

# Contracts<sup>1</sup>

## A General Pattern for Business Modeling

*Pavel Hruby*

*Navision a/s  
Frydenlunds Allé 6  
DK-2950 Vedbaek, Denmark  
E-mail: ph@navision.com*

A contract is:

- a) an agreement between two or more parties, especially one that is written and enforceable by law, or
- b) the writing or document containing such an agreement.

(The American Heritage® Dictionary of the English Language, Fourth Edition.)

### **Context**

Contracts are at the core of business. Business administration systems implement functionality that focuses on various aspects of relationships between business partners, such as customers and vendors, employers and employees, service providers and service receivers. The relationships usually manifest themselves in the form of exchanges of economic resources, such as the purchase and sale of products and the exchange of financial resources. These exchanges are accompanied by various business documents, such as purchase and sales orders, invoices and receipts. These documents specify commitments and constraints for the participating parties. Some examples of such documents are as follows:

- Purchase order: a commitment for the vendor to deliver goods and obligation for the customer to pay for them. In addition, purchase order specifies other purchase properties, such as delivery date and what happens if the delivery date is not met.
- Invoice: a declaration of the claim that the buyer owes a specific amount of money to the seller. In addition, an invoice typically specifies other properties such as payment terms.
- Employment contract: specifies details about relationship between employee and employer. In addition, the employment contract specifies other conditions of the employment, such as position and compensation.
- Delivery receipt: a declaration of a material movement between business partners, warehouse sites, or between a warehouse site and a business partner.
- Payment: a transfer of money from one business partner to another.

I call the business partners participating in a business relationship *parties*. In keeping with the authors of other publications [1], [3], [4], [6], [7], I use the term *party* to mean a business entity that can participate in a contractual relationship with another party, such as a person, company, legal entity, team, or organizational unit.

---

<sup>1</sup> A note for the reviewer: This pattern does not cover agreements, that is, general entities such as service level agreements that contain rules that control specific entities, such as purchase and sales orders. Therefore, the name contracts might be confusing. Alternative names might be something like business events, business relations, or “work before party”.

I call the abovementioned business documents *contracts*. The term *contract*, as used in this paper, covers relationships between parties in various scopes and at various levels of abstraction. An example of a general scope contract is a contract for providing a maintenance service in a given period of time. This contract can result in more specific contracts, such as service orders, and they result in more specific contracts, such as material movements.

The *economic resource* is a subject of trade. I use the term resource to mean a concrete physical product, asset, inventory, or service that has identity. For example, a product that has a serial number, or a service that has a start time and end time. I use the term *resource type* to mean the description of a tangible product, or the description of a service.

## **Problem**

Have you ever tried to describe an object model of a business system and struggled to find the right relationships between business entities, such as customers, business partners, products, sales and purchase orders, invoices and credit memos? Have you ever wanted to know a simple rule for modeling the business system in a consistent manner?

## **Forces**

1. Parties have relationships between each other. We can model these relationships as associations between the parties. However, these relationships often have specific attributes, describing details of this relationship rather than details of one of the parties. Examples of such attributes are delivery due, validity period, payment terms and employment position. Because of these attributes, the relations between parties cannot be modeled as pure associations.

2. Relationships between parties involved in business are manifested in various documents, such as purchase and sales orders, quotes, invoices, payments, and delivery receipts. These documents vary in complexity and it is not possible to determine a complete set of business documents that fit all situations in all businesses. However, you want to capture all these relationships in a uniform way in the object model.

3. If a relationship exists between parties, this relationship may, under certain conditions, create or cause the creation of other relationships between the same parties. For example, a purchase order may result in a delivery of goods. You want to capture this fact in the object model. However, traditional object-oriented modeling techniques do not give any hints for how to describe the fact that one relationship between objects can imply another.

## Solution

The relationship between parties is encapsulated in the entity called *contract*. The contract entity is related to exactly<sup>2</sup> two party entities: the supplier and the consumer. Examples of parties are customer, vendor, or employee. Each party represents a physical person or organization. Each contract is related to exactly one resource or resource type. Each resource or resource type represents an asset, a physical product or other subject of trade.

The structure of the contract pattern is illustrated in Fig. 1.

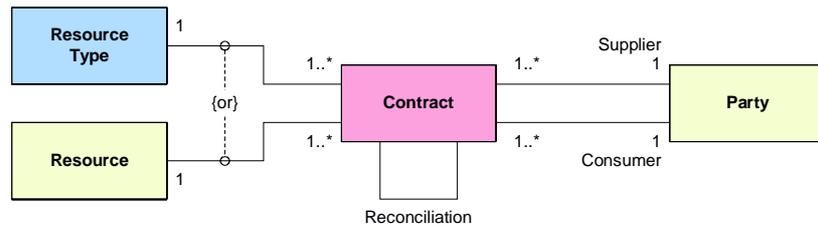


Fig.1. Structure of the contract pattern

Please note that in this contract there is a one-to-many relationship between the resource and the contract. If the business case requires the contract and several resources are related, the variant of the pattern with contract lines should be used.

Sometimes the contract entity contains a variable number of contract lines, each of them precisely linked to one resource or resource type. Please see the sub-pattern *contract lines* at the end of this paper for details. The structure of the contract with contract lines is in Fig. 2.

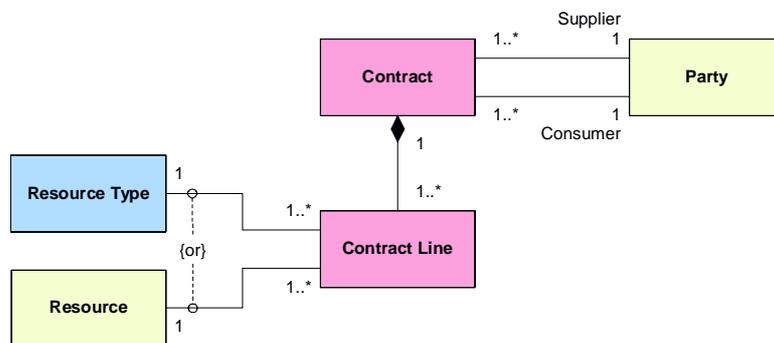


Fig. 2. Structure of the contract pattern with contract lines

The pattern specifies that in the business cases covered by the contract pattern, one party *cannot* be directly related to another party. The contract lines follow the lifetime semantics of the whole contract, that is, if a contract is copied or deleted, the contract lines are copied or deleted together with the contract.

<sup>2</sup> We have not come across a situation of a multi-party contract that could not be decomposed into a number of binary contracts. I would appreciate to know a contra example.

## Examples - Known Uses

The contract pattern can be applied to any business document that:

- indicates relationship between parties,
- specifies time interval, event, or several events that are significant to this relationship, or
- specifies the exchange of resources, or conditions of ownership of resources.

Examples of contracts are: purchase and sales orders, invoice, credit note, delivery of goods, and employment.

## Sources

This section compares four different descriptions of this pattern: David Hay's contract, Peter Coad's domain-neutral component, McCarthy's Resources, Events-Agents model (REA) and Order, Shipment, Invoice and Employment by Silverston, Inmon and Graziano. I believe that these four descriptions are simply variations of the same pattern. The authors of these four patterns do not cite each other, therefore, they probably were not aware of this similarity.

## Contracts by David Hay

David Hay describes the contract pattern in [6]. The contract pattern links together four entities: *the party*, *contract*, *contract line* and *asset*. David Hay's contract pattern is illustrated in Fig. 3. I use Peter Coad's color scheme [1] to show correspondence between entities in the discussed contract models.

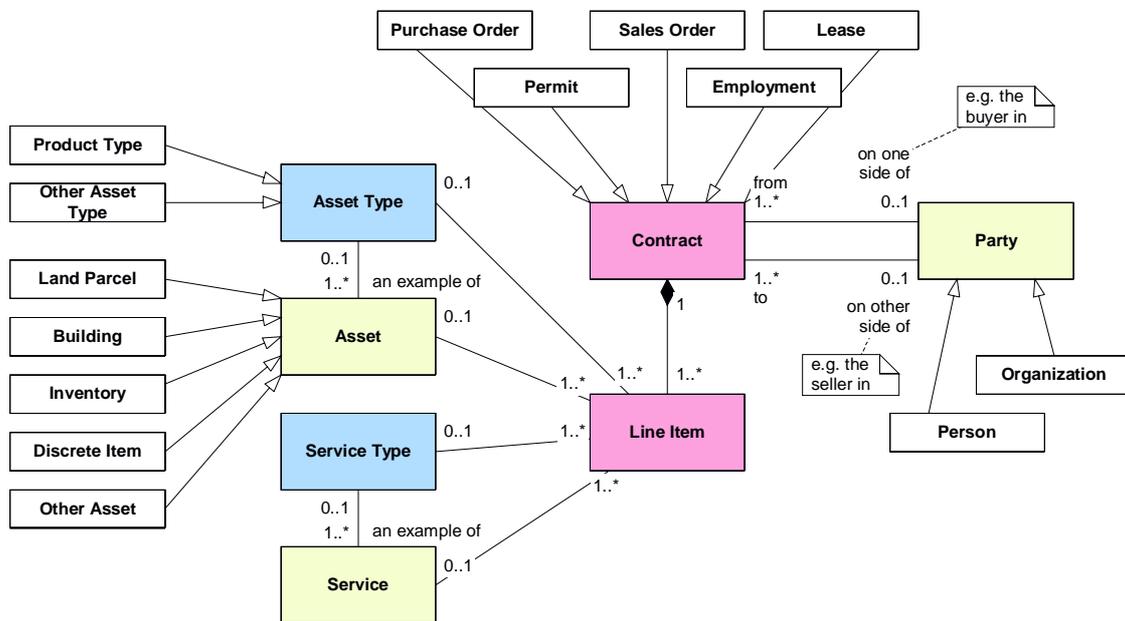


Fig. 3. Contract pattern by David Hay. Adapted from Fig. 6.5 in [6].

Examples of David Hay's contracts are purchase orders and sales orders, lease, permit and employment contract. Attributes of the contract entity are the contract number,

date, contract terms and the total contract value. Contracts consist of line items. Attributes of the line items are the quantity, price and extended value.

Party is used with the same meaning as in our pattern.

Asset corresponds to the resource in our pattern. Examples of assets are land parcel, building, inventory, and discrete item. Attributes of the asset entity are the description and the actual price.

Asset type corresponds to resource type in our pattern. The attributes of asset type are the asset description and unit price.

David Hay discusses a variation of the contract pattern in which contract is linked to more parties than the buyer and seller. The entity that encapsulates these additional parties is called *contract role*. David Hay discusses deliveries of products and services, but considers them as additional entities that are linked to the contract entity. He does not consider deliveries as other instances of the contract pattern, as I do.

### Domain-Neutral Component by Peter Coad

Peter Coad in [1] describes a generic business component that contains four interconnected building blocks, called archetypes. The archetypes are:

- the moment-interval
- the party, place, or thing
- the role of the party, place or thing
- the catalog-like description of the party, place or thing

An archetype is a form, from which all things of the same kind *more or less* follow (in contrast to types, stereotypes or interfaces that *must* be followed). The archetypes are illustrated in Fig. 4.

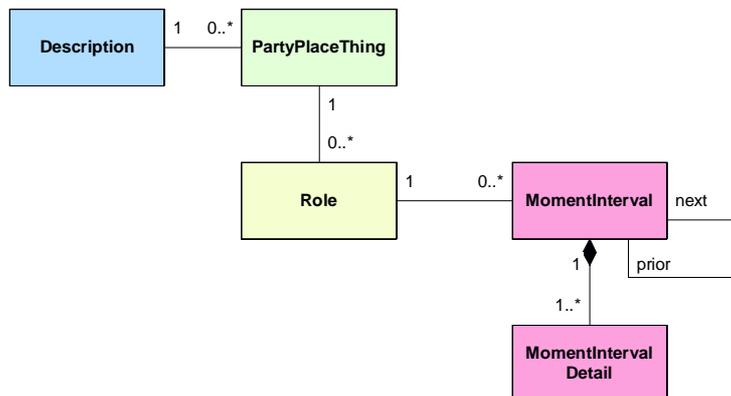


Fig. 4. Archetypes

The moment-interval represents something that occurs at a moment in time or over an interval of time. Examples of moment-interval are sale, rental, and reservation. The attributes of the moment interval archetype are its number, date, time, or interval, its priority, its total, and its status. Moment-intervals often have parts, called moment-interval details.

Party, place or thing are archetypes that can play different roles. Examples are person, which can play roles of customer or employee; and product, which can play the roles of raw material or finished goods. Attributes of the party, place, or thing archetypes are the serial number, name, address and custom value.

The role is a way of participation by a party, place or thing. Examples of roles are employee, customer, raw material or finished goods. Attributes of the role archetype are the assigned number and status.

The catalog-entry like description (of a party, place or thing) represents the collection of all things that correspond to its description. An example of a description archetype is a car model. Attributes of the description archetype are the type, description, item (type) number, and default value.

The domain-neutral component is illustrated in Fig. 5

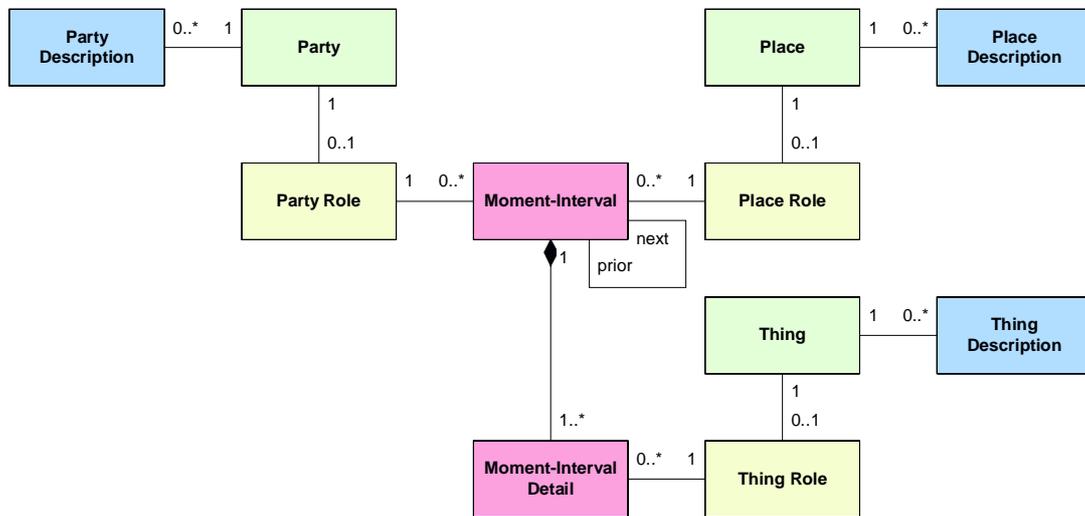


Fig. 5. Domain-neutral component by Peter Coad

As all things of the same kind more or less follow the archetypes, the same applies for the relationships. In some cases, the moment-interval is connected directly to the party, place or thing, or directly to the description. The role is sometimes directly connected to the description.

## Order, Delivery and Invoice by Silverston, Inmon and Graziano

Len Silverston, W.H. Inmon and Kent Graziano in [7] describe patterns order, delivery and invoice, which have similar structure and can be considered as instances of our contract pattern. The structure of the order pattern is illustrated in Fig. 5.

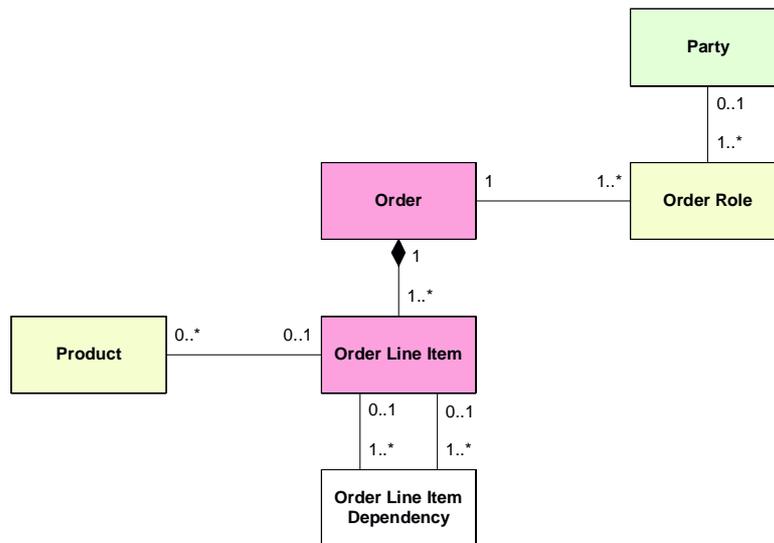


Fig. 6. Order by Silverston, Inmon and Graziano

Examples of the order pattern are purchase and sales orders. In addition, a similar structure applies to shipment, invoice, work order and position (an employment contract), all of which are discussed in [7].

Attributes of the order are order ID, order date, and order entry date (that is, when order has been registered in the system). Attributes of order line item are quantity and unit price, estimated delivery date, estimated ready dated for shipment, and estimated start date (in the case of service order).

Order line item dependency is an entity that relates line items to the line items of another contract. For example, in the wholesale scenario, this entity traces the product on sales order line to the same product on the purchase order line. The order item dependency corresponds to the reconciliation relation in our contract pattern.

## REA (Resources, Events, Agents)

The REA (Resources, Events, Agents) pattern was originally formulated as an alternative model to traditional accounting systems. The REA describes exchanges of economic resources between business organizations and the rules that control these exchanges. We believe that the REA pattern describes the contract pattern, with additional constraints.

The REA pattern was originally formulated by McCarthy [8], [5]. A very illustrative explanation of the contract pattern can be found in [2]. The REA pattern is shown in Fig. 7.

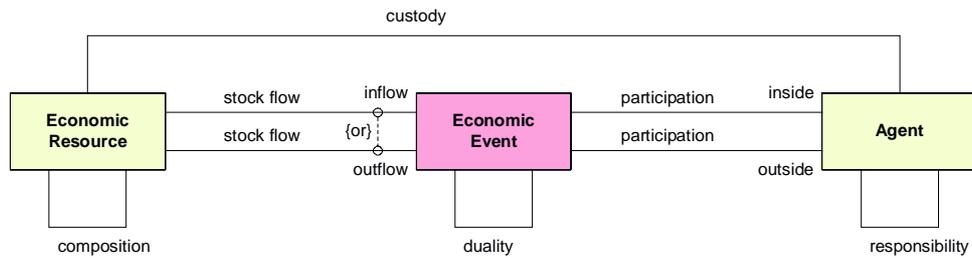


Fig. 7. The REA pattern. Adapted from Fig. 2 in [5].

The REA model contains three related entities: resource, event and agent.

An economic resource is an object that is scarce, has utility and is under the control of an enterprise. Examples are cash, inventory, labor service and machine service. REA economic resource corresponds to the resource of our pattern.

An economic event is the change of control of an economic resource from one business partner to another. Examples of economic events are sale, cash payment and shipment. An REA event corresponds to the entity contract of our pattern.

Agents are parties that participate in the events. Internal agents are inside the business unit, and they are responsible for performing the business event. Examples of internal agents are salesperson, and cashier. External agents are outside the business unit. Examples are customer, vendor. The REA agent corresponds to the entity party of our pattern.

There are certain constraints in the REA model, formulated as REA axioms:

- At least one inflow event and one outflow event exists for each economic resource – conversely, inflow and outflow events must affect identifiable resources.
- All events effecting outflow must eventually be paired in a duality relationship with events effecting an inflow, and vice versa.
- Each exchange needs an instance of both the inside and outside agents.

Guido and McCarthy in [5] extended the REA model of the concept of commitment. Commitment is an agreement to execute an economic event in a well-defined future that will result in either an increase of resources or a decrease of resources. An REA model with commitments is illustrated in Fig. 8. In our pattern, REA commitment corresponds to contract entity.

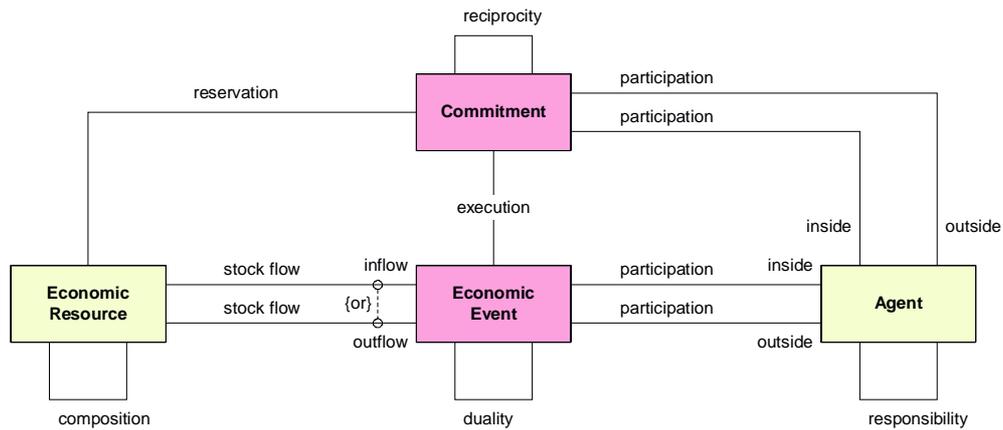


Fig. 8. The extended REA model with commitments<sup>3</sup>

### Comparison of the Models

Mapping between the entities used in the four contract models is illustrated in Table 1.

The REA model provides more information than other contract models by specifying economic constrains – the REA axioms. Furthermore, REA distinguishes between contracts as commitments and contracts as economic events.

This pattern	Hay	REA	Coad	Graziano
Contract	Contract	Economic Event, Commitment <sup>4</sup>	Moment-Interval	Order, Quote, Shipment, Invoice, Position
Contract Line	Line Item	Economic Event, Commitment	Moment-Interval Detail	Order Line Item, Invoice Line Item, ...
Resource	Asset, Service	Resource	Thing Role	Inventory Item
Resource Type	Asset Type, Activity	--	Description	Product
Party	Party	Agent	Party Role	Order Role
--	--	--	Thing	--
--	--	--	Party	Party
Reconciliation	--	Duality Execution	Prior, Next	Dependency

Table 1. Comparison of the contract patterns

The contract in the pattern by Silverston, Inmon and Graziano is related only to one party. In their pattern, the entity “This Business Unit” is implicit.

<sup>3</sup> Note for the reviewer: We get a more symmetric model if we add the “expectation” relationship between the commitment and economic resource. While the “reservation” commitment executes the “outflow” economic event, the “expectation” commitment would execute the “inflow” economic event. Shall I add the “expectation” relation to the figure, event if it is not present in McCarthy’s model?

<sup>4</sup> Both the contract and the contract line can be, but are not always, the REA event and commitment.

REA and Hay's contract do not have an entity equivalent to Coad's Party, Place or Thing. The party entity in Hay's contract corresponds to the agent entity in REA. They correspond to party-place-thing role archetype of the Coad's pattern. Please note that Hay's party and REA agent are, in fact, roles. For example, the same physical organization or person can, in fact, represent both the customer and vendor. In other words, REA and Hay's parties do are not the physical organization or person, only their roles in the contract.

The party entity in Hay's pattern can be extended to cover Coad's place entity. For example, a warehouse can be considered as a party in the material movement contract.

## ***Sub-Pattern: Contract Lines***<sup>5</sup>

### **Context**

Information between business partners is exchanged in terms of various business documents, such as contracts, invoices, and payments. Some of them, such as invoices, orders, consist of header and lines. Some do not, such as reminders and notifications.

### **Problem**

Should a business document that we are designing consist of a header and lines?

### **Forces**

- A business document specifies the business partners and the subjects of information exchange, such as products and various messages. The same product information or message can appear in various business documents. For example, specific product information or a greeting message can appear both on the order and the invoice. It is possible to create this information individually for every business document, but we would like to reuse it across different documents.
- It is technically possible to create individual business documents for each subject of information exchange, such as the individual product. However, the information about several products is often sent or received in chunks, and the times of sending and receiving are common for the entire chunk.
- Some operations in business documents, such as posting, can be called individually for each product that is referred to in the business document. However, we would like the posting date be the same for all products in the business document.
- Some business documents represent claims to the business partner. For example, a purchase order is a claim to the vendor to deliver products. Some attributes of this claim, such as quantity, are associated with individual products, rather than shared across them. If several products are referred to in the business document, we need a placeholder for this product-specific claim.
- There is a many-to-many relationship between the business document and the product. This many-to-many relationship is conceptually possible, but technically difficult to realize.

---

<sup>5</sup> Note for the reviewer: I am trying to understand the forces why some business documents have lines. I would appreciate feedback on this.

## Solution

Make a business document that consists of header and lines. The lines specify the resources, products and other subjects of exchange. The header specifies the sending and receiving dates, the posting method, the business partners, and other information that is common across the lines.

## Sketch

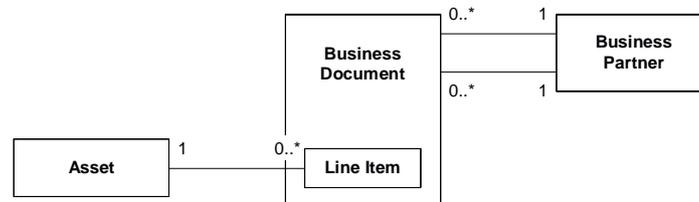


Fig.11. Contract Lines

## Acknowledgements

I wish to thank Christian V. Scheller of Simcorp for discovering the importance of this pattern, and Henning Kjærsgaard Nielsen, Jesper Kiehn and Lars Hammer of Navision a/s, for their valuable contributions.

## References

- [1] Coad, P., Lefebvre, E., DeLuca, J.: Java Modeling in Color with UML, Prentice Hall PTR, 1999.
- [2] David, J. S.: Three events that defined an REA methodology for systems analysis, design and implementation
- [3] Ericsson, H., Penker, M.: Business Modeling with UML, John Wiley & Sons, 2000.
- [4] Fowler, M.: Analysis Patterns, Addison Wesley Longmann, 1997.
- [5] Guido L. Geerts and William E. McCarthy: The Ontological foundation of REA Enterprise Information Systems, 1999-2000.
- [6] Hay, D.: Data Model Patterns, Conventions of Thought, Dorset House Publishing, 1996.
- [7] Silverston, L., Inmon, W. H., Graziano, K.: The data model resource book, John Wiley & Sons, Inc. 1977.
- [8] William E. McCarthy: The REA Accounting model: A generalized framework for accounting systems in a shared data environment. The accounting review (July) pp. 554-578, 1982.