



# A Generic DEMO Model for Co-creation and Co-production as a Basis for a Truthful and Appropriate REA Model Representation

Frantisek Hunka<sup>1(✉)</sup> and Steven van Kervel<sup>2</sup>

<sup>1</sup> University of Ostrava, 701 03 Ostrava, Czechia  
frantisek.hunka@osu.cz

<sup>2</sup> Formetis Consultants BV, Boxtel, The Netherlands  
steven.van.kervel@formetis.nl

**Abstract.** DEMO (Design Engineering Methodology for Organization) has its foundation in the DEMO Enterprise Ontology (DEO), and provides a strong theoretical foundation and a generic platform for business process modeling. The REA (Resource-Event-Agent) ontology, which originates from accountancy systems, provides a domain-specific platform for value modeling business processes. Rather than traditional approaches to accountancy, REA captures the details of each resource under an enterprise's control, and thus is able to offer a wider, more precise, and more up-to-date range of reports. Despite its great potential, REA ontology suffers from anomalies which have their origin in the absence of rigorous theoretical foundations. These anomalies can be overcome either by introducing rigorous theoretical foundations for the current REA ontology, or by useful collaboration of REA ontology with an ontology that provides a strong theoretical foundation. The paper deals with the latter option. It not only contemplates different aspects of both ontologies, but also analyzes and proposes a possible way for collaboration between these modeling frameworks.

**Keywords:** DEMO Enterprise Ontology (DEO) · DEMO methodology · Co-creation co-production · REA ontology · REA model

## 1 Introduction

The DEMO methodology is based on *the theory of* DEMO Enterprise Ontology and provides a generic platform for business process modeling. DEMO is based on a generic ontology, e.g., enterprise ontology, which meets the strictest requirements provided by conceptual modeling theories. DEMO is further based on the social communication and language theories of Habermas and others, general systems theory, the design science paradigm [8], conceptual languages, native executing software engines [16], and has strong formal foundations. DEMO is a modeling methodology of prescriptive knowledge - that provides four so-called aspect models of an enterprise. More specifically, it provides prescriptive knowledge (for execution) and descriptive knowledge (facts) about the enterprise. These four DEMO models [1, 4] are propositions in a formal language, each with a precisely defined grammar and vocabulary. Due

to the high degree of abstraction, it is conceptually guaranteed that *any* imaginable enterprise that may exist in reality - the real world, can be modeled in one, and only one, way. Its strong formal foundations enable the design and implementation of a software engine that directly executes DEMO models. This approach eliminates any programming; the model is the executable specification. Once DEMO models have been accepted as “the best representation of the enterprise”, these models can be executed in a production environment. DEMO model execution in production provides many valuable capabilities; complete workflow (-like) control of the actors in the enterprise; total knowledge of each atomic communication act of each actor, with complete audit trails, and process-mining (-like) analysis of daily business process execution.

DEMO Enterprise Ontology is a generic ontology in the formal sense [1, 2] which means that it strictly and exclusively captures the generic theoretical concepts within the domain ontology. These concepts are defined by the DEMO operation axiom [1]: (i) “there is a world of human actors that fulfill actor roles”; (ii) “there is a world of communication (coordination), of communicative acts and facts between actors”; and (iii) “there is a world of productions delivered by actors”. In addition, the DEMO transaction axiom states that actors that communicate with each other following a specific transaction pattern [3]. They cannot deviate from the transaction pattern.

The REA modeling framework is a domain-specific approach, which originated from the accountancy domain. This ontology is called the REA Enterprise Ontology as three of the fundamental concepts are Resources, Events, and Agents [6, 12]. The main benefit of the REA approach is that it enables the keeping track of primary and raw data about economic resources. All accounting artifacts are derived from the data describing exchange and conversion REA processes [9]. All reports based on the accounting artifacts are always consistent, since they are derived from the same data.

The structure of the paper is as follows. Section 2 shortly describes the main features of the DEMO methodology. The REA model with commitments and claim entities is depicted in Sect. 3. Section 4 deals with factual information support for the REA model analysis. The DEMO CC-CP model and its possibilities are presented in Sect. 5. Section 6 illustrates a simple example of a practical cooperation between the two modeling approaches. Discussion is delivered in Sect. 7. Conclusions and future research are depicted in Sect. 8.

## 2 The DEMO Methodology – Main Features

According to the DEMO methodology [1], an organization is composed of people (social individuals) that perform two kinds of acts, *production* acts and *coordination* (*communication*) acts. The result of successfully performing a production act is a *production fact*. An example of a production fact may be that a payment has been paid *and accepted*, or that an offered service has been accepted. All realization-specific details are fully abstracted out. Only the acts and facts as such are relevant, not how they are achieved. The result of successfully performing a *communication act* is a *communication fact*. Examples of coordination acts include *requesting* and *promising* a production fact, which essentially constitutes a mutually binding obligation (contract).

The subsequent communication acts and facts “*state*” and “*accept*” of the production constitute the fulfillment of the obligation (contract), agreed on by both actors.

A fact is a proposition about the real world that can be either false or true, and can be validated by empirical observation. A fact may encompass a single object, or may encompass more objects. Depending on the number of objects that are involved in a fact, we speak of unary, binary, ternary, etc., facts. An example of a unary fact is: *the Vendor is a Person*. An example of binary fact is: *a Customer receives a Pizza*.

In DEMO modeling, enterprises are represented by discrete deterministic systems that may exist in a set of precisely defined allowed states; the so-called state space [5]. For each state, there is a set of allowed transitions to another state, the so-called state transition space. All other state transitions are forbidden and cannot occur. In general, a state is determined by the set of facts that exist at that moment. A state change or state transition consists of one or more facts starting or ending to exist. The occurrence of a transition at some moment is called an event.

Events are widely defined as “things that happen in the real world” and that cause some effects. In DEMO there are only (i) communication (coordination) acts - one actor communicating with another actor, following the transaction pattern; (ii) production facts that describe the production of a specific actor; and (iii) facts, that are caused by acts in the real world that may become true or false. Example (i): a pizza has been requested by a customer and promised by the pizza baker, a contract has come into being. Example (ii): the production *fact* of the pizza baker is a pizza margarita. Example (iii): the exchange rate between the US dollar and the euro is 1.234. By empirical observation of the real world, this fact is either true or false.

The results of the DEMO methodology are the Construction Model, the Process Model, the Fact Model, and the Action model (four aspect models). For the CC-CP model presentation, the Construction Model is utilized.

### 3 The REA Model – Main Features

The REA model is composed of two kinds of different transactions termed “increment” and “decrement” with respect to the view of one of the agents. The two kinds of transactions form a ‘dual notion’, e.g., the type sale for the enterprise agent (left side), and the type purchase for the customer agent (right side). The term “increment” means that the value of resource(s) in the corresponding transaction(s) will increase, and the term “decrement” means that the value of resource(s) in the corresponding transaction (s) will decrease after completion of the REA model describing an exchange process [6, 12]. In the case of the REA conversion process, the resource(s) in one kind of transactions are consumed or used, and the resource(s) in a different kind of transaction(s) is/are produced (created), or some of its/their features change.

Each REA transaction is comprised of *commitment* and *event* entities, forming the dynamic part of a transaction. Further entities of a transaction make up a pair of economic *agents* with different interests and a *resource* entity. Apart from an agent and resource entities, representing “physical items”, an REA transaction can contain “category items” for resource and agent entities in the form of resource type entity, and agent type entity.

The commitment entity addresses the issue of modeling promises of future economic events, and the issue of reservation of resources. The reason for this solution is that economic events specify only actual increment or decrement in resource values, not the future increment or decrement in resource values. Commitment entities and their relationships with other entities are shown in Fig. 1. Each commitment is related to an economic resource by a *reservation* relationship that specifies which resources will be needed or expected by future economic events. A commitment entity is related to event entity/entities by the *fulfilment* relationship. The event entity represents the point in time at which actual change of property rights, or conversion of economic resources occurs.

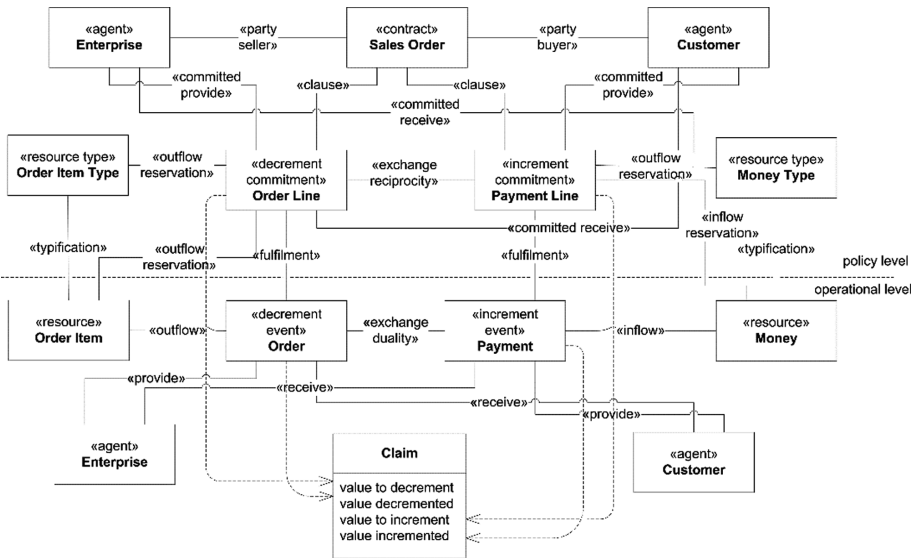


Fig. 1. REA model with commitments and claim entities. Source: [9]

Different kinds of transactions are related to each other by the *reciprocity* relationship, which relates different kinds of commitments, and by the *duality* relationship, which relates different kinds of economic events.

#### 4 Factual Information Support for the REA Model Analysis

Both modeling frameworks utilize the notion of a transaction or transaction pattern, which, in general, contains common things such as the two human beings partaking in the transaction, the resulting product (in REA *economic resource*) for which the whole transaction takes place, a promise given by one human being to perform a production act, and an event representing the occurrence of the production act (activity). However, a DEMO transaction represents a precisely defined state machine, and transactions are

ordered in a tree structure, utilizing production facts aggregation (*DEMO composition axiom*) to form a business process. In addition, the DEMO transaction contains complete possible states such as *decline* or *reject* etc., including revoking operations. The DEMO transaction “infrastructure” is robust enough to meet all real-world requirements, and thus forms a prescriptive system.

The REA framework assumes that a business process is composed of two kinds of transactions which are bound by the *reciprocity* and *duality* relationships see Fig. 1. In REA, one kind of transactions is performed in consideration of performing the other kind of transactions. In the view of one agent (actor role) one kind of transactions represents a decrement in value of economic resources, and the other kind of transactions represents an increment in value of economic resources. However, the REA approach does not provide a truthful state machine in the sense of the DEMO methodology, and represents a functional approach, thus forming a purely descriptive system.

In ontology collaboration, it must be taken into account that the DEMO methodology strictly distinguishes between the *coordination* and the *production* world, whereas the REA modeling approach only deals with the *production* world. It implies that the way in which DEMO captures real world phenomena is much broader and more comprehensive. The next step of collaboration concerns the locating and pin-pointing of appropriate conceptual mapping between the modeling approaches.

A *top-down conceptual mapping from REA to DEMO* would require the expression of each REA concept to the DEMO primitives, which would fail because REA does not provide all concepts and primitives that are necessary in DEMO. As a result, the DEMO models would be largely undefined, and hence useless.

A *bottom-up conceptual mapping from DEMO to REA* is the alternative way, in which REA might provide an accounting and financial perspective on an enterprise, and the DEMO concepts would be mapped to the REA concepts. This approach guarantees that useful results may be achieved, as DEMO models capture everything that is happening in the real world, with good empirical evidence.

In DEMO, actors communicate about a production fact. DEMO “says” nothing about production facts except that there is a hierarchical structure (*DEMO composition axiom*) in which production facts are arranged. Human actors commit themselves to a production fact (*DEMO Request* and *Promise*), and agree by communication (*DEMO State* and *Accept*) that the production fact exists in the real world. So, by analyzing the communication acts and facts it is possible to derive factual propositions about production facts, such as “the actors agree that a product has to be delivered and has to be paid for”, which is a mutually binding contract. The production facts “product X has been produced and accepted”, and “payment Y has been made” are the fulfilment of a contract. Since every atomic communication act and fact is precisely known and recorded at production time, it should therefore be possible to provide complete and correct information about all REA events that occur in the real-world.

In DEMO, real-world states and state transitions are expressed in the form of facts. DEMO models are able to supply REA model with all the facts currently needed by the REA model and possibly supply the REA model with other facts that may increase their applicability.

The inclusion of a state machine inside a DEMO transaction enables us to distinguish two complementary REA views on each REA model in one DEMO transaction. In a simple case, which can be used to demonstrate the approach, the *purchase/sales* and *money receipt/money disbursement* are utilized. One DEMO transaction represents both REA dependent views of *purchase* and *sales* in one. The request and accept DEMO transaction steps stand for *purchase*, whereas the promise and state transaction steps represent *sales*. The same holds for *money receipt* and *money disbursement*, in which *money receipt* represents the request and accept DEMO transaction steps, whereas the promise and state DEMO transaction steps represent the *money disbursement* REA dependent views.

The product part contains production facts and their specifications. More specifically, it is composed of independent facts and dependent facts. The product consists of independent facts e.g.: “purchase 6145 is completed”, and dependent facts e.g.: “article type is pizza Margherita” etc.

From the REA point of view, the most important coordination facts are e.g.: “sales 1658 is promised”, and “purchase 6145 is completed”, because they express a *committed* phase of transaction and a *fulfilled* phase of transaction.

However, the current DEMO Enterprise Ontology does not enable us to explicitly express all communication facts or to deal with any logic aggregated facts or dependent facts. The FAR (Fact, Agenda, Rule) Ontology [13], which is an extension to the current DEMO Enterprise Ontology, enable to support above mentioned issues.

In general, DEMO transactions are arranged in a tree structure with a parent-child relationship between them (utilizing the Composition axiom). The parent-child relationship is very effective and natural, but in some cases it is unable to capture all real world phenomena. By this we mean “the same level” transaction relationship, which is an inseparable part of a contract model, and must be signed by parties and evaluated in terms of contract fulfillment. The model of contract implicitly covers different kinds of transactions that are “on the same level” relationship. In order to solve the above described issues, the DEMO co-creation and co-production (CC-CP) model was conceived [10, 11].

## 5 The DEMO CC-CP Model

The FAR ontology [13] specifies 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. To illuminate the previous, let us consider the following example. Fact: “the invoice (xyz) has been paid”. Value true: the invoice has been paid, which can be validated empirically by checking the bank statement. Value false: the invoice has not been paid, also as shown by the bank statement. Value undefined: it is not known, probably because there is no access to any bank statements for empirical validation.

The FAR ontology enables the CC-CP model to utilize all communication facts and any logic aggregated facts. The DEMO CC-CP model captures all the facts relevant to the REA exchange process, and even other facts that are produced by REA information

and business events. Its main asset is in its ability to uniquely distinguish and capture *a contract on the table*, *a signed contract*, and *a fulfilled contract*.

### 5.1 The DEMO CC-CP Construction Model

This model is not only designed for utilizing individual exchange processes between a Principal and a Contractor, but it can also be used in production chains as an elementary building block. Its name is derived from its usage in production chains. Many highly specialized enterprises do not have a well-defined portfolio of products with fixed prices but offer their capabilities to meet the specific requirements of their Principals. Here we offer the following definitions: *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. The DEMO CC-CP model was firstly introduced in [10] and further developed in [11].

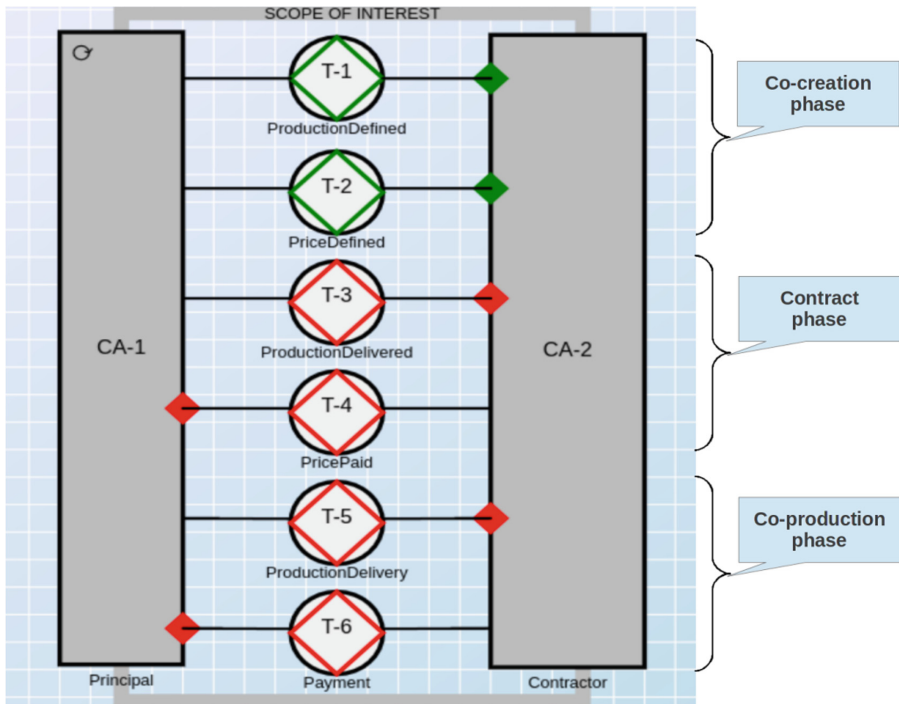


Fig. 2. The DEMO CC-CP construction model, Source: [11] (Color figure online)

The DEMO CC-CP construction model is illustrated in Fig. 2. This model is composed of three phases and each phase contains two DEMO ontological transactions. The *co-creation phase* represents a stage in which production (in REA resource types), as well as the price of production, are defined. When this phase is concluded, it means that the contract has been worked out but includes no obligations for the Principal and Contractor. The green color used in the drawings for T-1 and T-2 transactions indicates the *infological* layer, meaning that the production part and the price part of the contract are prepared but that the contract has not yet come into effect (business layer – the red color). The contract has not been signed in this phase.

The *contract phase* includes the signing of the contract by both parties (the contract comes into effect), which is represented by the reciprocity relationship in the REA model. This phase also involves fulfilment of the contract. The *co-production phase* addresses the individual production deliveries and individual payments for production deliveries. It follows from the nature of the exchange process that the products can be delivered in many sub-deliveries. Likewise, payment can be fulfilled by several partial payments.

## 5.2 The Bank Contents Table

The Bank Contents Table completes the Construction Model (see Fig. 2) by taking the state interpretation on transaction kinds. The Bank Contents Table of the DEMO CC-CP model is depicted in Table 1. In short, the Bank Contents table summarizes all production and coordination facts that the CC-CP model can provide for the REA model.

The left hand-side column marked “bank” contains individual transaction banks such as “T1 production definition”, or “T-5 production delivery”. Like the Bank Contents Table, the Construction model itself is composed of six transaction kinds (T-1–T-6). In transaction bank T-1, one can find every contract, enterprise, production and product kind that have been created. The fact that “the production of Contract is defined” is a production fact marked as P1. The other facts stated in the transaction bank T-1 are derived facts. In the case of aggregated facts, the bank column must encompass corresponding transaction kinds, meaning in particular that, for example, transaction banks T-3 and T-4 are mentioned instead of one transaction kind. The aggregated facts allow to express signing a contract as the fact: “the production agreement is promised and the price agreement is promised”, and fulfilling a contract as the fact: “the production agreement is accepted and price agreement is accepted”.

Apart from production facts, the transaction banks T-5 and T-6 also contain coordination facts. These facts are highly appreciated in accountancy systems, since they come into existence as a result of information or business events. Their meaning is usually self-explanatory, so only a few examples are stated. The coordination fact: “the [production order] was placed” means that the customer has sent out his/her production order to the enterprise. The coordination fact: “the [delivery order] was dispatched” represents the event during which the production contained in the delivery order was presented to the customer.



**Table 1.** The bank contents table of the DEMO CC-CP model

Bank	Independent/dependent fact
T-1 production definition	CONTRACT
	<b>The principal of Contract</b>
	<b>The contractor of Contract</b>
	ENTERPRISE
	<b>The production of Contract</b>
	PRODUCTION
	The product-kind of Production
	PRODUCT_KIND
	<b>The volume of Product-Kind</b>
	<b>The price of Product-Kind</b>
	<b>The delivery day of Product-Kind</b>
<b>The production of Contract is defined P1</b>	
T-2 price definition	PRICE
	<b>The price of Contract</b>
	MONEY_KIND
	The money kind of Price Line
	<b>The amount of payment of Money-Kind</b>
	<b>The day of payment of Money-Kind</b>
<b>The price of Contract is defined P2</b>	
T-3 production agreement, T-4 price agreement	CONTRACT_SIGNED
	The production_agreement is promised and The price_agreement is promised T3.pm and T4.pm
T-3 production agreement	PRODUCTION_AGREEMENT
	<b>The production_agreement is fulfilled P3</b>
T-4 price agreement	PRICE_AGREEMENT
	<b>The price_agreement is fulfilled P4</b>
T-3 production agreement T-4 price agreement	CONTRACT_FULFILLED
	<b>The production_agreement is fulfilled and The price_agreement is fulfilled P3 and P4</b>
T-5 production delivery	PRODUCTION_DELIVERY
	<b>The production delivery of Contract_Signed</b>
	<b>The product of Production_Delivery</b>
	PRODUCT
	<b>The actual volume of Product</b>
	<b>The actual price of Product (price per unit)</b>
	<b>The actual delivery day of Product</b>
	<b>The production order placed (sent) T5.rq</b>
	<b>The production order declined T5.dc</b>
	<b>The production order received T5.pm</b>
	<b>The delivery order handed over T5.st</b>
<b>The delivery order receipt T5.ac (P5)</b>	

*(continued)*

**Table 1.** (continued)

Bank	Independent/dependent fact
	<b>The</b> delivery order rejected T5.rj
T-6 payment	PAYMENT
	<b>The</b> payment of Contract_Signed
	<b>The</b> money-kind of Payment
	MONEY_KIND
	<b>The</b> actual amount of payment of Money-Kind
	<b>The</b> actual day of payment of Money-Kind
	<b>The</b> invoice placed (sent) T6.rq
	<b>The</b> invoice declined T6.dc
	<b>The</b> invoice received T6.pm
	<b>The</b> payment made (sent) T6.st
	<b>The</b> payment receipt T6.ac (P6)
<b>The</b> payment rejected – dispute T6.rj	

The Bank Contents Table provides a detailed overview of all production and coordination facts that the DEMO CC-CP model is able to capture and deliver for further processing. In general, it also means that not all the facts may be needed by the REA model.

## 6 An Instance of the REA Model and Facts Identification

The Bank Contents Table, which is part of the CC-CP Construction Model expresses both production and coordination facts of a simple purchase/sales process, see Table 1. In order to get a deeper insight into REA semantics, an example of an REA instance model that describes a sales process is stated. The process represents a sales order between the customer (Adam) and the salesman (Mia’s pizzeria). The customer orders two Margherita Pizzas and one Cola 0.5 l. The REA model can be described as a contract (Sales Order) which is composed of two kinds of transactions. These transactions are marked as increment or decrement transactions according to the value of the related resource entity. In the REA transaction, it is usually possible to distinguish between a physical item and category item. In this case, a resource entity represents a physical item and a resource-type entity represents a category item. The same holds for an agent entity and agent-type entity. Resource and resource-type entities represent domain-specific entities.

As is obvious from Fig. 3, the REA instance model consists of two decrement transactions and one increment transaction. The first decrement transaction includes the *commitment Line1*: Sales Line captures the liability of the salesman to sell two pizzas to the customer. The *event Line1*: Sales Line represents production itself, which means

that two pizzas were delivered to the customer by the salesman. This transaction contains the resource type and resource of Margherita pizza and a pair of agents: the salesman and the customer.

The second decrement transaction includes the *commitment Line2*: Sales Line which captures the liability of the salesman to sell one Cola 0.5 l to the customer. The *event Line2*: Sales Line represents the delivery of Cola 0.5 l from the salesman to the customer. The transaction is accompanied by the resource type and the resource representing Cola 0.5 l, and a pair of agents.

The third increment transaction includes the *commitment Total*: Sales Line and captures the liability of the customer to pay the salesman for the ordered goods. The *event Total*: Payment Line represents the making of a payment by the customer, and the acceptance of it by the salesman. The resource type is represented by money.

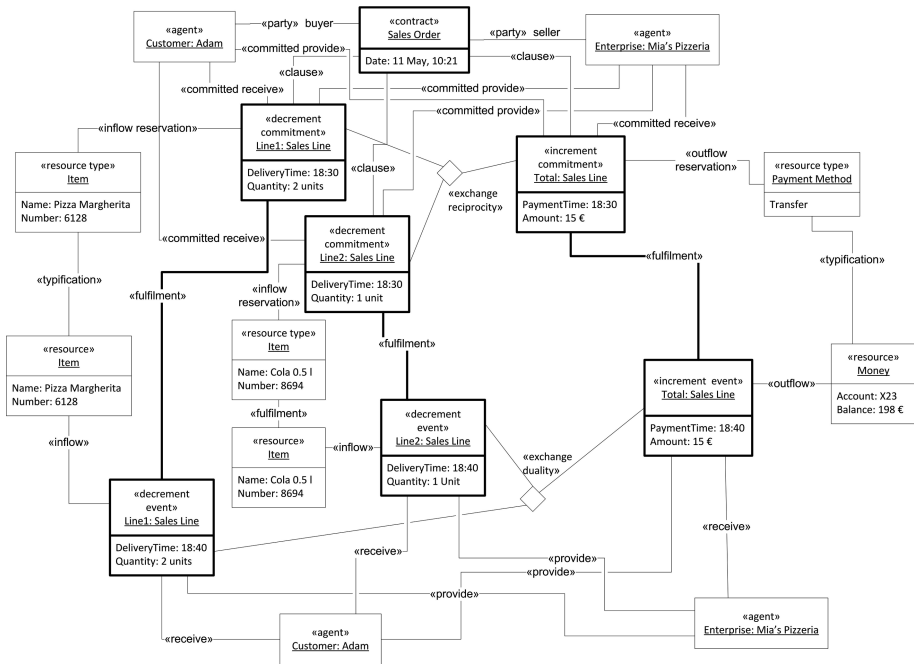


Fig. 3. An instance of the REA application model of a sales order

The core relationships in the REA model are relationships which relate the decrement commitments to increment commitments (the reciprocity relationship), and which relate the decrement event to the increment events (the duality relationship).

The reciprocity relationship represents signing a contract in which one kind of transactions (e.g., decrement transactions) is in consideration of the other kind of transactions (e.g., increment transactions). It means that the corresponding economic agents (actor roles) have agreed on the resource types, and their amount that will be exchanged for another amount of resource types, at the time and place promised in the commitment. This agreement also supposes that the promised amount of resource types will be available at the promised time.

Only a simple contract is considered, and therefore no further commitments reflecting, e.g., penalties, are stated. The duality relationship represents the fulfillment of individual transactions and the whole contract.

The process of fact identification will proceed from the Bank Contents Table, containing all production facts and necessary coordination facts, and the REA instance sales model. The coordination facts and production facts will be identified in a simple example of purchase/sale and money receipt/money disbursement REA process, which is shown in Fig. 3. As can be seen in the Figure, the REA process is composed of two different kinds of transactions: two goods (products) transactions and one money transaction.

DEMO, as mentioned earlier, utilizes only one view on the REA exchange process. The first kind of transaction is called purchase/sale transaction, in which a customer is in the role of the purchaser and a vendor is in the role of the salesman. More specifically, *request* and *accept* transaction steps are issued by the customer, and *promise* and *state* transaction steps are issued by the salesman. In the other transfer, money receipt/money disbursement are also complementary operations, as in the previous case. The *request* and *accept* transaction steps are issued by the salesman (cashier), and the *promise* and *state* transaction steps are issued by the customer (payer). DEMO's coordination acts/facts enable to create a more vivid model, with only one "independent" view on both kinds of transactions.

The contract itself is a more specific entity than commitment, as it contains different options representing commitment that will be instantiated on the basis of different external conditions, or on the basis of the actor's choice. The DEMO CC-CP model can identify individual or aggregated facts. The reciprocity and duality relationships must be composed additionally from the individual facts. The model can provide more detailed facts which can be further elaborated. Only the basic facts that are needed by the REA model are described. A production line represents individual lines with a resource kind in Purchase Order. A price line represents total evaluation in money kind for all production lines in Purchase Order (Table 2).

**Table 2.** Summary of facts that the CC-CP model can provide for the REA sales order model

Fact No	Fact description	REA model relationship
1	Contract [ <i>sales order #132</i> ] come into enforce on [ <i>current day 18:15</i> ]	
2	Contract of [ <i>sales order</i> ] has ID [ <i>#132</i> ]	
3	Customer of [ <i>sales order #132</i> ] is [ <i>Adam</i> ]	Party
4	Enterprise of [ <i>sales order #132</i> ] is [ <i>Mia's Pizzeria</i> ]	Party
5	Production line of [ <i>sales order #132</i> ] is [ <i>Line1</i> ]	Clause
6	Product kind of [ <i>Line1</i> ] is item [ <i>#6128</i> ]	Inflow reservation
7	Quantity of [ <i>Line1</i> ] is [2]	
8	Delivery day of [ <i>Line1</i> ] is [ <i>current day 18:30</i> ]	
9	Item [ <i>#6128</i> ] has name [ <i>Pizza Margherita</i> ]	
10	Price per unit of item [ <i>#6128</i> ] is [9 €]	
11	Production line of [ <i>sales order #132</i> ] is [ <i>Line2</i> ]	Clause
12	Product kind of [ <i>Line2</i> ] is item [ <i>#8694</i> ]	Inflow reservation
13	Quantity of [ <i>Line2</i> ] is [1]	
14	Delivery day of [ <i>Line2</i> ] is [ <i>current day 18:30</i> ]	
15	Item [ <i>#8694</i> ] has name [ <i>Cola 0.5 l</i> ]	
16	Price per unit of item [ <i>#8694</i> ] is [1.5 €]	
17	Price line of [ <i>sales order #132</i> ] is [ <i>Total</i> ]	Clause
18	Payment method of Price line [ <i>Total</i> ] is [ <i>method</i> ]	Outflow reservation
19	Payment time of [ <i>Total</i> ] is [ <i>current day 18:30</i> ]	
20	Money kind of [ <i>Total</i> ] is [ <i>money kind #3541</i> ]	
21	Total amount of [ <i>Total</i> ] is [19.5 €]	
22	Actual product delivery of [ <i>Line1</i> ] is product ID [ <i>#6128</i> ]	Fulfilment
23	Actual quantity of product delivery of [ <i>Line1</i> ] is [2]	
24	Actual delivery day of product [ <i>Line1</i> ] is [ <i>current day 18:40</i> ]	
25	Actual price per unit of product ID [ <i>#6128</i> ] is [9 €]	
26	Actual product delivery of [ <i>Line2</i> ] is product ID [ <i>#8694</i> ]	Fulfilment
27	Actual quantity of product delivery of [ <i>Line2</i> ] is [1]	
28	Actual delivery day of product [ <i>Line2</i> ] is [ <i>current day 18:40</i> ]	
29	Actual price per unit of product ID [ <i>#8694</i> ] is [1.5 €]	
30	Production agreement [ <i>#3132</i> ] was fulfilled	
31	Actual payment of [ <i>Total</i> ] is money [kind]	outflow
32	Actual payment amount of [ <i>Total</i> ] is [19.5 €]	
33	Actual payment day of [ <i>Total</i> ] is [ <i>current day 18:45</i> ]	
34	Price agreement [ <i># 4132</i> ] was fulfilled	
35	Contract [ <i>sales order #132</i> ] was fulfilled on [ <i>current day 18:45</i> ]	

This section shows that the DEMO CC-CP model is able to capture and provide all facts (production, coordination, and aggregated) for the REA model representation. Software execution of the CC-CP model should provide all information needed for a REA compliant accounting system.

## 7 Discussion

There are two principal reasons for a truthful and appropriate REA model representation by a generic DEMO model for co-creation and co-production. The first reason is that the DEO (DEMO Enterprise Ontology) ontology is a generic foundational ontology and the REA ontology is a domain-specific ontology [1, 2, 14]. This implies that the generic ontology with DEO qualities and capabilities should support a domain-specific ontology. It can be emphasized that among other benefits that DEO provides, there is the capability of grasping all the phenomena that occur in reality with good empirical evidence [7, 15]. In general, this feature of DEO is worthwhile for the REA ontology because it could considerably extend its functionality.

The second reason potentially supporting cooperation is that the DEO provides also prescriptive information systems of the enterprise (not only descriptive information systems). If we apply the DEMO engine and execute the REA model in DEMO modeling language, then the generic transaction pattern gives the actor roles firm guidance from which they cannot be deviated; it is an enforcing business procedure. This feature may be highly useful for REA ontology since it only provides descriptive knowledge. The DEMO prescriptive capabilities can dramatically improve the rather “loose coupling” between the REA’s commitment entity and economic event entity, thus forming a principal element of REA transactions. For REA, this would, in essence, entail a shift towards financial information systems with precisely defined relations between a commitment entity and economic event entity.

To realize collaboration between different ontologies, some kind of mapping between ontologies must be set up. Whereas the *top-down approach* (starting from accounting artifacts trying to capture the phenomena and things in the real world) proves to be ineffective, the *bottom-up approach* (to develop some DEMO model that captures all REA artifacts well and without anomalies) shows to be a passable means of potential collaboration. As can be seen from the previous text, collaboration of both ontologies doesn’t represent a horizontal way of collaboration between two more or less equal sides. Collaboration utilized in the described approach represents a systematic hierarchical approach, in which the DEMO CC-CP model –the bottom part– supplies factual knowledge to the REA model –the upper part. The mapping itself is based on elementary parts - facts that can be transferred to the REA model.

Information contained in the form of facts would require some other (additional) operations to transfer these facts into the form of the REA information system. But this demand is less difficult than supplementing REA ontology with features described above.

## 8 Conclusions

The paper deals with the idea of a generic and a domain-specific ontologies collaboration in a *systematic hierarchical way*. This collaboration is designed and clarify in the form of facts (elements of information) that are produced by the DEMO CC-CP model and are intended for the REA model. The presented solution is based on systems engineering, the construction of a system, in such a way that a desired functional behavior of the system is realized. Two important quality criteria have been discussed; ontological truthfulness and ontological appropriateness.

All relevant real-world phenomena must be well captured by the DEMO CC-CP model; otherwise it is impossible to devise a working REA compliant accounting system. The DEMO CC-CP model together with the Bank Contents Table (Sect. 5) provide, in general, summary of all production and coordination facts that the proposed model is capable to capture and deliver in the area of reciprocal transaction modeling. In this way, a claim of appropriateness – execution of the DEMO CC-CP model provides all factual information for a REA accounting system – is provided. Failure to meet this quality criterion renders the DEMO CC-CP model totally useless.

Future research will be aimed at real-world verification and validation of the proposed DEMO CC-CP model towards REA model representation. The further goals of the future research will be analyze and modeling of a more complex and robust DEMO CC-CP model.

**Acknowledgements.** The paper was supported by the grant provided by Ministry of Education, Youth and Sports Czech Republic, reference number SGS06/PRF/2019.

## References

1. Dietz, J.L.G.: Enterprise Ontology Theory and Methodology. Springer, Heidelberg (2006)
2. Dietz, J.L.G.: Architecture - Building strategy into design. Academic Service (2008)
3. Dietz, J.L.G., Hoogervorst, J.A.P.: The discipline of enterprise engineering. *Int. J. Organisational Des. Eng.* **3**(1), 86–114 (2013)
4. Dietz, J.L.G.: The Essence of Organization, 2nd edn. Sapio Enterprise Engineering, Delft (2015)
5. Dudok, E., Guerreiro, S., Babkin, E., Pergl, R., van 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). [https://doi.org/10.1007/978-3-319-19297-0\\_1](https://doi.org/10.1007/978-3-319-19297-0_1)
6. Dunn, C.L., Cherrington, O.J., Hollander, A.S.: Enterprise Information Systems: A Pattern Based Approach. McGraw-Hill/Irwin, New York (2004)
7. Guizzardi, G.: Ontological foundation for structural conceptual models. Ph.D. thesis, University of Twente (2005)
8. Hevner, A.: A three cycle view of design science research. *Inf. Syst. Decis. Sci. Scand. J. Inf. Syst.* **19**(2), 87–92 (2007)
9. Hruby, P.: Model-Driven Design Using Business Patterns. Springer, Heidelberg (2006)

10. Hunka, F., van 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). [https://doi.org/10.1007/978-3-319-39567-8\\_4](https://doi.org/10.1007/978-3-319-39567-8_4)
11. Hunka, F., van Kervel, S.J.H.: The REA model expressed in a generic DEMO model for co-creation and co-production. In: Aveiro, D., Pergl, R., Guizzardi, G., Almeida, J.P., Magalhães, R., Lekkerkerk, H. (eds.) EEWC 2017. LNBIP, vol. 284, pp. 151–165. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-57955-9\\_12](https://doi.org/10.1007/978-3-319-57955-9_12)
12. McCarthy, W.E.: The REA accounting model: a generalized framework for accounting systems in a shared data environment. *Account. Rev.* **57**, 554–578 (1982). **2**(5), 99–110 (2016)
13. Skotnica, M., van Kervel, S.J.H., Pergl, R.: Towards the ontological foundations 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, Cham (2016). [https://doi.org/10.1007/978-3-319-39567-8\\_10](https://doi.org/10.1007/978-3-319-39567-8_10)
14. van Kervel, S.J.H., Dietz, J.L.G., Hintzen, J., Meeuwen, T., van Zijlstra, B.: Enterprise ontology driven software engineering. In: Proceedings of ICsoft 2012 – 7th International Conference on Software Paradigm Trends, pp. 151–165. SciTePress (2012)
15. van Kervel, S.J.H., Hintzen, J., van Meeuwen, T., Zijlstra, B., Vermolen, J.: A professional case management system in production, modeled and implemented using DEMO. In: CBI 2014, 8th TEE Workshop: Transformation & Engineering of Enterprises, Geneva, Switzerland, pp. 205–210 (2014)
16. van Kervel, S.J.H.: Ontology driven enterprise information systems engineering, Ph.D. thesis, University of Technology Delft (2012)