The REA Model Expressed in a Generic DEMO Model for Co-creation and Co-production

Frantisek Hunka[1(✉)] and Steven J.H. van Kervel[2]

[1] University of Ostrava, Ostrava, Czech Republic
Frantisek.hunka@osu.cz
[2] Formetis Consultants BV, Boxtel, The Netherlands
info@formetis.nl

Abstract. The REA ontology is a domain ontology that aims to support accounting information systems that must provide a truthful and appropriate – GAAP compliant - descriptive perspective of an enterprise in operation. While the application of a domain ontology provides strong benefits, the current representation of the REA model does not provide the desired results; an appropriate working accounting system. One of the root causes of this problem is the lack of a proper formal representation of the REA model. In this paper the DEMO methodology is applied to provide a generic domain and application-independent DEMO model (the CC-CP model) for co-creation and co-production in any industrial production chain. This model appears to be also appropriate to capture any interaction between an enterprise and any external parties, stakeholders, customers, suppliers, personnel etc., and support accounting systems. This approach offers several new advantages, notably: (i) prescriptive workflow-like operation of the enterprise with full transaction driven execution; (ii) process-mining (-like) analysis of daily operation; (iii) ontological completeness of factual knowledge as required not only for accounting systems but also for other descriptive information systems and (iv) completeness of implementation for any kind of business interactions between enterprises.

Keywords: REA model · DEMO enterprise ontology (DEO) · DEMO methodology · Co-creation and Co-production

1 Introduction

REA is considered a strong potential improvement of foundations for accounting systems. It aims at providing a domain ontology which is a necessary condition of any system that provides some perspective of “phenomena in the real world”. However, so far any attempts to apply REA for accounting systems using the current representation [8, 9] have encountered serious problems that are difficult to identify and even more difficult to fix [11, 12]. Its present formal representations appear not appropriate enough [12]. It is argued and observed that the root of these problems is obviously caused by: (i) a lack of good ontological foundations; (ii) lack of good formal methods and (iii) lack of an appropriate formal language to represent the model.

D. Aveiro et al. (Eds.): EEWC 2017, LNBIP 284, pp. 151–165, 2017.
DOI: 10.1007/978-3-319-57955-9_12

For the study and understanding of phenomena observed in reality, ontologies are designed and used to understand and reason about some specific domain of reality (reality "precedes" ontology). Section 5 specifies REA model-driven GAAP compliant systems and the core theories. An ontology is defined as a "formal, explicit specification of a shared conceptualization of reality" (Gruber 1993). The conceptualization is composed of concepts of objects, their attributes and their relations. The conceptualization is shared by knowledgeable human stakeholders, meaning in the first place that all stakeholders are assumed to have an identical understanding of each object or concept, each attribute and each relation. Informally, an ontology has two faces; one face to the phenomena in the real world, and the other face a formal explicit representation of concepts.

The REA ontology must capture all phenomena in the real world that are required by some accounting system to provide a GAAP compliant accounting perspective and representation of that enterprise. GAAP requirements define which phenomena must be captured with a completeness criterion. If not all phenomena are captured or if the phenomena are not captured in a truthful way, then any accounting system will provide a wrong perspective, a wrong profit -loss statement, general ledger etc.

In this paper a representation of the REA model expressed in a DEMO model for co-creation and co-production (CC-CP) [12] is assessed. The CC-CP model is generic, application and industry independent and captures all relations between suppliers, stakeholders and individual workers for our enterprise of interest. This claim demands future empirical support.

In Sect. 2 the ontological foundations of REA and DEMO are assessed. In Sect. 2.1 the strength and weaknesses of the REA ontology are described and why another representation of the REA model is needed. Notably the problematic notion of "value" has been addressed and solved [3]. In Sect. 3 the CC-CP model is presented and assessed. Section 4 describes the benefits of the CC-CP model representation of the REA model and several quality and completeness criteria.

Future research, our long term strategic vision and objectives are described in Sect. 5. If we have a proper formal representation of the REA model that is provided by the DEMO CC-CP model, then that is a promising foundation for future model driven GAAP compliant accounting systems, with strong benefits.

2 Ontological Foundations of REA and DEMO

The ontological foundations, the strength and weaknesses of REA are assessed. Its apparent weaknesses are notably lack of formal methods, lack of empirical theories and ontological flaws and incompleteness [14]. The strengths of the DEMO Enterprise Ontology (DEO) are described; notably strong empirical foundations, formal methods, design science and general systems theory. This approach is to express the REA ontology in concepts and relations provided by the DEO in such a way that strength and value of REA is kept and the apparent weaknesses are mitigated.

2.1 REA Ontology

The REA ontology originates from accountancy systems and provides a domain specific platform for value modeling business processes, see [10]. The principal economic concepts are economic resources, economic events and economic agents. Economic resources are things of economic value that have utility for economic agents and for this reason they used to be planned, monitored, and controlled. Economic events are activities within an enterprise that represent either an increment or a decrement in the value of economic resources. Economic agents are individuals or organizations that participate in the control and execution of economic events.

The other fundamental REA concepts, their relationships, constraints and rules for constructing application models are illustrated in Fig. 1. Apart from the above mentioned concepts, Fig. 1 also contains commitment and contract concepts and corresponding relationships.

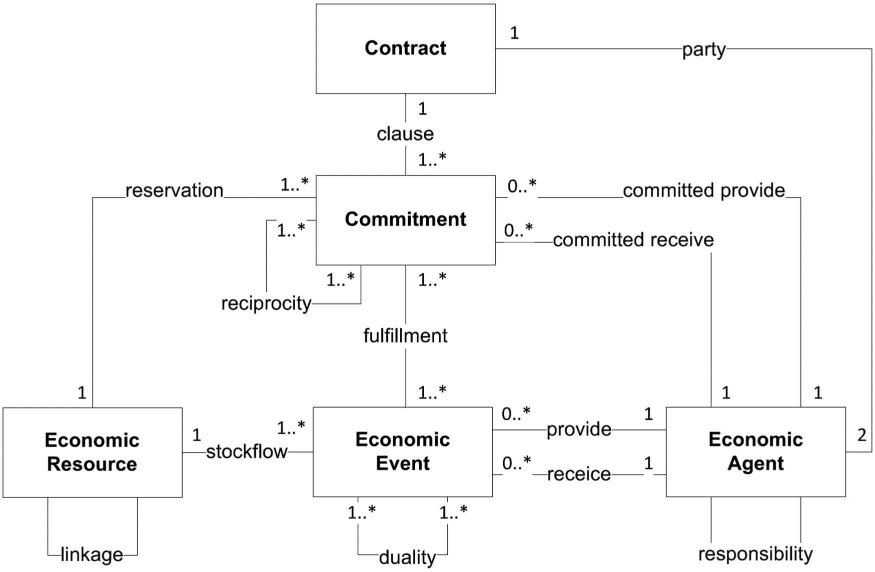

Fig. 1. REA metamodel level. Adopted from [9]

The main benefit of the REA approach is that all accounting artifacts such as debit, credit, journals, ledgers, receivables, and account balances are derived from the data describing exchange and conversion REA processes. It means that all accounting artifacts are always consistent, because they are derived from the same data; for example, data describing a sale event is used in warehouse management, payroll, distribution, finance and other application areas, without transformation or adjustment [9].

REA anomalies have its origin in absence, to some degree, of rigorous theoretical and philosophic foundations, mainly since these seem to be lacking. One of these lacking capabilities is that the REA model itself does not have specific states from which a state machine can be derived. Observation shows that in REA between two entities there are different discrete, disjoint, states of any economic transaction. Instead, only the resource states are identified and frequently used as the states of the state machine, which is incorrect. As a result of this, the REA model cannot provide decline and reject transaction steps, as well as revoking operations, things one observes happening in any economic transaction. In addition, the REA model is predominantly design to capture events that refer to the resource value or resource feature. The other events such as business events or information events are difficult to capture and further processed. Consequently, REA has not properly defined so called information or knowledge entities such as contract or schedule because these entities are not resources. REA contains a contract concept which represents a contract that came into effect. There is no concept in REA for unsigned (not coming into effect) contract. The REA modeling approach aims at descriptive information systems that are based on exchange, consumption, usage and production of economic resources.

To summarize the main observed limitations and flaws of REA in its current representation.

1. Restriction to capture only production facts that in addition refers only to the exchange of property rights to resources or to transformation of a set of resources into another set of resources.
2. Lack of ontological completeness of a *transaction* for example: sending a production order, receiving an invoice etc. These are all important facts for accounting systems. Yet not provided by the current representation [9].
3. Not capturing *transaction* events which imply that there is no truthful state machine based on transaction states and state transitions. This also includes missing transaction steps such as decline or reject, as well as cancellation patterns.
4. Conceptual mismatch – not capturing the real phenomena in the world. It is manifested in explicit distinction between past and current events and events which are performed in future and in impossibility to express the change of state in which a contract or a schedule comes into effect.
5. Restricted ability to express explicitly business rules. The type level mechanism applied in REA enables only to impose business rules on the instances which are in compliance with the given type.
6. No prescriptive capabilities – no certainty that “in real life” things go as defined.
7. REA and accounting apply the notion of value, but there are serious problems associated to this concept. The main problem is that value is subjective and does not exist in the real world. Discussed in detail in [3, 12].

Yet, there are valuable aspects of the REA model. The approach in this paper is to represent REA in a better way.

2.2 DEMO Enterprise Ontology

DEMO is an engineering methodology to derive conceptual models of enterprises, based on an ontological theory, DEMO enterprise ontology (DEO) [1, 2]. DEO is comprised of four axioms and a theorem. DEMO is part of the emerging discipline of ‘enterprise engineering’ (EE) [2]. EE is founded on the same kind of theories as more mature engineering disciplines such as civil engineering, aviation and electronics. A claim for the quality of the applied methodology is guaranteed by the underlying theories, methodologies, formal methods [2, 5, 7] and a good body of empirical cases in many domains [5].

The DEMO methodology claims to provide models that meet the so-called C4-ness quality criteria [7]. Comprehensiveness refers to the condition that the model should encompass everything that is part of the ontology. This includes all concepts and relations of the ontology; nothing is missing.

Consistency refers to the absence of any anomalies of any kind. Conciseness refers to the requirement that anything that is not in the domain of the ontology should not be represented in any model. Coherence refers to the 'semantic meaningfulness of the symbols and their relations from every perspective'.

Specific results of C4-ness qualities are (i) that any enterprise that may exist in the real world, including virtual CC-CP enterprises, can be modeled correctly in one and only one way; and (ii) the DEMO model(s) for any such enterprise must provide concise and comprehensive factual knowledge about the operation of the enterprise. These two claimed results must be empirically tested for co-creation and co-production (Sect. 3). It is assumed, to be proven by validation and assessment, that expressing the REA ontology in DEMO may provide a DEMO model that is truthful and appropriate to represent the REA model and support accounting systems well.

3 The CC-CP Model

The purpose of the proposed CC-CP DEMO model [12, 13] is to be a generic specification of any financial or business interaction or transaction between our enterprise of interest and any external stakeholders such as customers, suppliers, personnel staff and taxation or other governmental institutions. In execution of that enterprise model factual knowledge must be provided for information systems. This model is claimed to capture any interactions between an enterprise and any stakeholder. It is a *generic pattern of interaction,* equivalent to the DEMO transaction.

3.1 Co-creation and Co-production Between an Enterprise and Its Stakeholders

Many highly specialized enterprises 'Contractors' do not have a well-defined portfolio of products with fixed prices but offer their capabilities to meet the specific requirements of their Principals. We define: co-creation captures the principal and the contractor(s) working together on the engineering of an acceptable artifact; co-production captures the shared production of the engineering artifact by both principal and contractor(s), including matching financial transactions.

In this paper, the original scope of co-creation and co-production has been extended to any stakeholder that interacts with our enterprise of interest; including customers, sub-contractors, suppliers, workers, tax offices etc.

It is assumed, but not proven and future research, that all – not only accounting systems – information systems that provide some appropriate perspective of the enterprise must be supported. The CC-CP fact model must specify all possible facts for any descriptive IS. To prove this completeness claim is future research.

3.2 Ontological Completeness Quality Criteria of the CC-CP Model

There are several mandatory quality and completeness quality criteria applicable. Missing criteria 1 or 2 for only one case renders the model worthless.

1. Completeness of the CC-CP model to capture any business interactions with any imaginable stakeholder in a truthful and appropriate way.
2. Completeness of the CC-CP model to capture any factual knowledge (Sect. 3.2) that maps to any REA concept for accounting systems.
3. Completeness of the CC-CP model to capture all factual knowledge that may be needed by any GAAP compliant accounting system. This is a wider quality criterion than requirement 2. This quality requirement is desirable but demands assessment of the common foundations of GAAP compliant accounting systems, which is future research.

3.3 The CC-CP Factual Information Support for Accounting Systems

The model must provide factual knowledge (informally "data about events") about the operation of the enterprise of interest for any imaginable information system – an Enterprise Information System or "EIS" - that provides some descriptive perspective of the operation of this enterprise. Key notion is that "facts", which are propositions about the world of phenomena, provide all required information for the descriptive accounting information systems.

The term factual knowledge refers to truthful propositions about phenomena in the world, In our case some phenomena in or about our enterprise of interest. In the FAR ontology [13] is specified that a fact is a proposition that may have a logic relation with other facts in a recursive way. A fact is a proposition that may have three values; true | false | undefined. While the meaning of the values true and false are clear, the value of "undefined" reflects the situation that for some unknown reason factual information is not available. In the FAR ontology there exist four kinds of facts:

1. Communicative facts; as defined by the DEMO transaction axiom.
2. Infologic and datalogic production facts. An example is the text of the contract of the CC-CP model. It is precisely the 'text only', without any actor commitments.
3. Facts about the world of phenomena not captured by the DEMO ontology, the kinds 1 and 2. Example: the exchange rate dollar – euro = 0.85. The value of this proposition can be true | false | undefined.
4. Any logic aggregated facts, or dependent facts, composed of logic relations (AND | OR | NOT relations) of other facts. Evaluation laws for the three-state logic.

The notion "full factual knowledge" is important. There are three completeness claims; (i) all interactions with any stakeholder with which there are transactions (financial or otherwise) must be captured well; (ii) for each interaction with a stakeholder all relevant facts must be provided; (iii) in addition there is the requirement that for each fact all relevant attributes of that fact must be provided.

Example: the fact represented by proposition "Person a is member of Club c" can be defined to be true if: The age of the proposed member is above 18 years of age; AND the membership admittance procedure has been approved; AND the membership fee has been paid. If these requirements have not all been met then the fact is not true. If any of the composing facts cannot be evaluated and returns the value undefined then the value of the fact becomes undefined. Relevant attributes of that fact may be the date when the membership became true, the duration of the membership. There are also relations to the person that is a member etc.

To summarize, the following propositions are formulated:

1. The CC-CP model captures any transaction based on business interactions betweenour enterprise of interest and any stakeholders, suppliers, subcontractors, staff etc.
2. Capturing of the business interaction between the enterprise of interest and stakeholders implies that CC-CP model provides a truthful and appropriate representation of all DEMO transactions.
3. For the enterprise of interest there exist a number of valuable descriptive perspectives – seen from the perspective of the stakeholders, shareholders, management etc. - of the operation of that enterprise. The relevant perspective of this paper is an accounting perspective provided by an EIS (enterprise information system), a REA compatible GAAP compliant accounting system.
4. There is a completeness claim for factual knowledge claims; all facts, and facts are complete with all attributes.

3.4 The CC-CP Fact Model

The proposed CC-CP Fact Model strictly follows the CC-CP Construction Model which is composed of six transactions, see Fig. 2. The presented Fact Model is described in three phases which correspond to the phases of the Construction Model and is illustrated in Fig. 3.

The co-creation phase includes T-1 and T-2 transactions. The object class CONTRACT which is the core concept in the whole CC-CP Fact Model is identified in this phase. The other object classes that are identified in this phase are PRODUCTION, PRICE, PRODUCTION-KIND, MONEY-KIND and the external object class ENTERPRISE. All mentioned object classes are primal classes, which means that they cannot be defined on the basis of other fact types. The lines between CONTRACT and ENTERPRISE labeled "principal of contract is enterprise" and "contractor of contract is enterprise" represent property types. Mandatory and uniqueness constraints indicate that a contract must have one enterprise as a principal and one different enterprise as a contractor.

The property type between the object classes CONTRACT and PRODUCTION indicates that each contract has only one production and each production has only one contract. The same holds for the property type between the object classes CONTRACT and PRICE.

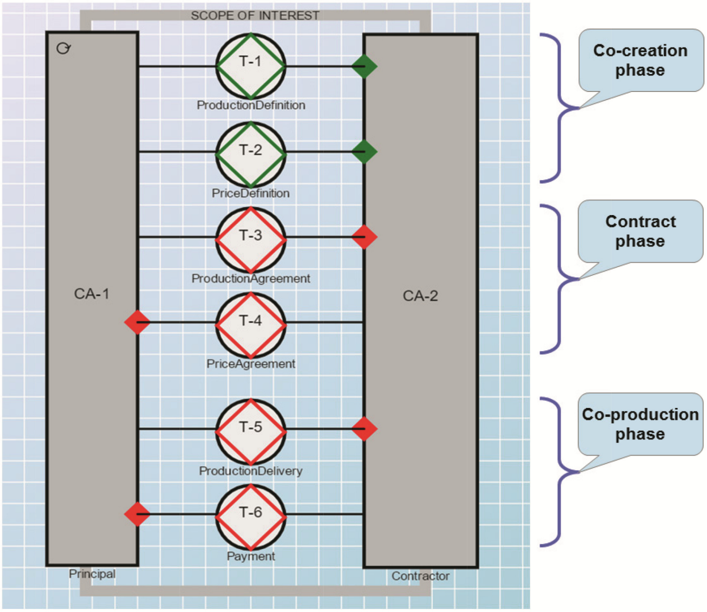

Fig. 2. The CC-CP Construction Model

The property type between the object classes PRODUCTION and PRODUCT-KIND expresses that one product can include more product-kinds which is in compliance with a purchase order containing more items. Each product-kind is further specified by value types which represent the volume (amount), the price per unit and the delivery day of the product-kind. The result kind "[production] was defined" is existentially independent unary fact kind which is the result of T-1 transaction.

The property type between the object classes PRICE and MONEY-KIND indicates that one price can have several money-kinds. Each money-kind is further specified by value types which represent the price of production and the day of payment. The result kind "[price] was defined" is existentially independent unary fact kind which is the result of T-2 transaction. From the implementation point of view it is supposed that T-1 transaction kind and T-2 transaction kind have each only one instance.

The contract phase includes T-3 and T-4 transactions. The result kind CONTRACT_SIGNED is another core concept and is a subclass of the object class

CONTRACT. The figure illustrates that any contract can become a contract signed. This result kind becomes existent when T-3 is Promised and T-4 is Promised. From the above follows that in order to model a contract signing it is necessary to explicitly express two coordination facts and perform a logical aggregate over them. As the traditional DEMO methodology does not cope with this requirement, the FAR ontology was utilized to capture the above described task. From the implementation point of view it is supposed that T-3 transaction kind and T-4 transaction kind have each only one instance.

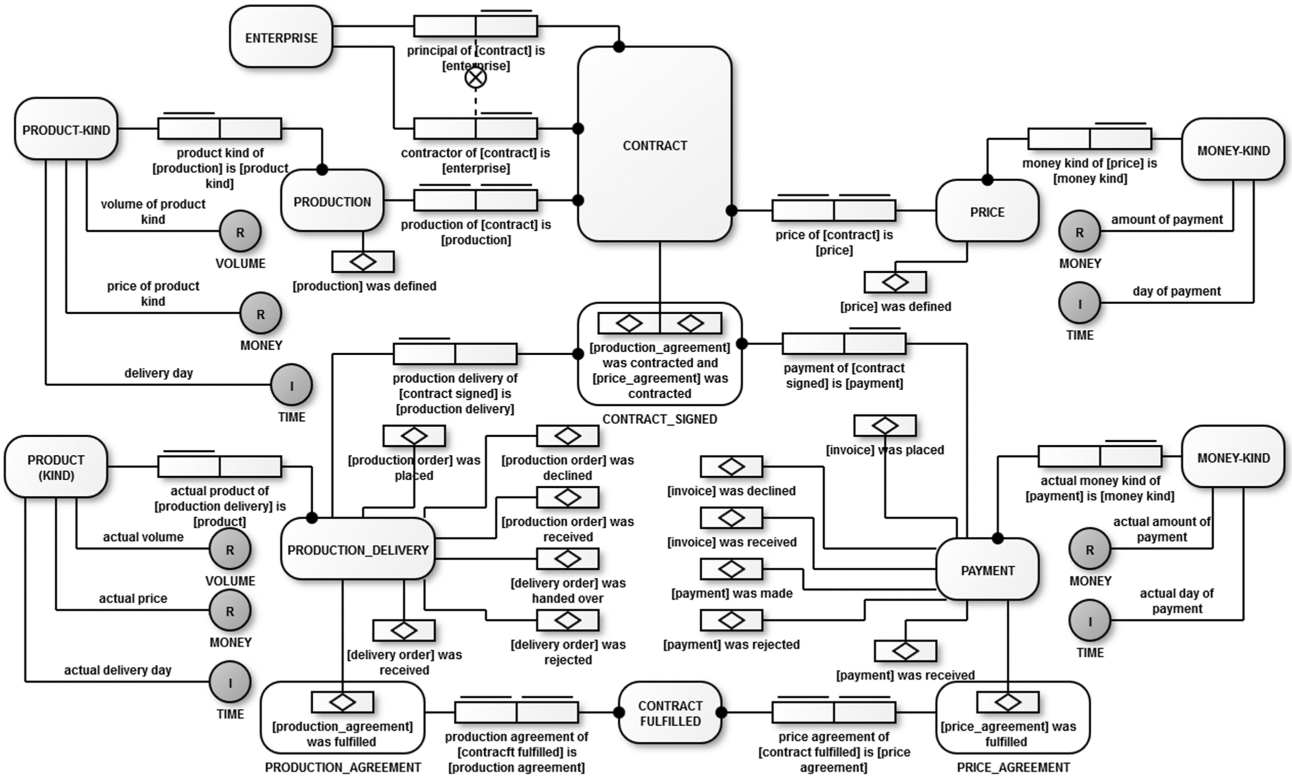
ENTERPRISE
principal of [contract] is [enterprise]
contractor of [contract] is [enterprise]
production of [contract] is [production]
CONTRACT
PRODUCT-KIND
product kind of [production] is [product kind]
PRODUCTION
volume of product kind
R
VOLUME
[production] was defined
price of product kind
R
MONEY
delivery day
I
TIME
price of [contract] is [price]
PRICE
money kind of [price] is [money kind]
MONEY-KIND
R
MONEY
amount of payment
[price] was defined
day of payment
I
TIME
[production_agreement] was contracted and [price_agreement] was contracted
CONTRACT_SIGNED
payment of [contract signed] is [payment]
production delivery of [contract signed] is [production delivery]
PRODUCT (KIND)
actual product of [production delivery] is [product]
actual volume
R
VOLUME
actual price
R
MONEY
actual delivery day
I
TIME
PRODUCTION_DELIVERY
[production order] was placed
[production order] was declined
[production order] was received
[delivery order] was handed over
[delivery order] was rejected
[delivery order] was received
[invoice] was placed
[invoice] was declined
[invoice] was received
[payment] was made
[payment] was rejected
[payment] was received
PAYMENT
actual money kind of [payment] is [money kind]
MONEY-KIND
actual amount of payment
R
MONEY
actual day of payment
I
TIME
[production_agreement] was fulfilled
PRODUCTION_AGREEMENT
production agreement of [contracft fulfilled] is [production agreement]
CONTRACT FULFILLED
price agreement of [contract fulfilled] is [price agreement]
[price_agreement] was fulfilled
PRICE_AGREEMENT

Fig. 3. The CC-CP Fact Model

The co-production phase is formed by T-5 and T-6 transactions and the execution and result phases of T-3 and T-4 transactions. The property type between the object classes CONTRACT_SIGNED and PRODUCTION_DELIVERY indicates that one contract_signed can have more production deliveries which is in compliance with the modeling reality. From the implementation point of view it is supposed that T-5 transaction can have one or more instances.

The property type between the object classes PRODUCTION_DELIVERY and PRODUCT (KIND) expresses that one production_delivery can have more products. Each product is further specified by value types which represent the actual volume, the actual price per unit and the actual delivery day. A product, as such, can be identifiable or quantifiable or both. In case a product is identifiable, it can have a serial number and the notion of product can be used. If the product is only quantifiable the notion of product kind is used. The result kind "[production] delivery" is existentially independent unary fact kind, which is the result of T-5 transaction.

The property type between the object classes PAYMENT and MONEY-KIND indicates that one payment can represents more money-kind which is in compliance with a payment order containing more money kinds. Each MONEY-KIND is further specified by value types which represent the actual price of production and the actual day of payment. The result kind "[payment] was made" is existentially independent unary fact which is the result of T-6 transaction.

The result kind PRODUCTION_AGREEMENT is a subclass of the object class PRODUCTION_DELIVERY. It means that the object class PRODUCTION DELIVERY can become the result kind PRODUCTION_AGREEMENT when the fact "production_agreement was fulfilled" becomes existent. The result kind PRICE_AGREEMENT is a subclass of the object class PAYMENT. It means that the object class PAYMENT can become the result kind PRICE_AGREEMENT when the fact "price_agreement was fulfilled" comes into existence.

The object class CONTRACT_FULFILLED becomes existent when the result types PRODUCTION_AGREEMENT and PRICE_AGREEMENT come into existence. The meaning of this object class is that obligations concerning production_agreement and price_agreement as declared in the object class CONTRACT_SIGNED were fulfilled and the contract is completed. The object class CONTRACT_FULFILLED represents the duality relationship in REA.

3.5 Conceptual Mapping of the CC-CP FACT Model to REA Model Concepts

Despite the fact that the conceptual mapping is rather simple and needs further rigorous elaboration, it captures the core issue. The DEMO Bank Contents Table, which is shown in Table 1, contains object classes, fact types, and transaction banks, in which their instances are contained.

Table 1. The Bank Contents Table of the CC-CP Model

Bank	Independent/Dependent fact
T1	CONTRACT
	the principal **of** Contract
	the contractor **of** Contract
	ENTERPRISE
	the production **of** Contract
	PRODUCTION
	the product-kind of Production
	PRODUCT_KIND
	the volume **of** Product-Kind
	the price **of** Product-Kind
	the delivery day **of** Product-Kind
	the production **of** Contract is defined P1
T2	PRICE
	the price **of** Contract
	MONEY_KIND
	the amount **of** payment **of** Money-Kind
	the day **of** payment **of** Money-Kind
	the price **of** Contract **is** defined P2
T3	**the** production_agreement **is** fulfilled P3
T4	**the** price_agreement **is** fulfilled P4
T3, T4	**the** production_agreement **is** promised **and** **the** price_agreement **is** promised
	the production requisition
	CONTRACT_FULFILLED
	the production_agreement **is** fulfilled **and** **the** price_agreement **is** fulfilled P3 and P4
T5	PRODUCTION_DELIVERY
	the production delivery **of** Contract_Signed
	the product **of** Production_Delivery
	PRODUCT
	the actual volume **of** Product
	the actual price **of** Product (price per unit)
	the actual delivery day **of** Product
	the production order placed (sent) T5.rq
	the production order declined T5.dc
	the production order received T5.pm
	the delivery order handed over T5.st
	the delivery order receipt T5.ac (P5)
	the delivery order rejected T5.rj
T6	PAYMENT
	the payment **of** Contract_Signed
	the money-kind **of** Payment
	MONEY_KIND
	the actual amount **of** payment **of** Money-Kind
	the actual day **of** payment **of** Money-Kind
	the invoice placed (sent) T6.rq
	the invoice declined T6.dc
	the invoice received T6.pm
	the payment made (sent) T6.st
	the payment receipt T6.ac (P6)
	the payment rejected – dispute T6.rj

The coordination fact "the delivery order receipt" means that the production was not only delivered but was also accepted by the principal. At this time, the corresponding production fact comes into existence. The same holds for the coordination fact "the payment receipt" which means that the payment was not only sent by the principal but was also accepted by the contractor. At the same time the corresponding production fact becomes existent.

The conceptual mapping deals with the DEMO CC-CP model fact kinds and their mapping to REA concepts and relationships as follows. The production fact "the production of Contract is defined" which becomes existent as a result of T-1 transaction contains all dependent facts (property types, attributes types) that are needed for one kind (decrement/increment) of an REA commitment. "The price of Contract is defined" is the next production fact which comes into existence as a result of T-2 transaction. The T-2 transaction instance contains all dependent facts (property types and attribute types) that are needed for one kind of an REA commitment. The aggregate coordination fact "the production of Contract is promised and the price of Contract is promised" is mapped into the reciprocity relationship that relates a different kinds of commitments to each other. The commitments are related to the corresponding resource types and economic agent types. "The production requisition" is a dependent fact type, which is mapped into the reservation relationship in the REA model.

The number of instances of the T-5 and T-6 transaction types corresponds to a number of production deliveries and a number of installments, respectively. The T-5 transaction instance captures one production delivery, which is in compliance with reality. The independent production fact "the delivery order receipt" is accompanied by the dependent facts of the property types and attributes types. From the accounting perspective the most important are explicitly expressed coordination facts that capture the necessary inventory system events. The CC-CP model is able to register all these events.

The T-6 transaction instance captures one payment (installment) in compliance with reality. The independent production fact "the payment receipt" is accompanied by dependent facts of the property types and attributes types. From the accounting perspective, explicitly expressed coordination facts that capture the necessary accounting system events are the most important. These events are: *sending an invoice*, *receiving an invoice*, *making a payment*. The T-5 and T-6 transaction instances can provide coordination facts of decline and reject. Their practical meaning is as follows. "The production order was declined" or "the delivery order was rejected" in case of the T-5 transaction instance, and "the invoice was declined" or "the payment was rejected" in case of the T-6 transaction instance.

To summarize the following results are found. From the above simple analysis follows that the CC-CP model provides all the facts that are necessary for the REA exchange model. The decline and reject coordination facts have no equivalent in the REA model but have equivalents in reality. In addition, the CC-CP model captures more precisely and truthfully the facts that pertain to the signing of a contract and the facts that concern the fulfilling of a contract.

Based on these simple and not rigorous assessments it is claimed that the DEMO CC-CP model fully captures the facts needed by the REA model.

4 Benefits of the REA Model Represented by the DEMO CC-CP Model

The following benefits are provided by the CC-CP model and DEMO [1]:

1. The CC-CP model is extensible without loss of its capabilities. Supporting transactions can be added to provide more control of the enterprise operation. Example: An employee is permitted to send a quotation for an order to a customer, a legally binding commitment, but must have first an approval from a colleague. This is an imposed business rule that must be enforced. To model this correctly, a transaction must be created between the employee and the colleague. The production fact of that transaction is an approval or a rejection. A business rules inhibits the c-act to send the quotation until that permission Pfact becomes true – approved. If the Pfact is rejected it will be never possible to send the quotation. DEMO enables precise definition and execution of these kinds of rules [13].
2. The provision of all historic events, all documents, all commitments, with time/date stamps, with guaranteed completeness in case of a dispute. This is also a complete litigation case file. By applying the blockchain technology, the case file becomes absolute trustworthy, it will be impossible to modify it.
3. The model can be extended or refined for any imaginable specific business situation and adding defined business rules [13]. Including partially accepted deliveries, return deliveries, not accepted payments, transaction roll-back etc. These claims are promising but unproven benefits and can be considered more as a topic of future research.
4. The model must be free from anomalies such as deadlocks. While in the real world it is possible to devise business rules may create anomalies such as a deadlock, a deadlock condition can be modeled also. Though undesirable, it must be possible to implement some system with a deadlock. Model simulation and validation identifies and mitigates anomalies such as deadlock and other anomalies.

5 REA Model-Driven GAAP Compliant Systems and Theories

New ontological theories promise a model-driven approach for the development of GAAP compliant accounting systems. The development of some GAAP compliant accounting system is then simplified to devising a conceptual model, expressed in a GAAP language, typically done by accounting experts. This model with matching software engine constitutes directly the GAAP compliant accounting system, which eliminates programming to a large degree (future research).

The theoretical foundations of the proposed approach are briefly described:

Guizzardi [4] proposed the foundations of ontological theories and a framework. This framework captures (i) the phenomena of a specific domain in the real world; (ii) the corresponding conceptualizations and (iii) an ontological modeling language. Any proposition expressed in that ontological modeling language specifies some phenomena that (may) exist in that domain in the real world.

Dietz J.L.G. [1] provided the DEO, DEMO Enterprise Ontology, a domain ontology that captures any enterprise that operates in the real world. The DEMO methodology provides conceptual models, formal representations of enterprises. Dietz J.L.G. provided also the Generic Systems Development Methodology [GSDP].

Van Kervel [6] extended the Guizzardi framework for static ontologies also for dynamic ontologies and for a model executing software engine. This is based on the GSDP methodology and results in the Generic Systems Development Process for Model-Driven Engineering [GSDP-MDE] of (software) systems. This approach has been proven; the DEMO engine has been built in this way [7].

The benefits are that in this way the development of a GAAP compliant accounting system demands much less resources; "only" a conceptual model is needed (best case). Also the validation that the accounting system is GAAP compliant is much easier. In case the GAAP rules change, the model can be changed very quickly. The automatic integration of different GAAP compliant systems to one coherent representation is another promise.

6 Conclusion and Results

It has been shown that he CC-CP Fact Model contains all required facts, with proper fact mapping for REA accounting systems, plus transactional behavior such as reject delivery, decline order, reject payment etc. The complete and correct factual mapping shows that the CC-CP model is appropriate to serve REA accounting systems.

However, much future research is needed to validate our generally careful claims: (i) more rigorous assessment of conceptual alignment REA - DEMO concepts; (ii) more empirical appropriateness case studies to support the claim that the CC-CP model captures any enterprise - enterprise co-creation and co-construction operation; (iii) in this perspective, many implementation-specific extensions of the CC-CP model; (iv) progress in the application of the GSDP-MDE approach and in conceptual modeling; the fact that one application - the DEMO engine - works well does not guarantee its generic applicability; (v) Notably conceptual modeling of GAAP compliant systems is a new domain.

Acknowledgements. The paper was supported by the grant provided by Ministry of Education, Youth and Sports Czech Republic, reference no. SGS09/PRF/2017.

References

1. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer, Heidelberg (2006)
2. Dietz, J.L.G., Hoogervorst, J.A.P.: The discipline of enterprise engineering. Int. J. Organisational Des. Eng. **3**(1), 86–114 (2013)
3. Dietz, J., Aveiro, D., Pombinho, J., Hoogervorst, J.: An ontology for the τ-theory of enterprise engineering. Frontiers Artif. Intell. Appl. **267**, 386–395 (2014)
4. Guizzardi, G.: Ontological foundation for structural conceptual models. Ph.D. theses, University of Twente (2005)
5. van Kervel, S.J.H., Dietz, J.L.G., Hintzen, J., van Meeuwen, T., Zijlstra, B.: Enterprise ontology driven software engineering. In: Proceedings of International Conference on Software Paradigm Trends (2012)

6. van Kervel, S.J.H.: Ontology driven enterprise information systems engineering: Ph.D. thesis, University of Technology Delft (2012)
7. Dudok, E., Guerreiro, S., Babkin, E., Pergl, R., Kervel, S.J.H.: Enterprise operational analysis using DEMO and the enterprise operating system. In: Aveiro, D., Pergl, R., Valenta, M. (eds.) EEWC 2015. LNBIP, vol. 211, pp. 3–18. Springer, Cham (2015). doi:10.1007/978-3-319-19297-0_1
8. Dunn, C.L., Cherrington, O.J., Hollander, A.S.: Enterprise Information Systems: A Pattern Based Approach. McGraw-Hill/Irwin, New York (2004)
9. Hruby, P.: Model-Driven Design Using Business Patterns. Springer, Heidelberg (2006)
10. McCarthy, W.E.: The REA accounting model: a generalized framework for accounting systems in a shared data environment. Account. Rev. **57**, 554–578 (1982)
11. Hunka, F., Zacek, J.: Detailed analysis of REA ontology. In: Aveiro, D., Tribolet, J., Gouveia, D. (eds.) EEWC 2014. LNBIP, vol. 174, pp. 61–75. Springer, Cham (2014). doi: 10.1007/978-3-319-06505-2_5
12. Hunka, F., Kervel, S.J.H., Matula, J.: Towards co-creation and co-production in production chains modeled in DEMO with REA support. In: Aveiro, D., Pergl, R., Gouveia, D. (eds.) EEWC 2016. LNBIP, vol. 252, pp. 54–68. Springer, Cham (2016). doi:10.1007/978-3-319-39567-8_4
13. Skotnica, M., van Kervel, S.J.H., Pergl, R.: Ontological foundation for the software executable DEMO action and fact models. In: Aveiro, D., Pergl, R., Gouveia, D. (eds.) EEWC 2016, LNBIP, vol. 252, pp. 151–165. Springer, Heidelberg (2016)
14. Nuffel, D., Mulder, H., Kervel, S.: Enhancing the formal foundations of BPMN by enterprise ontology. In: Albani, A., Barjis, J., Dietz, Jan L.G. (eds.) CIAO!/EOMAS -2009. LNBIP, vol. 34, pp. 115–129. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01915-9_9