed by the other party. We also assume like subsumed in UFO [26], that the commitments that are of interest for our ontology are *Appointments*, i.e., commitments whose propositional content explicitly refers to a *Time interval* (*Timing*), and *Complex closed appointments* that are composed of a number of commitments that should be achieved by executing a number of actions of a particular type, under certain types of situations (on the occurrence of a certain triggering event). These actions are part of a complex action — realization activity. The realization activity that fulfills a complex commitment is performed by the party or its agent for the benefit of the counterparty. Since the commitment specifies the execution of expected actions, it can be characterized by *Uncertainty* of fulfillment.

UFO-S — the Core Ontology for Services [27], which characterizes the service phenomena by considering service commitments and claims established between a service provider and a customer[2] along the service lifecycle phases: *Offer*, *Negotiation/Agreement* and *Delivery*. UFO-S presents general concepts spanning across several application domains so that its conceptualization can be reused for the economic exchange activity lifecycle.

During the *Offer* phase a service offering of the social agent – in the role of offeror – is presented by the provider to the eligible market of potential customers. Besides the offeror commitment, the offering also contains a customer condition — an offer of a commitment of the customer in the case that the customer accepts the offer.

In the *Negotiation/Agreement* phase, a potential customer negotiates the offering with the provider, who, after the acceptance of the offering, becomes a hired provider for the agreement (exchange contract) with the customer.

During the *Delivery* (exchange activity) phase the provider or its agent fulfills the commitments performing actions for the benefit of the customer that satisfy the goal of the commitment, in exchange for the customer fulfilling its reciprocal commitments by performing its actions for the benefit of the provider.

Legal aspects of service contracts were further elaborated in [28] within the UFO-L Legal Ontology, which is based on Hohfeld/Alexy's theory of fundamental legal concepts (see Fig. 2). A central element of UFO-L is the notion of legal relator, which is a social relator that is composed of externally dependent legal moments, each of which represents a legal position. The legal positions of UFO-L subsume claims and commitments in UFO-S, i.e., claim-right and duty, permission and no-right, power and subjection, immunity and disability. Two broader kinds of Legal positions are distinguished – *Entitlements* and *Burdens*, which we, following the economic and accounting traditions, further refer to as *Rights* and *Obligations* respectively. The above-mentioned right and obligation pairs form correlative associations [28], [12, para 4.25], which are foundations for a shared ledger perspective. Rights and duties for actions include also rights and duties to omit the actions (prohibitions). Since the obligations and rights are correlative, we may specify only one part of the pair for the contract or offering [10], usually the obligations.

3.3. Economic exchange life-cycle

We cannot describe the whole COFRIS ontology but will briefly recall the main economic events and relations of the exchange life-cycle within shared ledger context (see Fig. 3).

The Economic exchange [10] is conceived as a competitive *Exchange contract offering* of interaction made by an *Exchange offer* of one of two *Parties* (played by market participants), possibly followed by *Exchange agreement* with the *Counterparty* resulting in an *Exchange contract* of mutually beneficial *Exchange Obligations* to transfer *Contracted resources* (*claims*) with agreed *Valuation* of the rights to be received for the transfer. Legally speaking, an Exchange offer event transfers power to the offeree, who by accepting it, creates an obligation and a right for himself and the offeror to exchange. A contract is established.

The contract is to be *realized* by an *Exchange activity*[3] comprising:

- *fulfillment* of both parties' obligations by *Economic transfer* of instances[4] of *Contracted resources* (*claims*) in exchange for Rights to receive value (*Transaction price*) for the transfer,
- *realization* of value rights accrued for complete contract fulfillment by one party in exchange for claims against other party's unfulfilled obligations, and
- *settlement* of these claims by transfers of the other party.

An Exchange activity consists of possible economic transfer or contract breach events.

An *Economic transfer* event either conveys the rights over an object or the usage of such rights from one party to the other. Economic exchange events are performed by Market participants. Following [12,15] we define *Market participant* (or economic agent) as a UFO *mixin* played by social agents — persons and policy regulated enterprises, contractual groups of people and enterprises, rule regulated market, or the society at large, regulated by law. Market participants are capable of self and social committing and fulfilling economic actions, compliant with the market regulations. The contractual parties can be non-related, related, or represent different roles of the same market participant. Relationships between parties are consensual and correlative, e.g., being *related* should be agreed by both parties.

Economic exchange events create changes in enterprise-specific accounts (see the right part of Fig. 3) that will be described in the next subsection.

3.4. Economic resource — formal definitions

Economic exchanges are about resources and claims on resources. In the naïve view, the resource is identified with the physical good and the exchange with the logistic movement of the good. However, it does not really matter where the good is. Houses are goods that cannot move but still can be sold. From an economic perspective, the exchange is a transfer of value, and a distinction must be made between the good (e.g. the house) or service, and the right (e.g. ownership or use right) over that good or service. Briefly said, a resource is (a) a right over some object and (b) a right that has value.

Formally, in COFRIS the valuation for the transfer of a resource is regarded as potential or actual *right to receive* that value. A reciprocal social relator called economic relator is introduced to model relationships between transfer rights and obligations over an object, and rights to receive value (see Fig. 4). Market

2 In our model of economic exchange as an extension of service exchange, we generally use *party* and *counterparty* roles instead of *provider* and *customer*, because we do not give any priority to one of these two parties.

3 *Activity* denotes *Process* performed by a *Social agent*.

4 *Instance of* (iof) relationship here is used in UFO-MLT [29] sense and can include *subtype of* relationship.

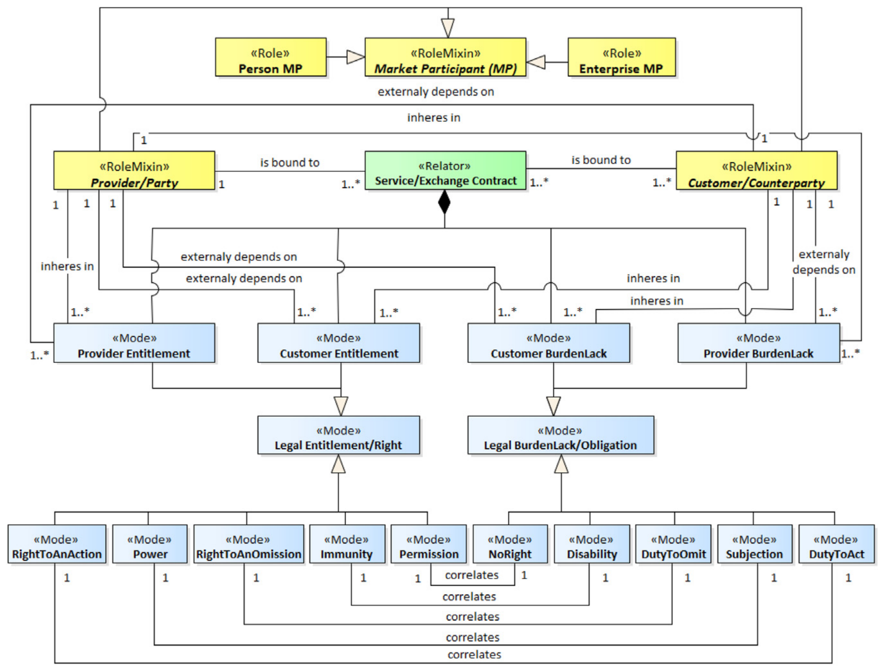


Fig. 2. An OntoUML diagram of UFO-L Service/Exchange contract ontology [28]. Market participants added. (Legend: Market participants in yellow, Legal positions in light blue, Economic relators in green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

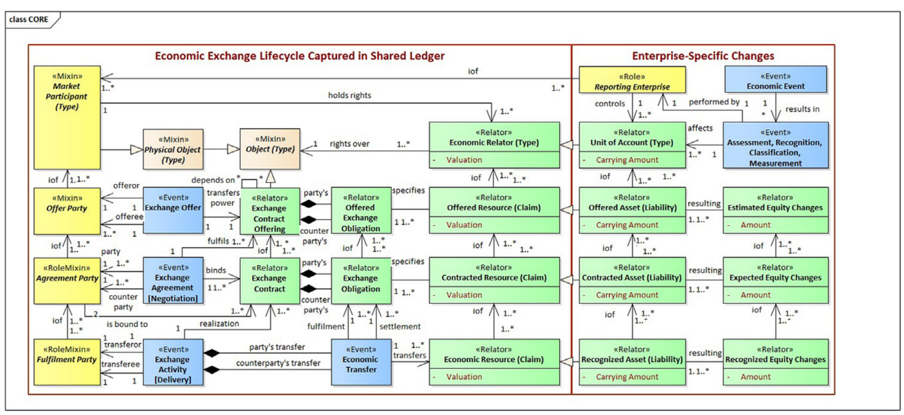


Fig. 3. COFRIS. OntoUML diagram of Economic Exchange life phases and their Enterprise-specific effects (direct Equity effects are not shown). Economic events in dark blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

participants enter into economic relationships that mediate a particular party with converse party — society and other parties. An *Economic relator (type)* is a UFO reciprocal legal relator [28], existentially dependent on involved market participants, that is a disposition of exchange of:

- Holder's rights or obligations (against Converse holder) - the disposition to transfer rights (or their usage) over some Object at some Timing, for
- Holder's right towards Target party to accrue value *for* the transfer.

Exchange obligation, a sub-kind of economic relator, represents an agreed promise to transfer a right (resp, obligation) over an object in exchange for a value right (resp, obligation) accrual.

Exchange contract integrates party's and counterparty's exchange obligations, governs their fulfillment, value exchange (realization) and settlement.

Economic resource, a sub-kind of economic relator, represents rights over an object that can be transferred (used) in exchange for value accrual – a right to receive value. Assuming that rights are allowed actions for allowed objects in allowed roles, we reconcile the above definition in terms of the one in [30]:

Economic resource is an *allowed role* played by an object in a transfer to make progress towards a goal of accruing value in an *allowed exchange activity*. The allowed activity and role are determined w.r.t. allowed object type and marketplace, disregarding the particular abilities of a holder (of a generic holder).

A transfer (including usage) should be physically or technologically possible, legally empowered (permissible), and financially feasible w.r.t. accrued value. *Converse party* for a resource is a society or a debtor. *Target party* for a resource is a target customer community [27] – ready to pay the price. A target party may be the holder itself – ready to incur the cost. *Value right* is accrued by the transfer of an economic resource and is to be realized after complete fulfillment or applied in a settlement. Thus, the timing of value right is within the production or exchange activity.

Economic claim, a sub-kind of economic relator, represents a duty or responsibility to transfer economic resources to which the market participant is legally or constructively bound (to make a settlement action). *Value obligation* is accrued by the transfer of an economic claim (an action different from a settlement).

Complex disposition is a group of economic relators which are offered, agreed, fulfilled, settled, transferred, or maintained together.

An *Economic transfer* event attributed by the *Business activity* type is performed by a *transferor* who fulfills an exchange obligation or settles an economic claim by:

- transfer of holder's rights or their usage *from* a stock controlled by a transferor *to* the stock controlled by a transferee. Transfer may occur instantly or over a period of time. *Stock* or address, in this case, is any collection of involved party's economic relators.
- accrual of transferor's value receipt rights for the transfer towards the transferee, (or settlement of an economic claim against the transferor).

Underlying *Object,* a UFO *mixin*, denotes the physical or social object, or their type characterized by *Quality, Disposition, Role, and Quantity* (of collective objects or *Amount* of matter).

Note that an economic relator itself can also be an underlying object (for another resource) thus modeling situations of power [28], e.g., when a *debt* (a right to receive from a converse holder) is transferred from one holder to a counterparty, or e.g., a *note* payable in Government bonds (an underlying object) gives the note holder the right to receive and the holder of the Government bonds the obligation to transfer Government bonds. The converse holder of the bonds is the Government. The underlying object of the bonds is cash.

An enterprise party specific exchange effects situation is depicted on the right side of Fig. 4. An *Economic event* affects the value or structure of an economic relator. The exchange events together with enterprise specific *Other events* or *conditions,* such as regulation, market participant, market price and underlying object changes, affect enterprise specific economic relators – assets (liabilities) – whose changes are specializations of transfers of resources (claims), extended by *intended activities* and *roles,* such as administration, sales of goods and rendering services and production.

Units of Account — Assets (*Liabilities*) are present rights (obligations) for resources controlled (claims indebted and unavoidable) by the Reporting enterprise, as a result of past activities which form their *Historical cost* [12] and *Experience*. The *Carrying amount* represents the present valuation and can be measured as historical or present (market) value. The *Intended role* of an asset (liability) is the one it will play in an *Intended activity*, such as raw material, labor, equipment and finished good.

Income (*Expenses*) characterize performance aspects of asset (liability) changes and the *Role performed* in a *Performed activity*.

A *Complex Account* is a group of controlled rights and/or obligations which are usually or mandatory transferred (fulfilled, consumed/used, produced, classified, valued) together, such as a business, cash generating unit, and economic contract.

While all exchanged resources (claims) are enterprise asset (liability) changes in Financial Reporting, some are regarded as *momentarily* [31], i.e., are transferred (consumed) as received. Momentarily assets (liabilities), such as services, increase (decrease) carrying amount of affected *stock* assets (liabilities or equity). Other asset (liability) changes are *recognized* for future recovery (transfer) or *derecognized*. *Classification, Valuation, and Uncertainty* of assets (liabilities) depend on the intended activities and roles, and can change as a result of their enhancement/impairment and market and own prices and risks.

An *Equity* is the value residual of assets (liabilities) and represents the claim of a group of enterprise owners – *Equity holder*. Changes in equity caused by inflow or outflow in the valuation of assets (liabilities) are classified as exchange *income or expenses*, or equity holder *contribution* or *distribution*.

Many assets (liabilities), such as human, digital, customer and environmental capital are not recognized or recognized as goodwill, because of the difficulties of quantifying and valuation, however, they are revealed in future transfer actions.

3.5. Obligations (rights) and their fulfillment details

Economic exchange is based on a contract, consisting of rights and obligations, so obligations must be first-class citizens in any AIS. Several obligations must be distinguished. An economic exchange generally consists of a number of actions separated in time, and the involved obligations progress through different levels and phases. Recognizing such levels and phases is important for the development of smart contracts that would capture shared ledger information and execute steps required for financial reporting.

As depicted in Fig. 5(a), the exchange contract recursively comprises of two valued conditional *Contract* (*exchange*) *obligations* (and correlative *Contract rights*) – one for an agreement party and another for a counterparty.

A Contract obligation comprises of several *Performance* (*exchange*) *obligations*, each aimed to create a *product* (a sub-kind of a resource) useful for the counterparty. As stated in e.g. ArchiMate [32] a *product* represents a coherent collection of services and/or passive structure elements (goods), accompanied by a contract/set of agreements which is offered as a whole to (internal or external) customers.

Performance obligation, in turn, comprises of *Transfer obligations*, which are required in order to fulfill a performance obligation. All obligations are fulfilled by economic transfer actions by the transferor (a role of an obligor) to the benefit of the transferee. The mutual fulfillment processes affect at least three realms:

- the transferor specific accounts, which represent the past (see Fig. 5(b), column 2);
- the consensual contract (shared ledger) perspective, which represents the past and present, and has consensual and correlative views of the transferor and the transferee, and the view of third parties, including Financial Reporting (column 1);
- the transferee specific accounts which tend to represent the future (column 3).

During the fulfillment, both parties may do their transfers concurrently (while their goal usually is not competitive). When all transfers for a contract obligation of one of the parties are done, its accrued values are accumulated into Contract value. The *Realization* event in turn exchanges party's (which becomes a *Creditor*) contract value for the consideration right of the other party (which becomes a *Debtor*), and raises an unconditional *Contract claim* against the other party. This exchange makes *Contract, Performance* and *Transfer value* rights exchanged and reclassifies Debtors's unfulfilled obligations as *Contract, Performance*, and *Transfer claims.*

The consideration right value typically is equal to the value of the contract claim, the exceptions are gifts, onerous contracts,[5] and the contracts where the values are determined based on the

[5] An onerous contract may arise in relation to the sale of commodities, when the market price declines below the cost required to produce a commodity. Another example of an onerous contract is when a lessee is still obligated to make payments under the terms of an operating lease but is no longer using the asset [13].

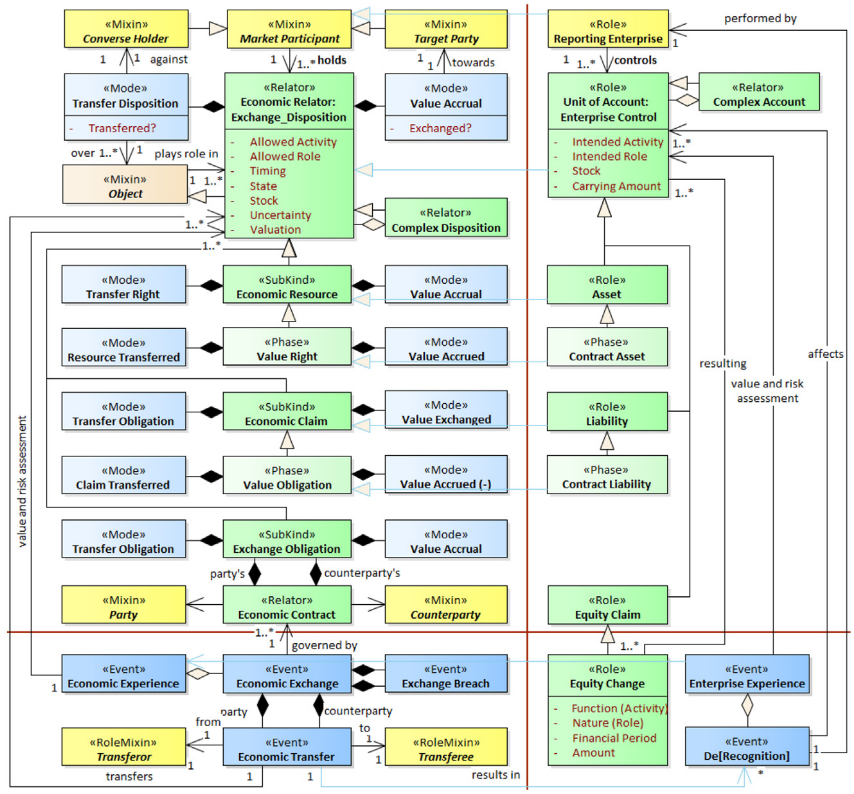
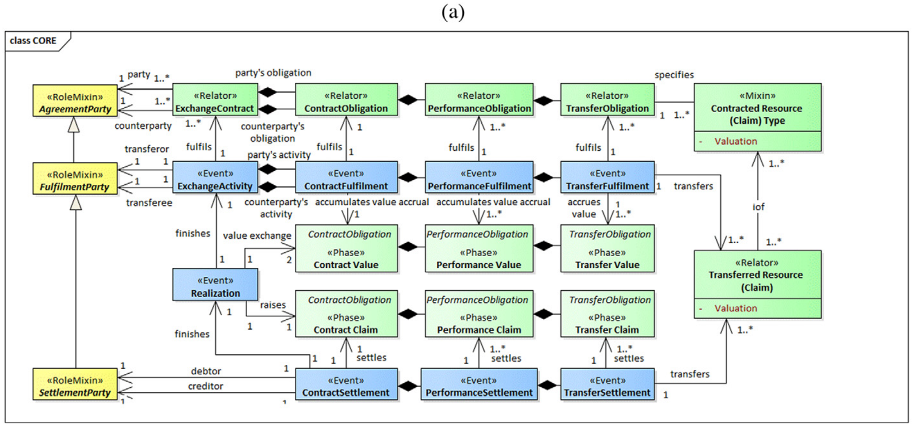


Fig. 4. COFRIS. OntoUML diagram of economic relator.

(a)



(b)

1. Economic Exchange Events				2. Transferor Enterprise-Specific Effects		3. Transferee Enterprise-Specific Effects	
Transferor (Offeror) view		Transferee (Offeree) view		Unit of Account		Unit of Account	
Transfer Obligation	Value Right	Transfer Right	Value Obligation	Outflow	Inflow	Outflow	Inflow
Exchange Agreement or Offer							
Party's Obligations							
[redacted]							
Counterparty's Obligations							
Transfer Right$_{PR,TR}$	Transfer Value Obligation$_{PR,TR}$	Transfer Obligation$_{PR,TR}$	Transfer Value$_{PR,TR}$	Contract Liability$_{PR,TR}$	Asset	Asset	Contract Asset$_{PR,TR}$
Performance Right$_{PR}$	Performance Value Obligation$_{PR}$	Performance Obligation$_{PR}$	Performance Value$_{PR}$	Contract Liability$_{PR}$	Contract Liability$_{PR,TR}$	Contract Asset$_{PR,TR}$	Contract Asset$_{PR}$
Contract Right	Contract Value Obligation	Contract Obligation	Contract Value	Contract Liability	Contract Liability$_{PR}$	Contract Asset$_{PR}$	Contract Asset
Party's and Counterparty's Fulfillment							
[redacted]							
...							
[redacted]							
Creditor's Realization							
[redacted]							
Contract Unconditional Right	Exchanged	Contract Claim	Exchanged		Receivable	Payable	
Debtor's Settlement							
Transfer Settled$_{PR,TR}$	Exchanged$_{PR,TR}$	Transfer Settled$_{PR,TR}$	Exchanged$_{PR,TR}$	Asset	Payable$_{PR,TR}$	Receivable$_{PR,TR}$	Asset
Performance Settled$_{PR}$	Exchanged$_{PR}$	Performance Settled$_{PR}$	Exchanged$_{PR}$	Payable$_{PR,TR}$	Payable$_{PR}$	Receivable$_{PR}$	Receivable$_{PR,TR}$
Contract Settled	Exchanged	Contract Settled	Exchanged	Payable$_{PR}$	Payable	Receivable	Receivable$_{PR}$

Fig. 5. COFRIS. Economic contract fulfillment. (a) OntoUML diagram, (b) Exchange events and their accounting treatment. (PO/PR and TO/TR — performance and transfer obligation/right).

current market prices. The exceptions lead to recognition of gain or loss (changes in equity) in the enterprise-specific accounts.

The contract claim (receivable) raises claims for all included obligations (and offsets income and revenue value rights) of the debtor. Currently in accounting accruals and receivables are differentiated such that the latter require a special agreement on invoice, while the former could be estimated [13, IAS 37]. We argue, that in the Shared Ledger and Standard Smart Contract environment explicit invoicing can be minimized, but transfer consensus should be established via contract or suspended for dispute.

As in the *fulfillment*, debtor *settlement* transfers resources (claims) of the promised resources (claims) type but does not accrue any value, because the value is already received. When all transfer claims of a performance claim are satisfied, it becomes *settled*, and *revenue* may be recognized at debtor accounts. Settlement of all performance claims finalizes the contract.

The valuations for each party are related by the following equation:

$$\text{Valuation}_{\text{Contract}} = \sum \text{Valuation}_{\text{Performance}} = \sum\sum \text{Valuation}_{\text{Transfer}}$$

And the stocks of each party are aggregated in the following way:

$$\text{Stock}_{\text{Contract}} = \cup\ \text{Stock}_{\text{Performance}} = \cup\cup\ \text{Stock}_{\text{Transfer}}$$

This means that the valuation could be given for the contract or the performance and calculated for the transfers and vice versa, and the stock given for a contract, as e.g. a stock representing the whole enterprise, or for a performance, e.g., a stock representing a construction project, is a stock for a service transfer, e.g., labor.

All events may be actioned by the market participant or its agent or specified in a smart contract as automatically executable — triggered by conditions, and timing of fulfillment, realization and settlement.

4. Design of a Shared Ledger framework

In this Section, we first identify what actually must be shared in the Shared Ledger and formulate three basic axioms that the SL accounting should adhere to (4.1). In 4.2, a Shared Ledger, based on the COFRIS ontology described in Section 3, is illustrated at the instance level for a standard sales transaction and we show advantages for the calculation of revenues and other cases, like in cost-plus contracts with multiple parties.

4.1. Correlative consensus as the key feature of SL

An advantage of the shared ledger is the actor-independent shared perspective that it offers — in terms of the exchange ontology, correlative consensus. This does not necessitate that *all* information is accessible to all parties. Information sharing in a shared ledger must be selective, ranging from public, i.e., among all members of society at large or market, to particular (private) – among members of a contractual group, a party and a counterparty, or participants within an enterprise. The accounting interpretation of the contracts and their fulfillment may be different for each party. Still, the goal should be to obtain a more correlative consensus, for resource (claim) interpretation in the contracts and transactions, while preserving sensitive information.

We assume that conceptually there is a shared contract – a pair of mutual obligations of the parties, and contract fulfillment exchange interactions in consensus. The following minimal characteristics of a contract should be shared for Financial Reporting:

- Contract activity type, id, inception date, parties, legislation, obligations;
- Transfer activity type, id, transferor, transferee, period or date, currency, and for each fulfillment, realization or settlement event to include:
  - fulfilled obligation (settled claim) level and identity, types and instances of transferred resource (claim) timing, rights (obligations), transferor and transferee stock, object, quantity and;
  - the value of the accrued rights (obligations) to receive (received).

The AIS tagging of the entries may be different for different agents, for several syntactic and semantic (objective) party specific reasons, such as:

- specific financial period, stock account name, unit of account granularity, local currency, rounding rules, and other future-oriented qualities;
- specific resource function (purpose) or restriction;
- different accounting classification, and valuation requirements.

Therefore, in COFRIS market participants may specialize/generalize at recognition/derecognition the resources (claims) in consensus, as their own assets (liabilities) and corresponding income (expenses) per accounting standards and their own operational purposes and include their specific (de)recognition modules into smart contracts that extend the contract manipulation and transfer events. For example, if a provider sells a product, such as fuel, the customer may classify it as a raw material, as held for sale, or for administrative expense — all these asset types are subtypes of the transferred resource.

However, many enterprise-specific characteristics of a transaction can be determined or restricted, and the mapping rules for the smart contracts established, by the:

- contract type and other party type,
- rates and prices, existing in the market,
- party's disclosed policies, financial period, transaction history,
- party's open profile characteristics, such as geographic area, base currency, ordinary activities, and major product lines,
- organization structure and operational segments, and
- form of incorporation and legislation.

The disclosure of enterprise specific information to the counterparty can be required by a contract or voluntary for improvement of resource (claim) specification, but not sensitive to party's confidentiality. The disclosure can be less sensitive if provided on the party, resource, activity high type level instead of instance level, the latter typically is not required for Financial Reporting.

The conventional accounting often loses the semantics of transfer events, because it recognizes only the *effects* of resource transfer instead of transferred resources themselves. The capturing of interactions that are shared and in consensus should serve as an additional source for (financial) disclosures. An example is services or other resources that are consumed as transferred. The accounts usually recognize only their effects and carrying value increase in e.g. equipment for which installation and testing services were provided. In general, we propose to have the transfer events with the transferred resources (claims) shared and the enterprise-specific effects of the interactions on the respective accounts, to be not shared (although this account information can still be part of the smart contract). The benefits of observing events in addition to observing their effects (focus) are classified in [33].

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

To maintain consistency, the exchange phenomena should be correlated in the shared ledger as depicted in the table of Fig. 5(b). This includes not only relationships like rights vs. obligations, but also events, e.g. transfer vs. receipt, and roles. The transferor view depicts the events for the contract, transferee shared consensus appears as a correlative view. Specific accounts of the parties — assets (liabilities) are specializations of the resources (claims) affected by the transfer event.

Exchange Axioms

The Shared Ledger is bound to the following three exchange axioms pertinent to the consensual part (*how* the consensus is achieved is not relevant at this level):

• *Transfer correlativity* – for each economic transfer event, the economic resource (claim) received by the transferee (one party) is the resource (claim) provided by the transferor (other party): the transferred resource (claim). In other words: there is a consistent actor-independent view of the transfer. This includes correlativity of exchange obligations.

• *Economic transfer* – each valid transfer event fulfills a prior exchange obligation with value accrual or settles a prior claim.

• *Realization* – complete fulfillment of exchange obligations by one party causes value exchange among parties resulting in raising claims against the other party for the unfulfilled obligations.

4.2. Illustration

We illustrate and validate the Economic exchange ontology [10] and its extension for shared ledger by instance cases and accounting standard schemas (see Table 1, for the format and the first case), represented in the form of a hierarchical Economic event table. Particular attention is given to the question of what should be shared in the shared ledger and what should not. Besides being used for examples, the table is intended to be used for the presentation of instances and assertion counterexamples for validation of COFRIS conceptual models using logic-based language Alloy and its analyzer [24]. The example adheres to the exchange axioms.

This event table has columns similar to the event logs used in process mining — process instance, event type, the performer or originator of the event (i.e., the actor executing or initiating), the timestamp and location of the event, data elements recorded with the event [34] *plus* the rights transferred or used over some object in quantity in exchange for value, referring to fulfillment and realization of the obligations (settlement of the claims) by the event and resulting party specific account and amount changes.

In the header (in dark blue) of an economic event, we have:

- *Event identifier* (EID),
- *Transferor type* - Customer or Provider (or more specialized role), that specifies the context view for the transfer,
- *Event type* - Offer, Agreement, Fulfillment or Settlement of Transfer, Performance or Contract obligations, Realization, Other Recognition, Reclassification, Revaluation (or more specialized subtype),
- *Date* or *Period*, and
- *Currency Unit,*
- *Provider* and *Customer identification,* and their
- *Local currency units* with the spot exchange rates.

Event detail lines depict events that fulfill (or settle) the obligations (or claims, respectively) identified by the referenced event, PO number and type characteristics, by transferring the promised economic resource (claim) instances in exchange for rights to receive (received) value or are triggered by the PO fulfillment or contract realization.

The *PO/PR – Performance Obligations/Rights*, *Timing*, *Rights* over an *Object*, *Quantity*, *Value* (and *Valuation*), *Stock* column concepts were briefly introduced in previous subsections. When contractual rights/obligations against a converse holder are conveyed, the Object column will reference the transferor's rights/obligations EID. The Value column may represent an amount or refer to an offer EID that would provide a valuation for the given context, e.g., a market price or a currency rate.

Provider and Customer have their specific, but potentially correlative columns that depict involved *Debited/Credited Accounts* and *Amounts*.

4.3. Application

Related to the example (Table 1), we show some representative applications of SL accounting and its benefits.

Example Case 1. Correlative and consensual product transfer fulfillment and revenue recognition. Enterprise A enters into a contract with customer B, identified and depicted by event id — EID:21 (that fulfills some offering created by EID:20), whereby A takes obligations to exchange some widget goods (PO:1), valued at 100€ and an accompanying setup (PO:2), valued at 10€, by specified dates, for the rights to receive cash of 110€ in the A's bank account (IBAN) by 2018.09.30. These rights/obligations are depicted in the agreement details, but the *expected* effect of their fulfillment is specified by provider and customer accounts, and amounts.

Event 22 fulfills A's obligation by transferring the goods promised by the Event 21 and accruing the A's Right to receive value; recognizing Contract asset of PO:1 and revenue. Event 23 provides setup services, recognizes revenue for PO:2, and completes performance obligation fulfillment that in turn leads to A's realization event that exchanges Consideration (value) right for Contract (value) right and recognizes Receivable. Finally, Event 24 represents customer's cash transfer and settlement of Payable raised in Event 23.

In this example, we consensually recognize revenue, as a result of the fulfillment of Performance obligations in the SL. Extending [13] we regard *Revenue* as inflow arising in the course of an enterprise ordinary performance, and fulfillment of performance obligation/claim agreed with a counterparty. It implies that a specific Contract asset of PO — Revenue right and correlative Contract liability of PR is recognized by the counterparty. Such an asset (liability) besides revenue recognition is important to distinguish for legal purposes, in the cases of contract breaches. For example, if setup for some reason is breached by the provider, it still may require the payment for the completed transfer of Widgets, but the latter should be consensually approved as a completed performance. In contrast, the Widgets may be unusable by the customer without setup, in such case, the provider can lose rights for the whole payment.

Example Case 2. Correlative and consensual prepayment accounting. If a contract is specified in foreign currency, accounting interpretations [13] require contract asset/liability revaluation into local currency according to the actual exchange rate. Per [13] a *Contract asset* is a party's right to (receive) consideration, in exchange for resources (claims) transferred to a counterparty, conditioned on something other than the passage of time (for example, the party's future performance), and *Contract liability* is an obligation to transfer resources to a counterparty for which the party has received consideration. These definitions are forward-looking and assign some features of the receivable (payable) resource to these assets (liabilities). Thus IFRIC 21 [13] interprets Contract liability, formed from prepayments, as a source for future non-cash assets and thus not subject to revaluation.

Table 1
Economic event table for Example Case 1.

EID:21 Provider Agreement 2018-08-29						CU:	€		Provider: A		€		Customer: B		€
Fulfils:	Obligation	PO	Timing	Rights	Object	Qty	Value	Stock	Debits	Credits	Amt	Stock	Debits	Credits	Amt
EID:20	Obligation	1	2018-08-29	Ownership	Widget	5	100		Cost	Finished goods	70	1	Raw materials	Contract Liability	100
									Contract Asset	Revenue	100				
		2	2018-08-30	Services	Setup	1h	10		Cost	Labor	10	1	Raw materials	Contract Liability	10
									Contract Asset	Revenue	10				
	Consideration		2018-09-30	Rights	Cash		110	IBAN	Cash in Bank	Receivable	110	IBAN	Payable	Cash in Bank	110
EID:22 Provider Transfer 2018-08-29						CU:	€		Provider: A		€		Customer: B		€
Fulfils:	Obligation	PO	Timing	Rights	Object	Qty	Value	Stock	Debits	Credits	Amt	Stock	Debits	Credits	Amt
EID:21	Fulfillment	1	2018-08-29	Ownership	Widget	5	100		Cost	Finished goods	70	1	Raw materials	Contract Liability	100
									Contract Asset	Revenue	100				
EID:23 Provider Transfer 2018-08-30						CU:	€		Provider: A		€		Customer: B		€
Fulfils:	Obligation	PO	Timing	Rights	Object	Qty	Value	Stock	Debits	Credits	Amt	Stock	Debits	Credits	Amt
EID:21	Fulfillment	2	2018-08-30	Services	Setup	1h	10		Cost	Labor	10	1	Raw materials	Contract Liability	10
									Contract Asset	Revenue	10				
	Realization		2018-09-30	Rights	Cash		110	IBAN	Receivable	Contract Asset	110	IBAN	Contract Liability	Payable	110
EID:24 Customer Transfer 2018-09-30						CU:	€		Provider: A		€		Customer: B		€
Fulfils:	Obligation	PO	Timing	Rights	Object	Qty	Value	Stock	Debits	Credits	Amt	Stock	Debits	Credits	Amt
EID:23	Settlement		2018-09-30	Rights	Cash		110	IBAN	Cash in Bank	Receivable	110	IBAN	Payable	Cash in Bank	110

However, we advocate the present view to these Contract assets and liabilities, meaning that they represent *in consensus* cash rights/obligations for the transferred resources, to be reimbursed in the case of a breach (for example, a return of a prepayment), thus they need to be constantly revaluated and maintained in consensus. This is arguably increasing the quality of the accounting and contractual information.

Example Case 3. Enterprise-specific transaction information disclosure to the principal. Cost-plus Contracts. A, a construction company, enters into a cost-plus contract with a customer B to build an object. What is the advantage of sharing information in a SL system with Smart Contract? B reimburses A for all its allowed expenses plus an additional variable payment that allows A to make a profit. A enters into contracts with subcontractors and vendors — Cs and allows these contracts and contract events [complying to IFRS requirements] to serve as inputs to the contract with B, sharing with B [and a global Financial Reporting system] all the required details in consensus with Cs, possibly omitting the names of Cs. Furthermore, in consensus with B, A shares all the required and non-sensitive details of the contract with B with a Financial Reporting system. During the warranty period, B shares all relevant events involving the built object with A. This set-up benefits from having a single source of truth, simplifying administrative and control procedures, and the possibility of automated execution of the smart contract. The example shows that both consensual data and enterprise-specific disclosure are required, as provided by the COFRIS SL ontology

Example Case 4: Unsymmetrical accounting standards. **Leasing**. It is important that provider and customer share and have consensus on the asset/liability evaluation/classification, especially in the case of obligations remaining/ongoing, such as in a lease. Unfortunately, existing accounting standards [13] are ambivalent on the correlation and prescribe different (not-correlated) lease accounting for the lessor and lessee [11]. When deciding between services and lease, the decision is not-correlative, while the decision has certain accounting consequences. It is quite possible that the lease asset and lease liability are recorded at different amounts in the books of lessor and lessee. Having a consensual and correlated SL increases the quality of the contractual and accounting information.

Example Case 5. A distinction between shared ledger resource and enterprise asset concepts. A corporation develops software products (intellectual property assets) and sells MRP II Software licenses — an economic resource with a market value of 200K€ per license. A manufacturing company buys this software license for 180K€ to be used in an MRP resource role in its manufacturing activities and recognizes the license combined with implementation services (a transferred resource) as a product and an asset at a cost of 250K€ to be amortized (i.e. used as a resource) by 25K€ per year. Soon after the purchase and implementation, the manufacturing activities are discontinued due to political sanctions — an economic event affecting the asset. The asset does not have a use value anymore and did not have an exchange value initially, because the company did not have sublicensing rights, and the company does not have any realistic opportunity to use the license to service other companies. The value of the asset is nil, and the asset is derecognized, while it still counts as an economic resource in a market perspective. It is *held* (owned) by the company, but it is not *controlled* by the company.

4.4. Evaluation

In this section, we have designed an SL accounting system on the basis of the COFRIS ontology, in terms of what is shared in the system and three basic axioms that the system must adhere to. The design is evaluated tentatively by means of instantiation for a typical sales transaction, and by showing the benefits of the SL approach for cases like revenue recognition, intermediate contract resources, lease contracts, and cost-plus transactions. Although the technical implementation has not been realized yet, the design shows that (on the conceptual level) the COFRIS exchange pattern is a solid and feasible basis for SL accounting and that such a system can not only meet the current requirements on accounting information systems but also has additional benefits.

5. Towards shared ledger implementation

Having established that the Economic Exchange pattern can be used as the basis of SL accounting on the conceptual level (essential level, in terms of the blockchain ontology of Section 1), we now turn to the question how such an AIS can be implemented. Following the blockchain ontology (Fig. 1), a distinction must be made between the datalogical level and the infological level. As the blockchain and smart contract implementation platforms are still very much under development, we focus on the infological level. In other words, we are interested in an infological model of the SL AIS that is (a) realizable in current or to-be-developed blockchain platforms and (b) sufficient as target of a model transformation from the COFRIS exchange pattern. We cannot claim that the infological model developed in this Section meets all requirements yet, but we propose it as step towards this goal.

5.1. Principles of smart contract SL implementation

It might seem trivial to realize an AIS on something (blockchain) advocated as a "Distributed Ledger". However, more is needed than a logistic transfer of money or other resource tokens. To meet the requirements set forward in our ontological analysis — in particular, the distinction between consensus and specific information, and the ability to deal with the whole contract cycle – a basic blockchain like Bitcoin clearly does not

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

suffice. However, the contract accounting model can be realized by (translated and extended to) a Smart Contract-based Shared Ledger model. We start by listing the most important principles for this realization:

1. *Smart contracts* of market participants, containing mutual obligations of resource (claim) transfers, including information sharing specification, and IFRS relevant characteristics, are added to a shared ledger by consensus of the parties. Smart contracts comprise a hierarchy of rules and include general principles and regulations, particular rules in consensus, and rules specific to the particular agent for relating its assets (liabilities) to transferred resources (claims).
2. A *Digitized resource* (*claim*) or "token" represents the valued rights of a market participant (over an underlying object) which can be transferred to a counterparty by simply transferring the token. For a referenced resource the token transfer can be a representation of another action of rights transfer or it can effectuate the rights transfer itself (depending on the legal context). As explained earlier, in addition to dealing with "traditional" resources such as the ownership of goods and (crypto-)money, *rights/obligations* that mark the progress of the Exchange lifecycle should also be considered. An atomic transfer event happens in a point in time or over time when, fulfilling contract obligations, tokens for rights/obligations of resources are conveyed from one market participant to another, in correlation. We adopt the word "token" here for the time being in a naïve sense, without binding ourselves to a specific implementation, and will be more precise below.
3. Transfers of digitized resources (claims) are *immutably recorded* in consensus in a shared ledger, completely, distinctively or partially fulfilling the smart contracts. Transfers, alongside caused claims accrual or settlement, are accounted within shared ledger by smart contracts, including information sharing and IFRS relevant characteristics.
4. The *effects* of transfers involving resources (claims) are [de]recognized as assets (liabilities and equity) changes and corresponding income (expenses) per IFRS requirements [13] and enterprise policies in the shared or in the individual ledger part, according to information sharing specification.
5. *Financial Reporting* relevant information gathered in activities 1 through 4 is abstracted to the type level, hiding sensitive instance details and forming an enterprise's multi-dimensional cube within a global Financial Reporting system.
6. The multi-dimensional cube is then aggregated, calculated, viewed, and mined per the IFRS [13] Taxonomy requirements and financial reports are issued. Market participants build their own *accounting reports* using the shared ledger.

5.2. Infological model

With the advent of smart contracts, also known as coded contracts on the blockchain that automatically move digital assets according to arbitrary pre-specified rules [35], it is possible for two or more parties to exchange digitized resources without the need for a trusted third party [36] (although some new kind of intermediaries may arise). The implementation of smart contracts by enterprise adopters represents a major transformation in business computing, with organizations take advantage of the benefits of perfect replication and high availability, cryptographically-verified transactions, and lucid, robust business logic [37]. Various DLT protocols are being developed around the world that enable the exchange of tokens through smart contracts, Bitcoin and Ethereum being the most affluent today. Hereby, the intent of smart contracts is to both verify and enact the full logic of any given transaction. While Bitcoin scripts are not powerful enough to write full DLT applications to support the AIS, Ethereum's smart contract languages (e.g. Solidity, Serpent, Pack) and underlying virtual machines are powerful yet dangerously unconstrained [37].

To separate content and form, we designed an infological representation of an IFRS compliant based smart contract (Fig. 6). Hereby, we have slightly extended the infological blockchain domain ontology as presented in [6] to accommodate COFRIS-related components. In addition, we present a platform independent ERC20 compliant representation at the datalogical layer in order to comply to implementation standards and prevent the mentioned implementation constraints. At the infological layer, the notion of transaction has been refined to three levels: transaction – event – posting. This refinement is problematic for blockchain, but very well possible in a Smart Contract based SL. The transactions are stored on the SL blockchain.

The classes that are depicted in Fig. 6 are defined in Table 2 with references to the corresponding COFRIS concepts.

In Section 3, we identified three exchange axioms. To realize the transfer correlativity we assume an equivalent axiom on Digitized Resource transfers. On the datalogical level, the axiom must be (and typically is) guaranteed by the token transfer protocol.

As uniform standards are essential to (1) ensure interoperability between these different blockchains and (2) increase security of smart contracts in general, we apply the ERC20 common list of token rules to design a smart contract for an AIS. ERC20 is considered to be the standard that facilitates low friction peer-to-peer exchange of so called WRC20 tokens on DLT. The protocol proposed by Ethereum is intended to serve as an open standard and common building block, driving interoperability among decentralized applications (dApps) that incorporate exchange functionality. In this context, a token hosted on the DLT can be sent, received, checked of its total supply, and checked for the amount that is available on an individual address. This is analogous to sending and receiving Ether or Bitcoin from a wallet, knowing the total amount of coins in circulation, and knowing a particular wallet's balance of a coin. A smart contract that follows this standard is called ERC20 compliant. The functions and events are listed in Table 3.

Tokens vs. Digitized Resources The infological model does not talk about tokens, but about Digitized Resources. Here the difference between the infological and datalogical level is essential. Digitized Resources can be represented 1–1 by ERC20 tokens, but this is not necessary. There are still many uncertainties about the evolution of smart contracts. It may be that only *some* digitized resources are represented by tokens – e.g. the ones representing the money, goods or service exchanged – while the state of others can be derived from the transaction journal by the smart contract. Moreover, there are other implementation platforms, for instance, Hyperledger Fabric[6] that supports the concept "asset" and not tokens. According to the documentation, Hyperledger assets can range from the tangible (real estate and hardware) to the intangible (contracts and intellectual property) – we interpret this as saying that the Hyperledger assets *represent* these tangible or intangible resources on the datalogical level. They are represented simply by key–value pairs. Hyperledger Fabric provides the ability to *modify* assets using chaincode *transactions*. For the time being, our claim is that (in ERC20 compliant implementations) tokens can be used and should be used for those Digitized Resources that exist beyond the context and life-time of the smart contract.

6 https://hyperledger-fabric.readthedocs.io.

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

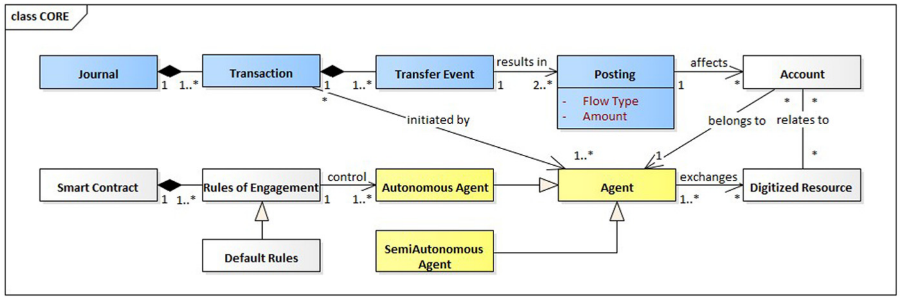


Fig. 6. Infological shared ledger model UML diagram. Agents in yellow, events in blue.

Table 2
Infological class definitions with mappings.

Class	Explanation
Shared ledger	A shared **ledger** is a set of accounts maintained in a smart contract. Shared ledgers can be said to be part of a (perhaps global) SL environment (not in the model).
Account	An (infological) **account** is a container of digital resources corresponding to a COFRIS unit of account (Stock): a group of enforceable/constructive rights and/or obligations, It may map to a datalogical account of tokens.
Agent	A **semi-autonomous agent** is the owner of accounts, registered within the SL environment, corresponding to a market participant in COFRIS. An **autonomous agent** is a registered Smart Contract that initiates a transaction, based on the commitments materialized in the rules of engagement. An agent can initiate a transaction.
Transaction (transfer, posting)	A **transaction** is a coherent set of **transfer**s that make **postings** on accounts corresponding to a COFRIS Economic event (including COFRIS fulfillment, realization, settlement)
Journal	A **journal** is a chronological list of transactions controlled by a Smart Contract. At the datalogical level, it corresponds to a continuously growing list of records called blocks. Each block contains a timestamp and a link to a previous block.
Digitized resource	**Digitized resources** represent the COFRIS Resources (Claims) or other rights which can be transferred to a counterparty by simply transferring the digital representation
Rules of engagement	**Rules of engagement** refer to the explicit rules to which a transaction should behave as specified in a Smart Contract including exchange pattern
Default	A **default** is a definitional clause, which defines relevant concepts occurring in the contract as constant variables

Table 3
ERC20 functions and events.

AccountBalance	The **AccountBalance** returns the balance of a smart contract account
ClauseFunction	**Clause functions** of a smart contract interface implement the rules of engagement at the datalogical level
TotalSupply	The **TotalSupply** returns the total supply of a certain token within a smart contract
TransferEvent	A **TransferEvent** is called to transfer tokens between accounts identified by an address. An infological transfer can be mapped to a TransferEvent.
ApprovalEvent	An **ApprovalEvent** is called by the TransferBetweenParties function to make tokens available to the Smart Contract
ClauseEvent	A **ClauseEvent** is an event that is triggered by a ClauseFunction

However, we do not commit here to one datalogical model or one possible mapping.

Accounts. In ERC20, an account is identified by an address. At the infological level, we propose to have a directory structure of accounts, where the roots are the agents. For instance, *P/r* stands for the *r*-account of agent *P*. In this way, also subaccounts are possible, for instance `P/raw_materials/internal_consumption`.

Transactions

The key COFRIS economic events (Fig. 5) are mapped to the following infological transactions that should be seen (and implemented) as units, with the same ACID properties that we know from databases:

agreement $(\mathtt{t}:\mathtt{time}, \mathtt{P}:\mathtt{agent}, \mathtt{R}:\mathtt{agent}, \mathtt{x}:\mathtt{obligation}, \mathtt{r}:\mathtt{right\_type}, \mathtt{o}:\mathtt{object}, \mathtt{v}:\mathtt{value})$

fulfillment $(\mathtt{t}:\mathtt{time}, \mathtt{P}:\mathtt{agent}, \mathtt{R}:\mathtt{agent}, \mathtt{x}:\mathtt{obligation}, \mathtt{r}:\mathtt{right\_type}, \mathtt{o}:\mathtt{object}, \mathtt{v}:\mathtt{value})$

realization $(\mathtt{t}:\mathtt{time}, \mathtt{P}:\mathtt{agent}, \mathtt{R}:\mathtt{agent}, \mathtt{x}:\mathtt{obligation}, \mathtt{v1}:\mathtt{value}, \mathtt{y}:\mathtt{claim}, \mathtt{r}:\mathtt{right\_type}, \mathtt{o}:\mathtt{object}, \mathtt{v2}:\mathtt{value})$

settlement $(\mathtt{t}:\mathtt{time}, \mathtt{P}:\mathtt{agent}, \mathtt{R}:\mathtt{agent}, \mathtt{y}:\mathtt{claim}, \mathtt{r}:\mathtt{right\_type}, \mathtt{o}:\mathtt{object})$

Here the fulfillment stands for the resource transfer that fulfills the obligation. It is one transaction, so there cannot be an isolated resource transfer (thus implementing the fulfillment axiom of Section 3). The obligations are created in the agreement transaction. The four transactions also correspond directly to the four parts of Fig. 5(b), and each of them consists of one or more infological transfers, as worked out below for fulfillment.

Transfers

- The transfer `resource_transfer(P:agent, R:agent, r:right_type, o:object)` stands for the transfer of resource *(r,o)* from Provider (Transferor) to Receiver (Transferee). A resource_transfer consists of the following balanced set of postings, for $r = \mathtt{control}$ or some other right type

  $\{\mathrm{out}(\mathtt{P/r}:\mathtt{account}, \mathtt{o}:\mathtt{object}), \mathrm{in}(\mathtt{R/r}:\mathtt{account}, \mathtt{o}:\mathtt{object})\}$

In terms of tokens, it means that a *DigitizedResource object token* is transferred from (owenership) account of P to the (ownership other other-typed) account of R.

- The transfer `fulfill(P:agent,R:agent, x:obligation, v:value)` stands for the fulfillment of obligation *(x,v)* from Provider P to Receiver R. *(x,v)* is the transfer obligation *x*, or a part of it, because it may be that the obligation is fulfilled in several steps. The fulfill transfer consists of the following balanced set of postings, given here for transfer obligation *r*, which may be referenced by a birth event identifier or a composite key. The (sub-) account is identified by a combination of the agent, the right type and the obligation identifier.

$$\{\mathtt{out}(\mathtt{A/transfer\_obligation/r} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{in}(\mathtt{A/asset} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{out}(\mathtt{B/transfer\_right/r} : \mathtt{account}, \mathtt{v} : \mathtt{value},$$
$$\mathtt{in}(\mathtt{B/liability}, \mathtt{v} : \mathtt{value})\}$$

- The transfer `exchange_values(P:agent,R:agent, x:obligation, v:value)` stands for the exchange (offset) of transfer obligation values of P and R in the case of complete fulfillment of transfer obligations *(x,v)* from Provider P to Receiver R, and raising of a claim against Receiver R for unfulfilled obligations. The exchange_values transfer consists of the following set of postings, given here for transfer obligation*r*, which may be referenced by a birth event identifier or a composite key. The (sub-) account is identified by a combination of the agent, the right type and the obligation identifier.

$$\{\mathtt{out}(\mathtt{A/transfer\_right/r} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{in}(\mathtt{A/receivable\_claim/r} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{in}(\mathtt{A/liability} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{out}(\mathtt{A/asset} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{out}(\mathtt{A/liability} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$

$$\mathtt{out}(\mathtt{B/transfer\_obligation/r} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{in}(\mathtt{B/payable\_claim/r} : \mathtt{account}, \mathtt{v} : \mathtt{value}$$
$$\mathtt{in}(\mathtt{B/asset} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{out}(\mathtt{B/asset} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{out}(\mathtt{B/liability} : \mathtt{account}, \mathtt{v} : \mathtt{value})\}$$

- The transfer `balance(P:agent,R:agent, r:claim)` stands for the removal of two reciprocal obligations, and is for the settlement of claim from Provider P to Receiver R. In that case, *r* is the transfer claim *x*, or a part of it, because it may be that the claim is settled in several steps. The balance transfer consists of the following set of postings, given here for transfer claim *r*, which may be referenced by a birth event identifier or a composite key. The (sub-) account is identified by a combination of the agent, the right type and the claim identifier.

$$\{\mathtt{out}(\mathtt{A/receivable\_claim/r} : \mathtt{account}, \mathtt{v} : \mathtt{value}),$$
$$\mathtt{out}(\mathtt{B/payable\_claim/r} : \mathtt{account}, \mathtt{v} : \mathtt{value})\}.$$

The transactions and transfers are illustrated below.

5.3. Example

Enterprise A contracts Counterparty B for a simple purchase order contract and assume an ERC20 implementation. We choose not to extend the ERC20 smart contract with business logic between B and A for the sake of simplicity and modularity. Two sets of rules of engagement (smart contracts) are needed: a parent contract, and an ERC20 compliant token contract to represent digitized resources for B and A. The parent contract is initiated through an (infological) transaction that creates the contract with all appropriate clauses (rules of engagement) to exchange goods or services, possibly including ground rules of the contract (defaults), like payment terms, notice periods etcetera. A and B both become owners of the contract. The initiating infological transaction (agreement) corresponds to the first COFRIS event — in the example of Table 1, this is the event with EID:21 that creates a number of transfer obligations. This is typically the case when the smart contract is instantiated from a template, which we consider to be the preferred option; but in principle, it is also possible to create an empty contract first and then create transfer obligations by means of transactions (like EID:21 in the example). This second option is possible because the contractual obligations are represented as digitized resources that are created or transferred by means of transactions; they are not built into the code. On the datalogical level an obligation token implies prefunding — the consequences in terms of token spending limits are not worked out in this paper.

Once the smart contract is accepted by both owners and stored on the blockchain, it can be effectuated/fulfilled through events that query or update the state of the contract on the SL. Transactions are recorded in the journal, optionally secured (see below). To have the maximum benefit of the SL, the visibility of the accounts to A and B and other parties should be flexible per account, so that some are private, some shared, some public. The parent smart contract instantiates or calls an ERC20 compliant contract for token transfer. Once it is confirmed that the goods/services have been exchanged (through a confirmation message from the customer or directly through IoT), the parent smart contract initiates a transaction to the ERC20 token contract to balance the obligation tokens to zero (as, in more detail, in example Table 1).

More specifically, consider COFRIS event with EID:22. Provider A delivers 5 widgets to Customer B, thereby fulfilling a transfer obligation as part of the contract obligation. The event maps to an infological "fulfillment" transaction with two transfers. The first one is a resource transfer (specified above)

$$\mathtt{resource\_transfer}(\mathtt{A}, \mathtt{B}, \mathtt{control}, < \mathtt{widget}, 5 >)$$

A and B are the agents, the right type is "control" and the object is "5 widgets". In terms of tokens, it means that a *DigitizedResource object token* standing for 5 widgets is transferred from (owenership) account of A to the (ownership) account of B.

The second transfer in the transaction is the performance fulfillment. A's contract obligation, for the transfer, is no longer open and B looses the correlated right. Here we get the infological transfer "fulfill" (see above):

$$\mathtt{fulfill}(\mathtt{A}, \mathtt{B}, \mathtt{obl21}, €100)$$

with agents A and B, the obligation obl21 and a value of € 100.

$$\{\mathtt{out}(\mathtt{A/transfer\_obligation/obl21}, €100),$$
$$\mathtt{in}(\mathtt{A/asset} : \mathtt{account}, €100),$$
$$\mathtt{out}(\mathtt{B/transfer\_right/obl21}, €100),$$
$$\mathtt{in}(\mathtt{B/liability}, €100)\}$$

The resource and fulfill transfers make up the consensual part. The rules of engagement should ensure that the exchange axioms are fulfilled, in particular, that every transfer is a fulfillment (and that the two match). Note that the matching may be specific–specific or specific–generic (as in the case that the obligation is to deliver "a" car of some type and the transfer is delivering a car with a specific chassis number).

Please cite this article as: H. Weigand, I. Blums and J.d. Kruijff, Shared Ledger Accounting — Implementing the Economic Exchange pattern, Information Systems (2019) 101437, https://doi.org/10.1016/j.is.2019.101437.

On the enterprise-specific part, according to the accounting rules, generally there is an additional posting (a) from contract asset account to "income" account at the side of A plus (b) an additional posting from "expenses" to finished goods. Finished goods is in fact a subaccount of the control account of A, so where the resource_transfer debits the latter, the posting *b* credits it by debiting the expenses (including transaction costs). Similarly, the contract asset received with the fulfillment is credited by posting *a* on the income account. On the side of counterparty B, we also have two additional postings (infological transfers). So in total, the COFRIS economic event "Transfer Fulfill" maps to an infological transaction consisting of *six* infological transfer events, where *two* are consensual.

The next economic event EID:23 first of all consists of another resource transfer, the setup services (labor), fulfilling the second performance obligation. The translation of this transfer is analogous to the EID:22 above. However, as all performance obligations are fulfilled now, a Realization transaction is triggered.

The realization means that Consideration Right is exchanged for Contract Claim, A's rights to receive value are received in exchange of the B's Obligations becoming unconditional — Receivable. This is mapped to an infological transfer that transfers the value from the "contract asset" account to "receivable" account. This is in the consensual part because the legal status of the Counterparty's obligation to pay has changed.

A design question is how this Realization transaction is generated. To keep the model transformation straight-forward, we keep the mapping of fulfillment events and realization events separate, but generate a *trigger rule* in the contract (the rules of engagement) that generates the Realization transaction automatically as soon as the performance obligations have been fulfilled. Similar for settlement. By means of the these rules, the third exchange axiom (Settlement) is implemented.

Although not all details are worked out, the example strongly suggests that the infological model is sufficient for being the target of a loss-less mapping from the essential (COFRIS) layer schema.

5.4. Implementation choices

The mapping from the infological level to the (datalogical) implementation is not 1–1. As with the physical realization of a logical data model in DBMS, there can be many alternatives that differ in speed, costs and security. For instance, there are two possible scopes for on-chain transaction processing for accounting systems [38]:

- *On-chain* This model assumes on-chain processing of consensus, settlement with an on-chain order book. Within the context of DLT, this is the most expensive option. Nevertheless, due to the fact that this is a single-entry solution, the individual stakeholders may avoid investments in compute, storage and security products.
- *Hybrid* This model assumes off-chain order relay with on-chain settlement. In this approach, cryptographically signed orders are broadcast off of the blockchain; an interested counterparty may inject one or more of these orders into a smart contract to execute trades trustlessly, directly on the blockchain.

Our smart contract based AIS assumes that every transaction is triggered and initiated by the smart contract based AIS through the ERC20 transaction functions and events. As a result, no 'standard' DLT transactions are allowed, where the agents transfer tokens directly to each other. It is possible to have ERC20 smart contract be orchestrated by an external smart contract based AIS and/or even a smart contract for zero-knowledge proof (see below). By doing so, the ERC20 token contract is kept original and only serves the purpose of conforming to uniform token standards, making it easier to update and align to best practices.

5.5. Benefits of smart contracts

A smart contract based AIS replaces the necessity of a trusted intermediary between the parties involved as in triple entry accounting, by a DLT-based consensus mechanism. Besides reducing the number of entries, this architecture introduces various side-benefits:

- **Less coordination.** The smart contract handles the initiation of transfers and the (internal and external) allocation of funds between accounts (e.g. goods payment and transportation). No longer are invoices need to be paid using conventional bank transfer; many transactions are dealt with by internal money transfers through the smart contract and at the edges of the corporation, formal and informal agents work to exchange between internal money and external money.
- **Increased privacy**. While parties may employ pseudonyms to enhance their privacy, research shows that anyone can de-anonymize DLT transactions by using information in the blockchain, like transaction structure, value, time and date [38]. Zero-knowledge proofs (e.g. Zero-Knowledge Succinct Non-Interactive Argument of Knowledge, zkSNARKS) allow one party (the prover) to prove to another (the verifier) that a statement is true, without revealing any information beyond the validity of the statement itself. By imposing zero-knowledge technology to a smart contract based AIS, participants do not have to share privacy sensitive information, like their private key, resource allocation strategy or transaction size, since the transactions on the blockchain are encrypted [17]. Today, zero knowledge proofs in combination with smart contracts is the future but is still in its early days and has many challenges to overcome to increase practicality, like computational intensity, setup phase and costs.

Similar to triple entry accounting's Shared Transaction Repository, the smart contract based AIS uses a shared environment to store the entry which is the transaction. By applying zero-knowledge proof consensus, the transaction is encrypted on the blockchain, making it impossible for non-participants of the smart contract to de-anonymize these transactions. Due to this encryption, it becomes irrelevant (from the point of view of confidentiality) whether a public or a private blockchain is used, giving participants options with regards to costs and extent of neutrality.

6. Conclusion

Shared ledger systems built on DLT may have a high impact on the current AIS, not only because of the claimed immutability of the records but also because of the shift from an internal actor-dependent to an external consensus view. So far, a worked out design for such a shared ledger system was missing. In this paper, we have taken an ontological approach, focusing on the economic exchange pattern. While COFRIS has been developed in the last few years in several workshop papers, this article gives the first full account of the consensuality and correlativity quality aspects, and so presents the first worked-out SL exchange ontology. Explicit attention has been given to the question what is to be shared in the shared ledger and what not, and how the two parts can be related in a rigid way. Where there are concerns that triple-entry accounting "may not be advanced enough" [3, p18], the paper aims to contribute to a foundation that is both ontologically sound and fully compliant with the accounting standards. Although compliant, the proposed Shared Ledger based on the economic exchange pattern is also a new concept that has potential to revolutionize the accounting field. Spreading of the SL technology would change the focus of financial reporting from enterprise-centric towards an exchange-centric model that will greatly improve quality and trustfulness. The possibility of consensual and correlative activity recording is an alternative for what in conventional accounting is achieved through an expensive audit process.

We have also described how this conceptual model can be realized at the platform-independent (infological) level, by using smart contracts and the ECR20 token standard. The smart contract does not only have the advantage of automated execution (that is, delegated fulfillment of commitments), but also provides an aggregation level close to that of the economic exchange contract. A limitation must be added: our infological model assumes a technical environment to be in place that is still in progress. The model may help to steer its direction. The infological model abstracts from current blockchain implementations. It is shown to be sufficient for SL implementability. However, it is also general enough for other DLT applications, such as supply chain management, where the accounts correspond to logistic nodes and the transactions to logistic movements.

DLT platforms are evolving rapidly now. For that reason, we have focused on a platform-independent model, and not on the coding, although we are also experimenting with the PIM to PSM level transformation [39]. We are planning to bring these efforts together in the future. Other research topics include public reporting directly based on the blockchain, the impact of DLT-based AIS on the auditing task, and the further development of a declarative language for commitment-based smart contracts.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Tapscott D., A. Tapscott, Blockchain revolution. Portfolio Penguin, New York (2016).
[2] J. Mendling, et al., Blockchains for business process management – Challenges and opportunities, ACM Trans. Manage. Inf. Syst. 9 (1) (2018) 1–16.
[3] J. Dai, M. Vasarhelyi, Toward blockchain-based accounting and assurance, J. Inf. Syst. 31 (3) (2017) 5–21.
[4] J.G. Coyne, A. McMickle, Can blockchains serve an accounting purpose?, J. Emerg. Technol. Account. 14 (2) (2017) 101–111.
[5] I. Grigg, Triple entry Accounting. Systemics Inc. http://iang.org/papers/triple_entry.html (2005).
[6] J. Kruijff, H. de Weigand, Understanding the blockchain using enterprise ontology, in: E. Dubois, K. Pohl (Eds.), Proc. CAISE 29, Springer, 2017, pp. 29–43.
[7] J. Dietz, Enterprise Ontology, Springer, 2006.
[8] ISO/IEC. Information Technology — Business Operational View — Part 4: Business Transactions Scenarios — Accounting and Economic Ontology. ISO/IEC FDIS 15944-4 (2015).
[9] W.E. McCarthy, The REA accounting model: A generalized framework for accounting systems in a shared data environment, Account. Rev. (1982) 544–577.
[10] I. Blums, H. Weigand, Towards a Reference Ontology of Complex Economic Exchanges for Accounting Information Systems. In: Proc. IEEE EDOC 20: 119-128 (2016).
[11] I. Blums, H. Weigand, Financial reporting by a Shared Ledger. In: Proc. JOWO (2017).
[12] IASB Conceptual Framework for Financial Reporting, IASB (2018).
[13] IASB homepage, http://www.ifrs.org/issued-st{and}ards/list-of-st{and}ards, IASB (2019).
[14] A. Hevner, S. March, J. Park, Design research in information systems research, MIS Q. 28 (1) (2004) 75–105.
[15] J.P.A. Almeida, G. Guizzardi, An ontological analysis of the notion of community in the RM-ODP enterprise language, Comput. Stand. Interfaces 35 (3) (2013) 257–268, http://dx.doi.org/10.1016/j.csi.2012.01.007, <http://linkinghub.elsevier.com/retrieve/pii/S0920548912000402>.
[16] D. Appelbaum, R. Nehmer, Designing and Auditing Accounting Systems Based on Blockchain and Distributed Ledger Principles, Feliciano School of Business, 2017.
[17] Y. Wang, A. Kogan, Designing Privacy-Preserving Blockchain Based Accounting Information Systems http://dx.doi.org/10.2139/ssrn.2978281 (2017).
[18] J. Kokina, R. Mancha, D. Pachamanova, Blockchain: Emergent industry adoption and implications for accounting, J. Emerg. Technol. Account. 14 (2) (2017) 91–100.
[19] Fibonacci's Liber Abaci A Translation into Modern English of Leonardo Pisano's Book of Calculation, Sigler, Laurence, Springer-Verlag, New York (2002).
[20] Y. Ijiri, Theory of Accounting Measurement, American Accounting Association, 1975.
[21] I. Blums, H. Weigand, A. Blums, Conceptualizing resources and claims in consensual economic exchanges, in: L. Jansen (Ed.), Proceedings of the Joint Ontology Workshops 2018: Episode IV: The South African Spring Vol. 2205, CEUR, Trier, 2018, pp. 1–15.
[22] I. Blums, H. Weigand, Towards a core ontology for financial reporting information systems (COFRIS). C. Debruyne, others, (Eds.) OTM 2017 Workshops, LNCS 10697 (2018) 302–306.
[23] G. Guizzardi, Ontological Foundations for Structural Conceptual Models, Telematics Instituut Fundamental Research Series, No. 015, ISSN 1388–1795, The Netherlands (2005).
[24] J. Guerson, et al., OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies. In: Proc. IEEE EDOCW 2015, 144-147 (2015).
[25] G. Guizzardi, et al., Endurant types in ontology-driven conceptual modeling: Towards ontouml 2.0, in: Proc. International Conference on Conceptual Modeling, Springer, 2018, pp. 136–150.
[26] B.B. Duarte, et al., Ontological foundations for software requirements with a focus on requirements at runtime, Appl. Ontol. 13 (2) (2018) 73–105.
[27] J. Nardi, et al., Towards a Commitment-based Reference Ontology for Services. In: Proc. IEEE EDOC 17, (2013) 175-184.
[28] B. Criffo, J.A.A. Almeida, G. Guizzardi, From an Ontology of Service Contracts to Contract Modeling in Enterprise Architecture. In: IEEE EDOC 21 (2017).
[29] V.A. Carvalho, J.P.A. Almeida, C. Fonseca, G. Guizzardi, Multi-level ontology-based conceptual modeling, Data Knowl. Eng. 109 (2017) 3–24.
[30] C.L. Azevedo, M.E. Iacob, J.P.A. Almeida, M. van Sinderen, L.F. Pires, G. Guizzardi, Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for ArchiMate, Inf. Syst. 54 (2015) 235–262, http://dx.doi.org/10.1016/j.is.2015.04.008i.
[31] FASB Exposure Draft. Statement 8 – Conceptual Framework for Financial Reporting, 2016.
[32] The Open Group, ArchiMate 3.0.1 Specification. Standard C179 (2017).
[33] N. Guarino, G. Guizzardi, Relationships and Events: Towards a General Theory of Reification and Truthmaking, AI*IA, 2016, pp. 237–249.
[34] W.M.P. van der Aalst, Process discovery from event data: Relating models and logs through abstractions. wiley interdisc, Rew. Data Min. Knowl. Discov. 8 (3) (2018) http://dx.doi.org/10.1002/widm.1244.
[35] V. Buterin, A Next Generation Smart Contract & Decentralized Application Platform (2014) Retrieved from https://github.com/ethereum/wiki/wiki/White-Paper.
[36] W. Warren, A. Bandeali, 0x: An open protocol for decentralized exchange on the Ethereum blockchain, https://0xproject.com/pdfs/0x_white_paper.pdf (2017).
[37] S. Popejoy, The Pact Smart-Contract Language, Whitepaper, http://kadena.io (2017).
[38] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, Zerocash - Decentralized Anonymous Payments from Bitcoin. Security and Privacy (SP). In: Proc. 2014 IEEE Symposium on Security and Privacy, 459-474 (2014).
[39] H. Syahputra, H. Weigand, The Development of Smart Contracts for Heterogeneous Blockchains, in: K. Popplewell, et al. (Eds.), Enterprise Interoperability VIII. Proceedings of the I-ESA Conferences, vol. 9, Springer, 2019.