3. Design Features of the Contract Data Model

3.1 Introduction

Building on the start made in the previous chapter, part two of this research consists of a description of the functional architecture devised as a solution to research objective one, as outlined in Chapter 1, Section 1.4.

> ***Research Objective 1:*** To propose a data organization framework allowing the storage of *ex post* and *ex ante* accounting data on BPIs suitable for supporting *changing* accounting information requests defined by *different* users.

The functional architecture is discussed in two consecutive chapters, this chapter being the second. The focus of this chapter is on exploring the data elements which can be captured via a 'contract', which is the central basis upon which the new accounting data model should be defined, as explained in Chapter 2. An outline will be provided of the relevant functional design principles of essential data components that conform together with the functional requirements of business process instance[14] data storage as described in Section 2.3 of Chapter 2. This chapter provides an answer to research question two of research objective one as outlined in Section 1.4 of Chapter 1.

> ***Research Question 2:*** What are the essential data components of the BPI data pattern required to service new and changing information needs?

The previous chapter focused on outlining the functional requirements of BPI data storage in general and introduced 'contracts' as central to a new accounting data model. The following requirements for data model design were described in Section 2.3. *1)* Objectivity of BPI data, *2)* data on a *single* BPI to be defined coherently, *3)* data between *different* BPIs to be defined coherently, *4)* data on business BPIs are *ex post* and *ex ante* and *5)* data are required on *internal* and *external* BPIs.

This chapter focuses on the definition of design features that detail the essential data characteristics of the information to be supported by the data model design to meet requirements 2, 3 and 4 as described above. The design of the contract data model itself using an accepted modelling technique is discussed in Chapter 4. Satisfaction of requirement 1 ('Only objective BPI data can be stored') is a precondition applicable to all design features. It relates to the fact that application artefacts such as debit-credit, general ledger account classification, etc. have to be explicitly excluded from the data model to meet the requirement of data objectivity (see also Everest and Weber, 1977; McCarthy, 1979; Verdaasdonk, 1998, p. 40, Chapter 2 Section 2.3). Requirement 5 ('Data are required on internal and external BPIs') is a data storage recommendation applicable at implementation level comparable to the 'Grundrechnung' data recording principles as described by Riebel (1994). This implementation recommendation has no impact on the definition of design features. In other words, no distinction is drawn between the storage of internal and external BPIs, rendering

[14] Business Process Instance is abbreviated in the remainder of this chapter as 'BPI'.

unnecessary the definition of additional design features to store data on internal BPIs differently.

This chapter proceeds as follows. The design features of single contract data storage are first explained in Section 3.2. Subsequently, an elaboration of the design features of the relationships between contracts is presented in Section 3.3. Next, in Section 3.4, an extensive illustration is provided of data capture using contract features. Afterwards, in Section 3.5, data storage using the contract data model is compared to data storage using the REA model on the one hand, and to data storage following 'Grundrechnung' principles on the other. Finally, some conclusions are formulated in Section 3.6.

3.2 Design features of single contract data storage[15]

3.2.1 Contracts versus clauses

BPIs specify the exchange of resources over their life cycle as formal agreements, recorded in 'contracts' (see Section 2.5 of Chapter 2). A contract was defined in Section 2.5 as '*a collection of contract clauses, where a contract clause is the expression of an exchange of resources or value-adding activities, executed through one or more fulfilments according to the clause terms*'. When considering a contract from a design perspective, it is a collection of mechanisms for grouping one or more contract *clauses*. Contract *clauses* contain the detail of resource exchanges. An 'exchange' is described by Ijiri (1975, p. 61) as '*an action whereby the entity foregoes control over some resources in order to obtain control over other resources*'. Contract *clauses* contain the detail of resource exchanges. For example, the formal data storage on the exchange of a garden set in a purchase contract can be detailed as three individual purchase contract clauses. Clause 1 details the procurement of a table, clause 2 details the procurement of 4 chairs and clause 3 details the procurement of a tablecloth. This exchange is illustrated in Figure 3-1. In information system design, it is important to know the *directions* of the various resource flows within the exchange to define the position of the contract clause participants explicitly. An 'exchange' involves the *supply* of one set of resources corresponding to the *demand* for another set of resources. Determining which participant 'supplies' or 'demands' presupposes the definition of a viewpoint. This can be illustrated by an example. Person A buys a garden set from person B. The transaction can be viewed from either person's perspective. From A's viewpoint, the situation can be explained as follows. A has to supply money to B and demands a garden set. From B's viewpoint, the opposite occurs. B has to supply the garden set and demands money from A. Following this approach, an exchange can be described as a specific example of the 'give' and 'take' relationship as described by Geerts and McCarthy (1997).

The resource types used in this exchange determine the contract clause *type*. A resource is defined by Ijiri (1975, pp. 51-52) as '*an object that is scarce and has utility and is under the control of an enterprise*'. Two types of resources can be distinguished, namely operational resources and financial resources. Operational contract clauses (e.g. purchase contract clauses, sales contract clauses, lease contract clauses, etc.) typically focus on the exchange of operational resources for financial resources. For example, in the BPI of a purchase contract clause, procured materials are exchanged for money. Operational contract clauses can also yield BPIs where only operational resources are involved, or where a mixture of resource types are exchanged. An example of the latter is where a new car is bought and exchanged for an old car and a surplus of money. Operational resources can be goods as well as services.

[15] A 'single contract' is described here as one contract in isolation, an 'unrelated' contract.

Financial contract clauses only focus on the exchange of different types of financial resources, e.g. loan contract clauses, hedging contract clauses, etc. In addition to the 'exchange' of financial resources, the 'conversion' of resources can be identified as an event. In a 'resource exchange' BPI, resources are transferred to other participants (in terms of possession or full ownership) but do not disappear (e.g. when a car is exchanged for money, both the car and the money remain in existence but are possessed or owned by others), whereas in 'resource conversion', resources are only 'hired out' to other participants who perform some activity on these resources. The end result is that the resources disappear in their original form, being *converted* into new resources. The converted resources are returned to their original owner. Conversion can be applicable to financial and operational contracts. An example of an operational conversion is a blue assembly that gets repainted black and becomes a black assembly. The original blue assembly is not available anymore. An example of financial conversion is a sum of Japanese Yen (JPY) converted to the equivalent sum of USD. The initial JPY sum is not available anymore. It should be noted that conversion of financial resources frequently occurs, is inexpensive and easily applicable, while conversion of operational resources can be an option but frequently turns out to be more expensive when tried.

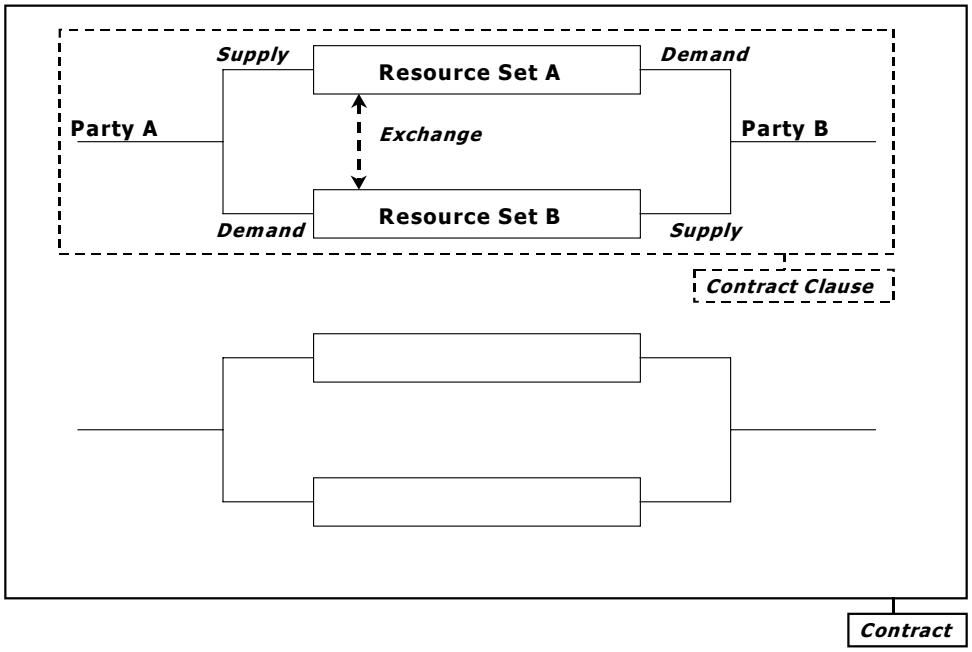

Figure 3-1. Contract versus contract clause and exchange

3.2.2 Roles of Clause Participants

Two different roles can be associated with participants defined in a clause, that of 'party' and 'opposing party'. One or more participants can be grouped under either the 'party' or 'opposing party' role, depending on the contract clause type. This is visualized in Figure 3-2. If the clause has reciprocally balanced implications for the two participants, one in the 'party' role and the other in the 'opposing party' role, it is described as a *bilateral* clause. If more than two participants are involved, with at least one as 'party' and one as 'opposing party', then the clause is described as *multilateral*. In a typical multilateral contract clause, more than two participants are defined but mutual rights and obligations can be grouped in pairs. For example, suppose that three participants are involved, participants A, B and C. A delivers resources to B, B delivers resources to C and finally, C pays A for the resources. This situation illustrates that multilateral clauses can be redefined as groups of bilateral clauses. Where two or more participants are involved as party or opposing party, where participants

acting in one or other role can exert influence over participants in the opposite role by defining the essential rights or obligations, the associated cause is described as *unilateral*. Participants filling the 'opposite party' role do not have to be named explicitly, e.g. when the government votes a tax rule for a specific type of organization where the individual organizations to which the tax rule is applicable are not explicitly named as clause participants. Bilateral and multilateral contract clauses frequently occur within the scope of contracts relevant to servicing the information needs of users in manufacturing organizations. Most unilateral contract clauses concern tax, social or environmental regulations.

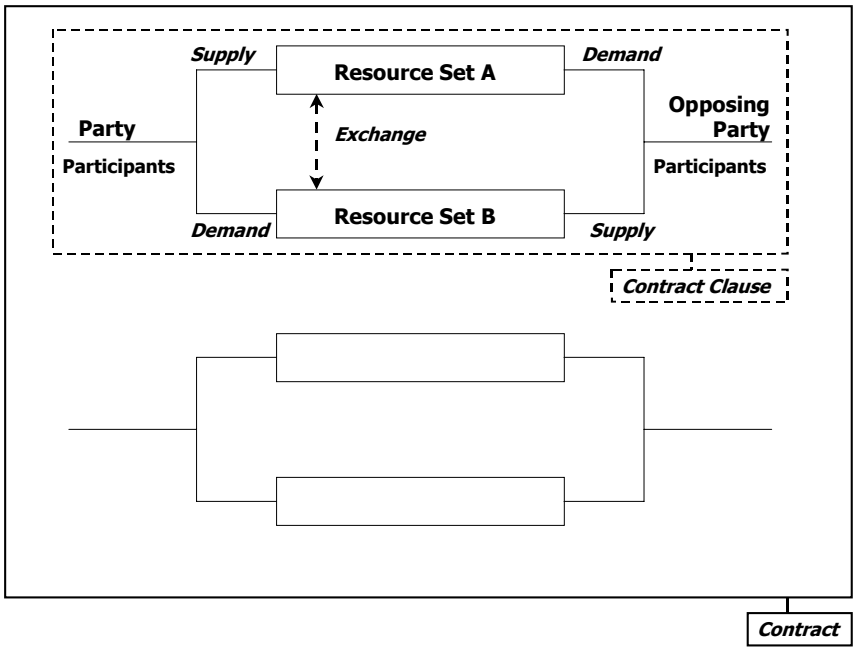

Figure 3-2. Roles of contract-clause participants

Clause participants are called 'persons' in legal theory. A 'person' can be a natural person (a human being) or a legal entity (e.g. a legal corporation, a legal partnership, etc.). Legal entities must always be represented by at least one natural person. Webster's dictionary defines a 'person' in this context as '*one (as a human being, a partnership, or a corporation) that is recognized by law as the subject of rights and obligations*'.

3.2.3 Different types of Terms

Earlier, in Section 3.2.1, it was explained that resource exchanges are detailed in contract clauses. Section 2.4 of Chapter 2 discussed that a particular limitation of double-entry bookkeeping data is the fact that data availability is restricted to the financial outcome of a business transaction. Data on the circumstances and conditions within which the business transaction was agreed upon or executed are lost. In the proposed contract-based data organization approach, the conditions under which the exchange takes place are described in the contract *terms*, detailed per contract clause. Webster's dictionary describes 'terms' as '*provisions that determine the nature and scope of an agreement*'. Two types of terms can be discerned, clause terms and abrogation terms. *Clause terms* determine mutual rights and obligations on executing the agreed exchange. *Abrogation terms* detail the rights and obligations contract clause participants have should one or more participant default. Clause and abrogation terms can be expressed in terms of the 'rights' and 'obligations' of their participants. This is visualized in Figure 3-3.

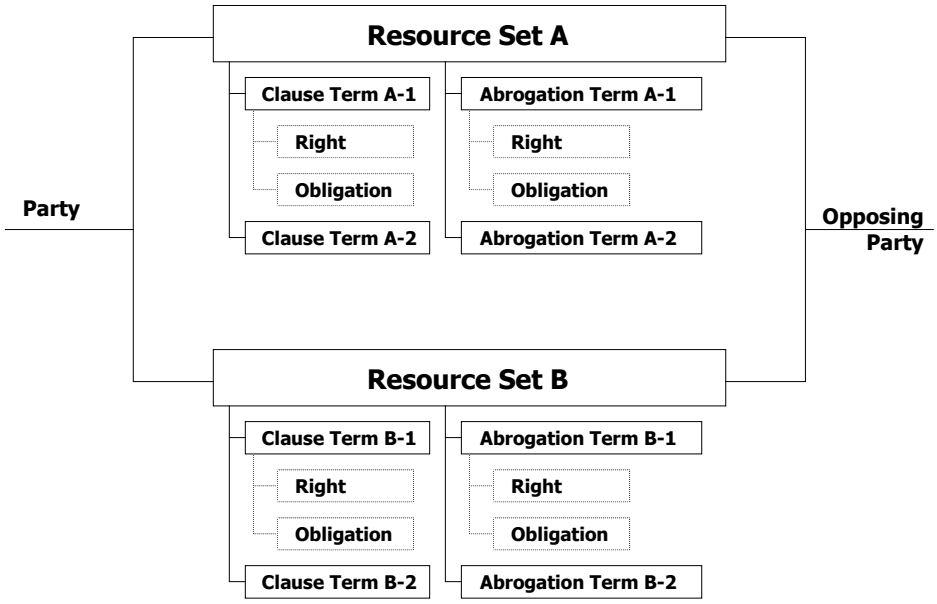

Figure 3-3. Clause and Abrogation Terms

Webster's dictionary describes 'rights' as '*Something to which one has a just claim: the power or privilege to which one is justly entitled*', and 'obligations' as '*Something (as a formal contract, a compromise or demand of conscience or custom) that obligates one to a course of action*'. 'Terms' can only be defined for bilateral and multilateral contract clauses. They are said to be 'balancing' in the sense that the rights of participants defined in one role should always equal the obligations of participants defined in the opposite role. For instance, where A is obliged to pay a sum of money, B has the right to receive the sum of money.

3.2.4 Clause Life-Cycle Aspects

A contract clause passes through different phases during its life cycle[16]. The life-cycle phases are punctuated by clause state changes. During the clause life cycle, the following two states can be discerned.

- *Planned State*. The planned clause phase is the phase during which the contract clause has already been proposed but not yet agreed upon. This means that various alternate versions of a clause can exist. A planned clause can be regarded as a simulation of the committed clause. Examples of planned state contract clauses include purchase quotations, loan quotations, hedging quotations, etc.
- *Committed State*. The committed clause phase starts when one clause is chosen and agreed upon out of the various clause proposals. Examples of committed clauses include purchase orders, sales orders, loan agreements, hedging orders, etc.

Once a clause is committed, it is executed following the procedures detailed in the clause terms. The various clause states are visualized in Figure 3-4.

[16] Here, focus is only on functional phases. Technical phases like creation, deletion, archiving, etc. are therefore ignored.

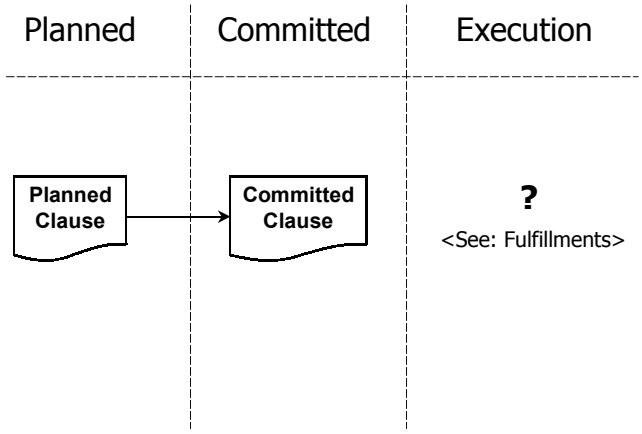

Figure 3-4. Definition of different clause phases

A uniform framework can be applied to describe the relevant contract clause dates and their implications, independent of the clause state. This is visualized in Figure 3-5.

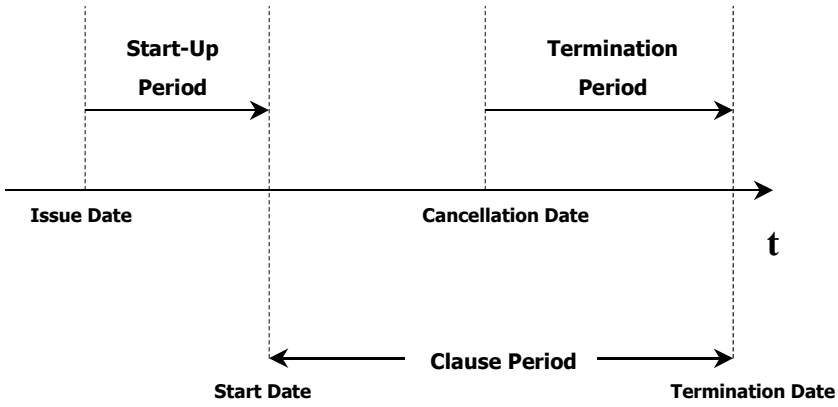

Figure 3-5. Relevant clause dates and clause periods (Sources: Corbey and Tullemans, 1991; Theeuwes and Adriaansen, 1994; Verdaasdonk, 1998)

The following clause periods and clause dates[17] can be distinguished (definitions are adapted from Corbey and Tullemans, 1991; and Theeuwes and Adriaansen, 1994).

- *Clause Issue Date*. The date at which all participants involved intend a resource exchange (planned clause) or agree upon a resource exchange (committed clause)
- *Clause Start Date*. The date at which a resource is expected to be available (planned clause) or committed to be available (committed clause)
- *Clause Cancellation Date*. The date at which a clause is expected to be cancelled (planned clause) or a clause is decided to be cancelled (committed clause)
- *Clause Termination Date*. The date the resource will no longer be available. The actual termination date is the date at which the other clause participants are notified (applicable

[17] Earlier, in Section 3.2.1, a contract was defined as collection of contract clauses. This means that depending on the implementation approach chosen, the dates and periods apply to individual contract clauses or to the entire contract.

to committed clauses). Otherwise the termination date is planned (applicable to planned clauses)

- *Clause Start-Up Period*. The time needed between the date at which the clause is negotiated (planned or committed) and the moment at which the resource becomes available (expected to be available for planned clauses or committed to be available for committed clauses)
- *Clause Termination Period*. The time needed between the demand for termination of a clause (the cancellation date, planned or committed) and the actual termination date of the clause (planned or committed)
- *Clause Period*. The period when resources become available, the interval between the clause start date (planned or committed) and the clause termination date (planned or committed).

It should be re-emphasized that the framework described above can be applied to 'planned' or 'committed' state contract clauses.

3.2.5 Clause Fulfilments

The conditions and circumstances under which a contract clause has to be executed are defined in the clause terms of a committed contract clause. The design features to capture essential data on contract clause *execution* should be defined against a recurring 'clause execution' pattern. This recurring pattern should characterise the commonality in clause execution features recognised across different contract clause types. A usable pattern can be found by investigating some examples of contract clause execution.

- *Execution of a procurement-contract clause.* The execution of a procurement contract clause details the delivery and payment of goods or services. Details on delivery are captured in the 'delivery note' whereas details on the payment execution are defined on a 'cheque', a 'draft' or whatever other payment document is used.
- *Execution of a loan-contract clause*. The execution of a loan contract details the availability of the amount borrowed on a particular date, the periodic interest payments and the repayment of the loan on its expiration date. The document detailing the availability of the loan is the 'bank statement' that details the loan amount (on the debit side). The document detailing the periodic interest payments is the 'bank statement' where the interest paid is recorded (on the credit side).

The recurring pattern for storing data on the execution of contract clauses as derived from the examples given above are different types of 'documents'. These documents are further described as 'fulfilment documents' and depend on the contract-clause terms. Comparable to the 'contract' where contract clauses hold the data on resource exchanges and the contracts themselves only 'group' the contract clauses, data on contract clause execution are stored in one or more fulfilment lines. From a design point of view, the fulfilment documents themselves are merely groupings of fulfilment lines. Looking forward, this is termed the 'Fulfilment Pattern', as visualized in Figure 3-6.

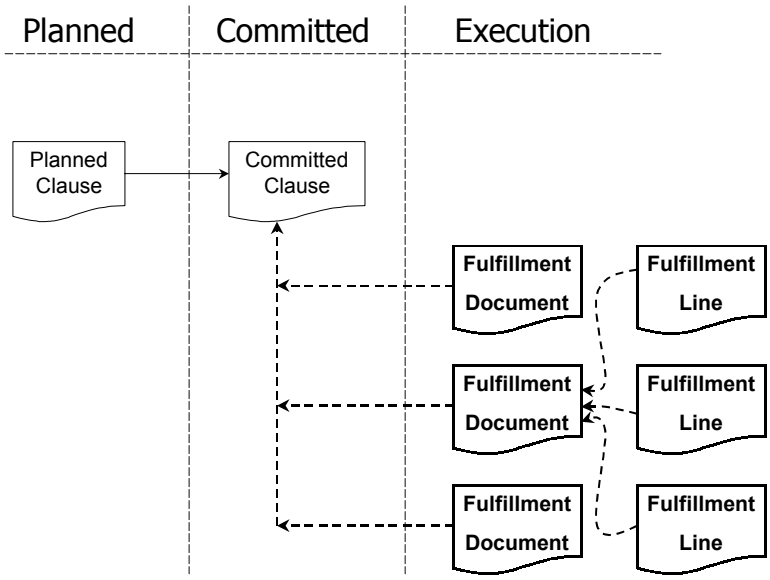

Figure 3-6. Definition of fulfilment documents to store data on contract clause execution

3.2.6 Summary: Overview of design features for single contract data storage

Sections 3.2.1 to 3.2.5 outlined design features of single contract data recording. A categorization of design features per BPI data storage requirements as defined in Section 3.1 and Section 2.3 of Chapter 2, will now follow.[18]

Requirement A: Data on a *single* BPI to be defined coherently

- *Design Feature 1*. Contracts are a grouping of contract clauses. Contract clauses contain data on resource exchanges.
- *Design Feature 2*. Contract clause participants are divided into two roles, 'party' or 'opposing party'.
- *Design Feature 3*. Clause participants can be natural persons or legal entities. The latter have to be represented by at least one natural person.
- *Design Feature 4*. Agreements on resource exchange execution are detailed in 'clause terms' whereas default procedures are covered in 'abrogation terms'.
- *Design Feature 5*. 'Terms' (irrespective of type) are defined as balancing participant 'rights' and 'obligations'.
- *Design Feature 6*. Contract clauses can be in either a 'planned' or a 'committed' state over their life cycle.

Requirement B: *Ex post* as well as *ex ante* data on single BPIs have to be defined

- *Design Feature 7*. *Ex ante* BPI data are recorded using 'planned' state contract clauses.
- *Design Feature 8*. *Ex post* BPI data are stored using 'committed' state contract clauses and one or more 'fulfilment lines'.
- *Design Feature 9*. Fulfilments are a grouping of fulfilment lines. Fulfilment lines contain data on the execution of resource exchanges.

[18] Design features meeting requirements 2 and 4 as described in Section 3.1 are summarized.

New data models have to comply with all the design features of requirements A and B *together* to ensure full accommodation of single BPI data across the entire projected time window.

3.3 Design features facilitating contract portfolio definition

Contracts relevant for data provision do not exist in isolation but have to be defined and stored structurally to mirror reality (see Section 2.6.4 of Chapter 2). From a semantic perspective, the structure found between contract clauses can be envisaged as a portfolio consisting of related contracts. The following sections explain how these different types of relationships can be translated into design features. Earlier, in Section 3.2.1, it was explained that a contract is only a grouping of contract clauses from a design perspective. Relevant relationships characterizing the portfolio are therefore only defined at contract clause level and not at contract level.

3.3.1 Hierarchical relationships between contract clauses

Contract clauses can be defined at different levels. Section 2.6.4.2 of Chapter 2 illustrates the generic characteristic of multi-level contract definition through an example of financial contracts defined at 3 different levels. Schneeuweiss (1995) discussed the interdependencies between different levels in a hierarchical structure. His findings are applied here to multi-level contract administration. Schneeuweiss explained that simple top-down relationships are too simplistic to describe the relationships in a hierarchical system. He therefore distinguishes three stages of interdependence between the aggregate level and the detailed level, namely *anticipation, instruction* and *reaction*. *Anticipation* describes a top-level contract using a number of relevant (aggregate) characteristics of the base-level contracts. The outcome of the top-level contract administration is passed on to the base-level contract administration as an *instruction*. This in turn may lead to a *reaction* from the base level. The contract storage processes at both levels can be iterative. It should be stressed that the contract clause portfolio is a clear example of multi-level contract administration. In this respect, the terms ‘top level’ and ‘base level’ as used by Schneeuweiss can be redefined at different levels. Hierarchical relationships in the contract clause portfolio are illustrated in Figure 3-7.

Two different contract clause types can be distinguished in the portfolio – overall contract clauses and contract clauses at the lowest level. Contract clauses defined at the lowest level record commitments on real resource exchanges. An example of this type of contract is the exchange of a car for money. Overall contract clauses are contract clauses where the agreed conditions of possible future resource exchanges are recorded. Examples of overall contracts include agreements with suppliers on the prices of cars for the coming year, agreements with financial institutions on the rate for certain banking services like foreign payments, etc. Overall contract clauses can be further detailed (consumed) as overall contract clauses at a lower level, or as contract clauses defined at the lowest level. Depending on the level at which overall contract clauses are defined, they can cover several levels of detail (e.g. five years planning at the top level, annual planning at the second level, etc.), or they can detail particular resources at another level of detail (e.g. the general category ‘cars’ at the top level becomes individual family cars like Mercedes, Volvo, etc., one level down).

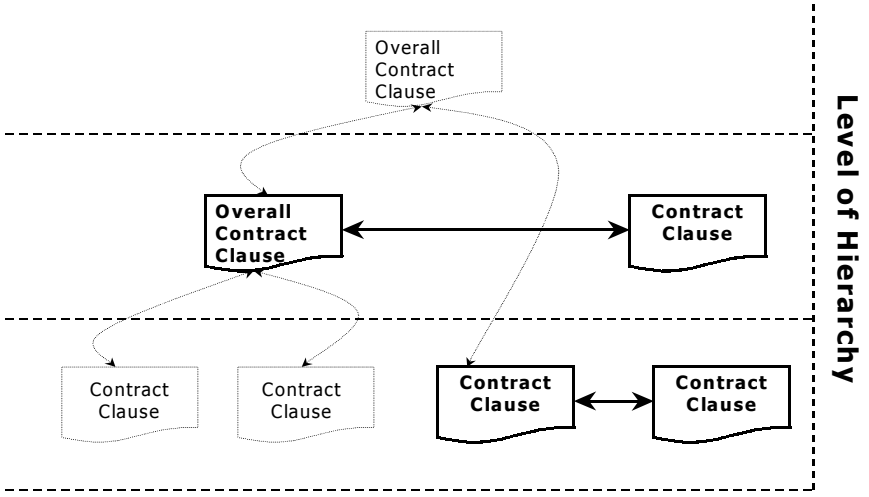

Figure 3-7. Hierarchical relationships between contract clauses

3.3.2 Horizontal relationships between contract clauses

Horizontal relationships defined between contract clauses represent a mutual dependency between related contract clauses. Three different possible definitions of horizontal relationships can be discerned. Firstly, a horizontal relationship defined between two overall contract clauses (e.g. a year purchase contract related to a year hedging contract). Secondly, a horizontal relationship defined between an overall contract clause and a contract clause at the lowest level (this situation is presented in Figure 3-8). An example of this situation could be the relationship between a year purchase contract and a specific financing contract. Thirdly, a horizontal relationship defined between two contract clauses at the lowest level (this situation is also illustrated in Figure 3-8), like an individual purchase order for which a specific hedging contract is defined. Horizontal contract relationships are presented in Figure 3-8.

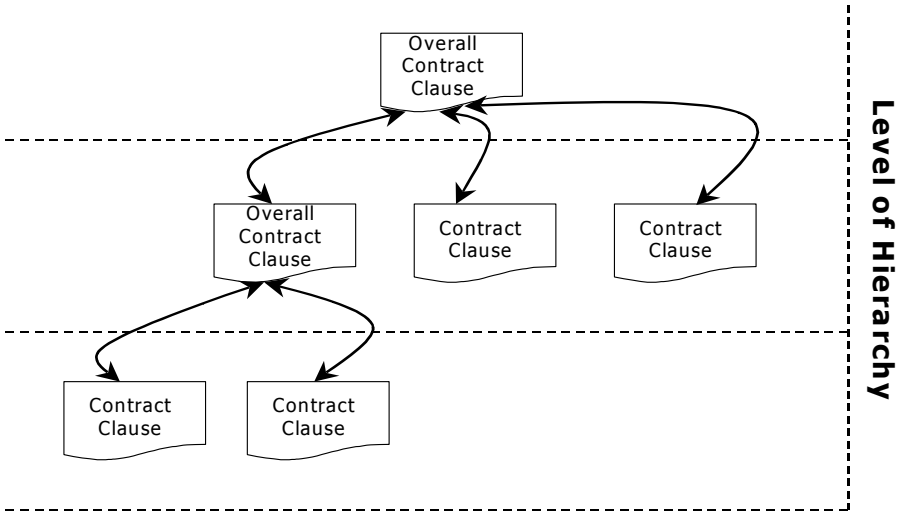

Figure 3-8. Horizontal relationships between contract clauses

3.3.3 Summary: Overview of design features for contract clause portfolio definition

Relationships are defined between contract clauses and not at contract header level as a result of design feature 1 (see Section 3.2.6) which prescribes that 'contracts' are only grouping mechanisms for one or more contract clauses. The following additional design features can now be defined to fulfil the remaining requirement as described in Section 3.1[19].

Requirement: Data between *different* BPIs to be defined coherently

- *Design Feature 10*[20]. Relationships (irrespective of type) can be defined between *a)* two contracts at the lowest level, *b)* two overall contracts or *c)* an overall contract and a contract at the lowest level
- *Design Feature 11*. Hierarchical relationships between contract clauses should be defined
- *Design Feature 12*. Horizontal relationships between contracts should be defined.

These three design features are required to support data provision for the contract clause portfolio.

3.4 Example of BPI data storage following the Contract Data Model

3.4.1 Illustration of single contract data storage

The objective of the design features defining the contract data model is to prescribe a generic yet specific approach to record BPI data phenomena in such a way that stored data are sufficiently complete and not influenced by artefacts applicable in a focused application scope as explained by Sakagami (1995). This is illustrated below using a representative example of how design features of the contract data model are used to store BPI data in practice. Design features are mentioned in brackets and refer to Sections 3.2.6 or 3.3.3. The design features pertaining to single contract data storage are illustrated next, using the following example.

The 'Wheelchair Corporation' organization assembles and sells wheelchairs. One of its customers is the 'Office Supplies' organization. The following sales transaction occurs: the 'Wheelchair Corporation' organization sells 1,000 wheelchairs to the 'Office Supplies' organization at USD 250 per wheelchair. The sales transaction can be detailed by the following events through its life cycle.

- 2 February, there is a call for quotation from a customer, 'Office Supplies'.
- 1 March, 'Office Supplies' receives a quotation for 1,000 wheelchairs at USD 250 per piece. Delivery is planned for 30 March and payment by cheque is planned 14 days after delivery.
- 10 March, 'Office Supplies' accepts the quotation with some small changes and the 'Wheelchair Corporation' receives an order. Delivery is agreed upon by 30 March. The wheelchairs have to be delivered to Mr Johnson, Warehouse Receipts Officer, at the 'Office Supply Warehouse' location. Mr Peters, shipment officer for 'Wheelchair Corporation', is in charge of wheelchair shipment. 'Office Supplies' negotiated payment 30 days after delivery by draft payment. Payments will be made by the 'Office Supplies' Payment Department and have to be sent to the 'Wheelchair Corporation' Central Department for Customer Collections.
- 30 March, the Shipment Department of the 'Wheelchair Corporation' sends 1,000 wheelchairs to 'Office Supply Warehouse' (dispatched by Mr Peters), received by Mr Johnson, Warehouse Receipts Officer. A delivery note provided.
- 30 March, a sales invoice from the 'Wheelchair Corporation' Accounts Department arrives at the 'Office Supplies' Accounts Department. The total invoice amount is USD 250,000.
- 30 April, the 'Office Supplies' Payment Office directs a draft payment to the 'Wheelchair Corporation' Central Department for Customer Collections to settle the invoice. The sales transaction is fully settled.

Situation 3-1. Example of a single BPI

[19] Design features as a solution to requirement 3 as described in Section 3.1 are summarized here.
[20] Design features 1 to 9 are defined in Section 3.2.6.

Contracts, Contract Clauses, Exchanges, Roles of Contract Clause Participants
Situation 3-1 concerns the exchange of 1,000 wheelchairs for USD 250,000. The exchange itself is stored in a contract clause, as 'contracts' are only grouping mechanisms for contract clauses [design feature 1]. The exchange takes place between the 'Wheelchair Company' organization (defined in 'party' role) and the 'Office Supplies' organization (defined in 'opposing party' role) [design feature 2]. Both organizations are legal entities consisting of departments represented by natural persons [design feature 3]. As operational resources (wheelchairs) are exchanged for financial resources, this exchange is a typical example of a standard operational contract. The 'Wheelchair Company' organization *supplies* 1,000 wheelchairs for which it *demands* USD 250,000. From the perspective of 'Office Supplies', the opposite applies. 'Office Supplies' *supplies* USD 250,000 and *demands* 1,000 wheelchairs. As two participants are specifically involved, this is an example of a bilateral contract. This is illustrated in Figure 3-9.

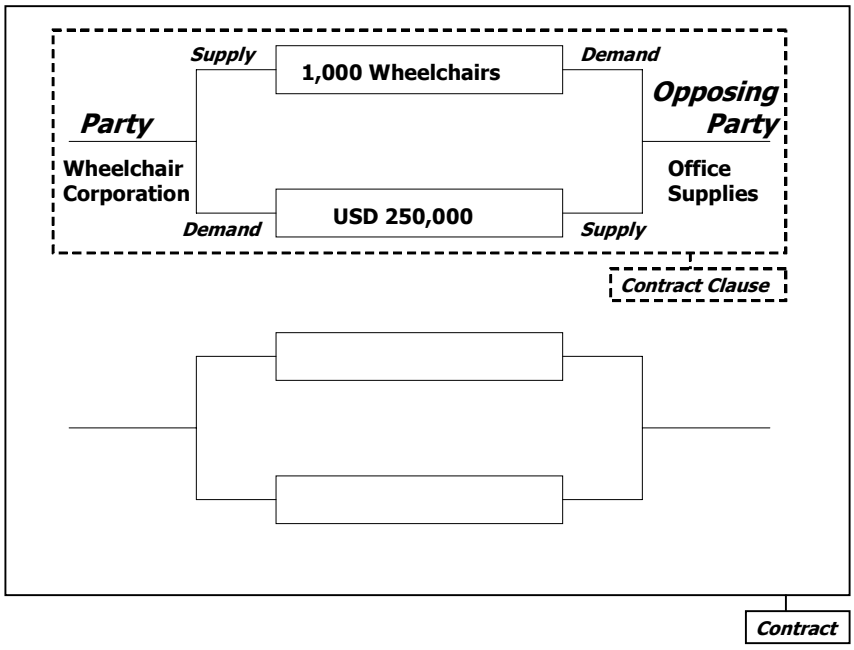

Figure 3-9. Example: contract, contract clause, exchange and roles

Clause terms and abrogation terms
Terms can be detailed as clause terms and abrogation terms. In Situation 3-1, clause and abrogation 'terms of delivery' and 'terms of payment' are specified [design feature 4]. Delivery terms are detailed as follows. Office Supplies has the right to delivery of 1,000 wheelchairs at the 'Office Supply Warehouse' location headed by Mr Johnson by 30 March, while Wheelchair Company has the obligation to deliver the above from its 'Shipment Department' headed by Mr Peters by 30 March 1999 [design feature 5]. The corresponding abrogation terms are detailed as follows. Office Supplies has the right to demand enforced delivery in court in the event that the actual delivery date is exceeded by more than 5 days. Wheelchair Company is obliged to pay a penalty for non-delivery on court request. The payment terms are defined as follows. The customer collection department of Wheelchair Company has the right to receive payment by draft from Office Supplies while the payment department of Office Supplies has the obligation to pay USD 250,000 by draft. The corresponding abrogation terms are detailed as follows. Wheelchair Company has the right to demand enforced payment in court 60 days after the payment due date whereas Office Supplies is obliged to pay a penalty for late payment on court request. Clause and abrogation terms applicable to Situation 1 are visualized in Figure 3-10.

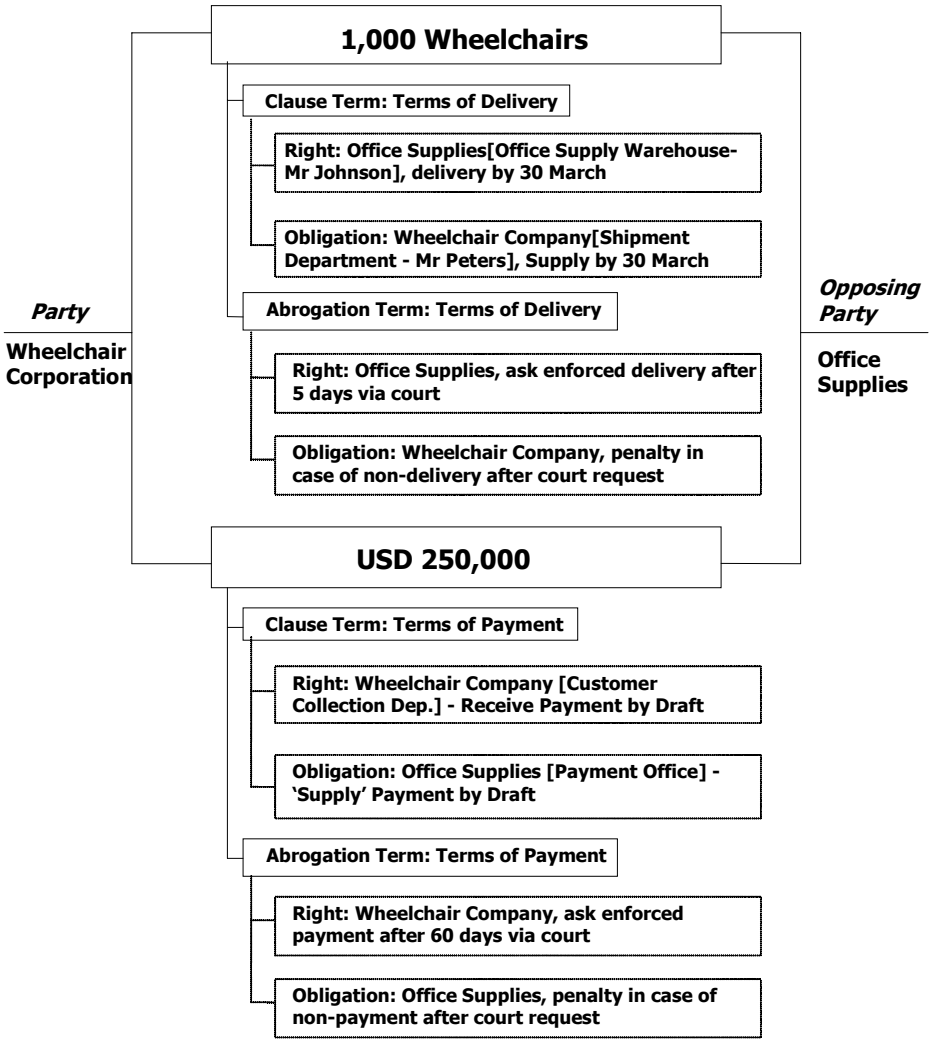

Figure 3-10. Example: contract clause and abrogation terms

Clause Fulfilments

From the 'Wheelchair Corporation' organization's perspective, the following clause fulfilments are defined.

- On 30 March, a *delivery note* detailing the delivery of 1,000 wheelchairs to the 'Office Supplies' location, in execution of the clause delivery terms
- On 30 March, a *sales invoice* for USD 250,000 sent to the Accounts Department of the 'Office Supplies' organization
- On 30 April, a *draft payment* of USD 250,000 received by the Central Department for Customer Collections of the Wheelchair Company.

This contract clause in its various phases and the fulfilments is visualized in Figure 3-11.

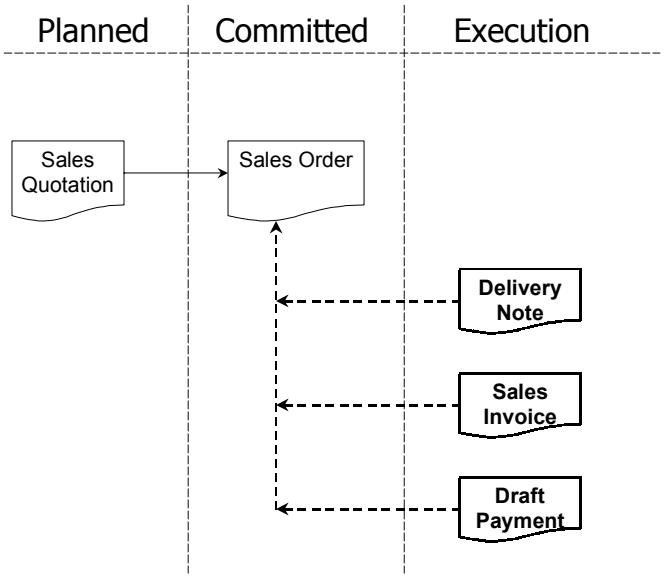

Figure 3-11. Illustration: Fulfilments

3.4.2 Illustration of contract portfolio administration

The following example is used to illustrate how design features of the contract portfolio are applied.

The 'Office Supplies' organization makes the following projection on the number of wheelchairs needed for the coming year. The following events occur, from the moment of making the year plan to the actual ordering of wheelchairs.

- 1 December 1999, the procurement manager and the sales manager agree that 240,000 wheelchairs will be sold in year 2000 and will need to be available. They agree on a linear, time-phased availability of 20,000 wheelchairs per month.
- 15 December 1999, the procurement manager of 'Office supplies' negotiates a year purchase contract with the sales manager of 'Wheelchair Company' for 120,000 wheelchairs for a fixed price of USD 250 per wheelchair. This contract starts on 1 January 2000 and ends on 31 December 2000. They agree that a maximum of 10,000 wheelchairs can be bought at this price per month. In the last period (December 2000), uncalled quantities have to be purchased.
- 20 December 1999, a hedging agreement is negotiated with 'United Bankers' bank covering the currency risk involved in the year purchase contract with 'Wheelchair Company'.
- 1 March 2000, 'Office Supplies' receives a quotation for 10,000 wheelchairs at a price of USD 250 per wheelchair from 'Wheelchair Company'[21]. This contract 'consumes' the year purchase contract committed on 15 December 1999.
- The number of chairs that can be bought using the year purchase agreement is insufficient. On 15 March 2000, a new order for 500 wheelchairs is placed with the 'Super Wheelchair Company' organization. The price per wheelchair is USD 285.
- 20 March 2000, a currency hedge contract is negotiated with 'United Bankers' covering the individual order from 'Super Wheelchair Company'.

Situation 3-2. Example of the definition of a contract portfolio.

[21] Note: The detail of this contract is described in Section 3.4.1 and was used to illustrate the design of a single contact.

In Situation 3-2, the following contracts were defined.

(1) *Overall Contract, Master Plan*. This contract was committed to on 1 December 1999 and details the agreement between the procurement manager and the sales manager on the availability of 240,000 units of the 'wheelchair' resource. This contract covers the year 2000.

(2) *Overall Contract, Year purchase contract*. This contract was committed to on 15 December 1999, between the procurement manager of 'Office Supplies' and the sales manager of 'Wheelchair Company'. They agreed on the availability of 120,000 units of wheelchair type X, 10,000 units per month at a fixed price of USD 250 per unit.

(3) *Contract at the lowest level, Purchase Order A-1*. A quotation for this purchase order was received on 1 March 2000. This is a consumption of the Overall Contract, Year purchase contract, detailed in **(2)**.

(4) *Contract at the lowest level, Purchase Order B*. This purchase order was made on 15 March 2000 and directly consumes the Overall Contract, Master Plan.

(5) *Contract at the lowest level, Hedging Agreement A*. This hedging agreement was made on 20 December 1999 with the 'United Bankers' bank.

(6) *Contract at the lowest level Hedging Agreement B*. This hedging agreement was made on 20 March 2000 with the 'United Bankers' bank.

The following hierarchical and horizontal relationships were defined in Situation 3-2.

Hierarchical Relationships

The following hierarchical relationships were defined[22].

- Hierarchical relationship between (1) and (2). This relationship represents the consumption of the master plan (1) with 120,000 wheelchairs, specified in year purchase contract (2)
- Hierarchical relationship between (2) and (3). The relationship represents the consumption of the year purchase contract (2) with 10,000 wheelchairs, defined by a purchase order (3)
- Hierarchical relationship between (1) and (4). This relationship represents the consumption of the master plan (1) with 500 Wheelchairs, defined by a purchase order (4).

These contracts, and hierarchical relationships defined between these contracts, are visualized in Figure 3-12.

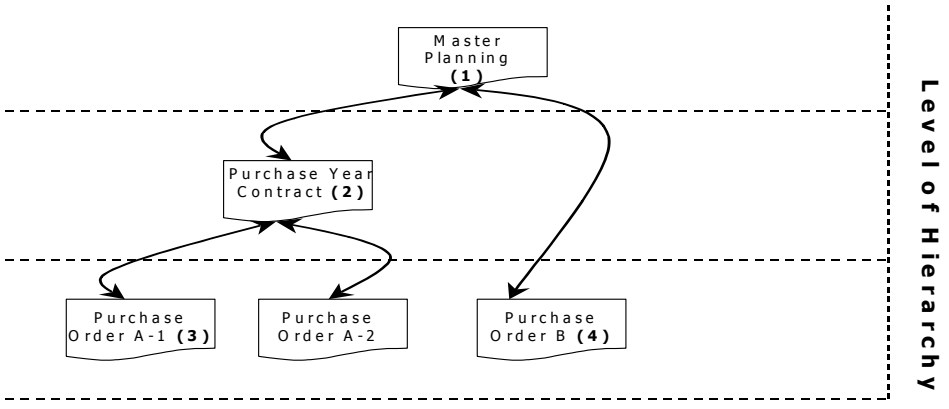

Figure 3-12. Example of Hierarchical Relationships between Contracts

[22] It should be noted that each of these contracts consists of contract clauses and that the hierarchical relationships are defined between contract clauses. However, as Situation 3-2 does not focus on the detail of single contract registration, no contract clauses are mentioned. Relationships are here defined between contracts.

Horizontal Relationships

The following horizontal relationships are defined in Situation 3-2.

- Horizontal relationship between (2) and (5). This relationship represents the mutual dependency between the 'year purchase contract' overall contract (2) and the 'Hedging Agreement A' lowest-level contract (5). Where the amount specified in one of the contracts changes and the other contract does not change accordingly, the year purchase contract is over or under hedged.
- Horizontal relationship between (4) and (6). This relationship represents the mutual dependency between the 'Purchase Order B' lowest-level contract (4) and the 'Hedging Contract B' lowest-level contract (6). The mutual dependency expresses the same implications as the previous example.

The horizontal relationships between contracts applied in Situation 3-2 are visualized in Figure 3-13.

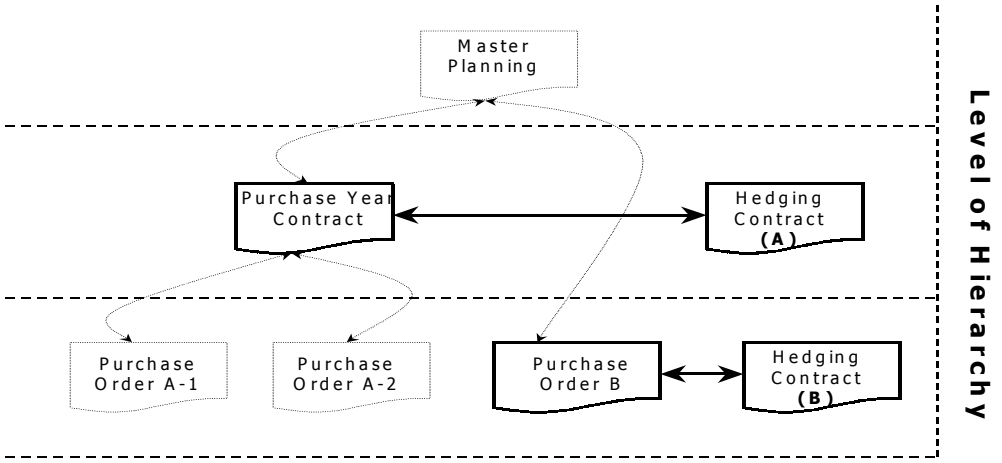

Figure 3-13. Example of Horizontal Relationships between Contracts

3.5 Contract Data Model compared to REA and 'Grundrechnung' data storage principles

3.5.1 Design features of the Contract Data Model compared to aspects of the extended REA Model

Since the original version of the REA model was released in 1982, there have been frequent publications in proceedings on research of the REA model (e.g. see: McCarthy, 1979, 1982; Geerts and McCarthy 1997), Geerts (1997). Design features of the contract data model have been compared to the latest versions of the REA model (i.e. the extended REA model, Geerts and McCarthy 2000; and the REA ontology description, Geerts and McCarthy, 2002). As the core of the REA model (i.e. the REA components and the relationships between REA components) has not changed over time, some of the contract design features are compared with aspects of the original REA model (McCarthy, 1982).

The most prominent enhancements of the extended REA model in comparison with the initial REA model are *1)* the addition of the ontological concept of 'commitment' – further differentiated into 'contracts' and 'schedules' in the REA process specification (Geerts and McCarthy, 2000), *2)* REA Tasks and *3)* the Meta Layer (i.e. the REA Process Configuration and the REA Task Configuration, Geerts and McCarthy, 2002 p. 4-5). Geerts and McCarthy have positioned these new aspects without detailing the data components relevant for adoption in the data model. The contract data model covers each of these three aspects.

First, the relationship between REA commitments (i.e. 'contracts' and 'schedules') and the contract data model is explained. The closest relationship between the two can be seen in REA commitment subtype 'contracts'. The REA contracts are similar to contract clauses at the lowest level in the contract data model. The 'schedule' REA commitment subtype is also supported in the contract data model, by defining contract clauses at various levels (i.e. the contract portfolio idea, see Section 3.4.1). In the contract data model, the REA schedule aspect is defined as a higher level contract clause with vertical relationships defined by relevant lower level contracts (design feature 11, Section 3.3.3). As Geerts and McCarthy (2000) do not provide detail on REA contracts and REA schedules, it is unclear whether or not they aimed at a different design for these two REA commitment subtypes. In the definition of the contract data model, there are clear reasons for not creating distinctions in defining the characteristics of a 'schedule' in the relationship between contract clauses. This approach offers advantages since both a 'schedule' and a lowest level contract clause can have common features such as contract clause terms, contract clause abrogation terms, etc., defined by expressing the conditions under which the 'plan' or 'execution' occurs. Geerts and McCarthy (2000) are unclear as to which additional characteristics the 'schedule' and 'contract' REA commitment should have in addition to the REA components and the REA relations applicable in the original version of the REA model. The most fundamental difference is that the extended REA model does not contain the features of contract terms and abrogation terms, therefore does not detail the *conditions* under which the business transactions and/or schedules are maintained. In the definition of the contract data model, this is considered as a fundamental data component (see design feature 4, Section 3.1, also Section 2.5 of Chapter 2).

Second, a comparison is presented of REA 'tasks' and the design features of the contract data model. The essential features of business transaction data using a systematic definition of Resources, Events and Agents, and the required relationships between these components is still the core of the REA model in its capacity as an accounting data model, and was the subject of one of the initial publications (McCarthy, 1982, p.561-562). This detail is available in but not repeated for the extended REA model.

- *Economic Resources*. The REA model in its basic format considers physical resources only. In the contract data model, the same scope of resources is applicable. Section 3.2.1 explained that two types of physical resources could be used in the resource exchanges detailed in contract clauses (design feature 1), i.e. operational resources and financial resources.
- *Economic Events*. The scope of economic events supported by the REA model extends to '[economic resources] *resulting from production, exchange, consumption and distribution*'. The contract data model covers a comparable scope. Production and exchange are defined via different contract clause types. Production is defined in this context as an exchange of value-adding activities. Consumption and distribution are detailed through hierarchical relationships. These relationships can be identified between an overall contract clause and a lower level overall contract clause (e.g. see example Section 3.4.1: hierarchical relationships expressing a master plan which is consumed by a year purchase contract), or an overall contract clause and a contract clause at the lowest level (e.g. see example Section 3.4.2: hierarchical relationship expressing a year purchase contract consumed by an individual purchase order) (design feature 10). The event type is defined by the contract clause type (e.g. procurement contract clause, sales contract clause, etc.).

- *Stock-Flow Relationships*. In the REA model, these relationships connect economic resources to economic events. The equivalent functionality in the contract data model, is the definition of resources with the direction given to a resource (supply or demand) viewed from a certain role for clause participants (design feature 2).
- *Duality*. This is used in the REA model *'to link each increment in the resource set of the enterprise with a corresponding decrement*' (also Ijiri 1975, Ch.5). Corresponding functionality in the contract data model, is supported by the definition of the conditions of a resource exchange (as defined in a contract clause – see design feature 1) via contract clause terms (design feature 4). Contract clause terms are always defined as rights and obligations of participants in different roles (design feature 5).
- *Economic Agents*. These are defined in the REA model as '*persons and agencies who participate in the economic event of the enterprise or who are responsible for subordinates' participation*.' The equivalent of 'Agents' in the contract data model, is the definition of 'participants'. In design feature 2, it was explained that participants could be defined in the role of 'party' or 'opposing party'. The 'role' aspect is not supported in the REA model.
- *Economic Units.* In the REA model, 'units' are defined as '*inside participants: agents who work for or are part of the enterprise being accounted for*.' The contract data model considers 'REA Units' as a specific category of 'participants', i.e. internal participants (see also Section 2.6.3 of Chapter 2).
- *Control Relationships.* These are defined as '*3-way associations among (1) a resource increment/decrement (event), (2) an inside party (unit), and (3) an outside party (agent)'*. In the contract data model, the full definition of a contract clause or a fulfilment line (i.e. participants in a certain role, resources, direction of the resource flow (supply or demand)) cover this functionality.

The REA model specifically focuses on replacing the general ledger and supports the classification in this respect of accounting phenomena as stock objects and flow transactions (McCarthy, 1982, p. 561). The design of the contract data model enables the storage of BPI data over its entire life cycle and thus covers a much broader scope of information. *Ex ante* and *ex post* data are available. In the extended REA model, this scope is also covered (Geerts and McCarthy, 2002).

3.5.2 Compliance of the Contract Data Model with 'Grundrechnung' data storage principles

Schmalenbach's (1948, 1956) contribution concerned the separation of the data area (called 'Grundrechnung') from the application area (called 'Sonderrechnungen') at a time when computers were not yet widely used for business transaction data processing (see also Chapter 2, Section 2.2). In an effort to bring 'Grundrechnung' into practice, Riebel (1994) defined data storage principles. To what extent the design features of the contract data model comply with the data storage principles of 'Grundrechnung' is described next.

- *Data Storage Principle 1. 'No heterogeneous classification or summarizing of data elements'*. This data recording principle prescribes that data classification should be homogeneous and that a detailed definition of data elements should be maintained. This is achieved in the design of the contract data model, as the contract-based data organization approach concentrates on the definition of specific design features for the storage of data phenomena that recur over different types of BPI homogeneously and in detail. Recurring aspects include the definition of resource exchanges by clauses, the definition of contract clause execution by fulfilments, etc. Storing the circumstances and agreed conditions, by which business transactions take place, are examples of data storage with sufficient detail.
- *Data Storage Principle 2. 'No arbitrary division and allocation of accounting data'*. This data recording principle refers to restrictions applicable, for instance, when data are stored through double-entry bookkeeping, because accounting artefacts like debit and credit classification, specific data allocation on general ledger accounts, etc. influence the

generic applicability of stored data. Data storage in the contract data model is based upon the recurring pattern recognized in data characteristics of business transactions themselves (i.e. exchange definition and execution stored using contract clauses and fulfilment lines). This data recording approach is independent of the data reporting requirements of a focused user group (e.g. users of external control data using double-entry bookkeeping data). Because application-independent data on business transactions are stored as contracts and fulfilments, the absence of arbitrary classification or arbitrary data allocation is guaranteed.

- *Data Storage Principle 3. 'Recording an entry at the lowest level possible'*. This data recording principle refers to limitations that occur because detailed data on business transactions are lost when stored at an aggregate level. The approach chosen in the design of the contract data model implies data storage of each individual BPI over its entire life cycle in the data organization approach via contracts and fulfilments. In this respect, the contract data model always stores data on business transactions in the BPI at the lowest possible level.
- *Data Storage Principle 4. 'Characterization with all attributes of interest and importance'*. This data recording principle refers to the importance of completeness in storing relevant data on BPIs in the data model in order to be able to service all the information needs of existing and future users. The contract data model data storage approach complies with this data recording rule as particular importance has been paid to the systematic and user-independent definition of circumstances and conditions, in which BPIs take place through the definition of generic design features on contract clause terms and contract abrogation terms. Attributes of interest and importance are dependent on the specific needs of the information users themselves. Because neutral data storage of BPIs should disregard specific user requirements, storing the resource exchange execution terms (i.e. contract clause terms) and execution abrogation terms is considered to be the only appropriate approach in complying with this data recording principle.

This reconsideration of the 'Grundrechnung' data storage principles demonstrates that the design features of the contract data model are fully compliant with these data storage rules.

3.6 Summary

The 'contract principle' was proposed as a suitable aspect of reality useful to design a data model which can overcome the known disadvantages of existing data models (see Section 2.5 of Chapter 2). This chapter builds on this choice and provides an outline of design features that together facilitate a comprehensive and transparent disclosure of BPI data. The design features chosen correspond to requirements as defined on the availability of BPI data to support the information needs of a multi-disciplinary user community (see Section 2.3 of Chapter 2). To achieve this objective in data recording in the contract data model, a total of nine design features were derived to store a single contract's data over its life cycle, and three design features were described to organize data on relationships between contracts. The use of design features was then illustrated by an example. A comparison was made between data provision through the contract data model using design features as proposed in this chapter, and data organization using the extended REA model on one hand and 'Grundrechnung' principles on the other. This investigation has demonstrated that contract data model can overcome the known disadvantages of the extended REA model and complies with all 'Grundrechnung' data storage principles. The definition of design features can be considered as a logical first step in the design of the new contract-based BPI data organization approach. The next step in the definition of the contract data model is the design using a commonly used data modelling technique. This will be the subject of Chapter 4.