Accounting information for changing business needs

Vandenbossche, P.E.A.

Part II: Functional Architecture

2. The contract as the aspect of reality used in designing the data model

2.1 Introduction

Part two of this research consists of a functional architecture description; the solution to research objective one as outlined in Section 1.4 of Chapter 1:

> ***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. Chapter 2 outlines the functional requirements for business process instance[8] data storage and proposes 'contracts' as the relevant aspect of reality around which the new data model has to be designed. This is in answer to research question one of research objective one as outlined in Section 1.4 of Chapter 1:

> ***Research Question 1:*** How should an alternative data organization method be defined from the recurring pattern recognisable in BPI data?

Chapter 3 will concentrate on outlining the essential data components of contracts. These fundamental data elements will be described as design features to facilitate the design of the contract data model at a later stage.

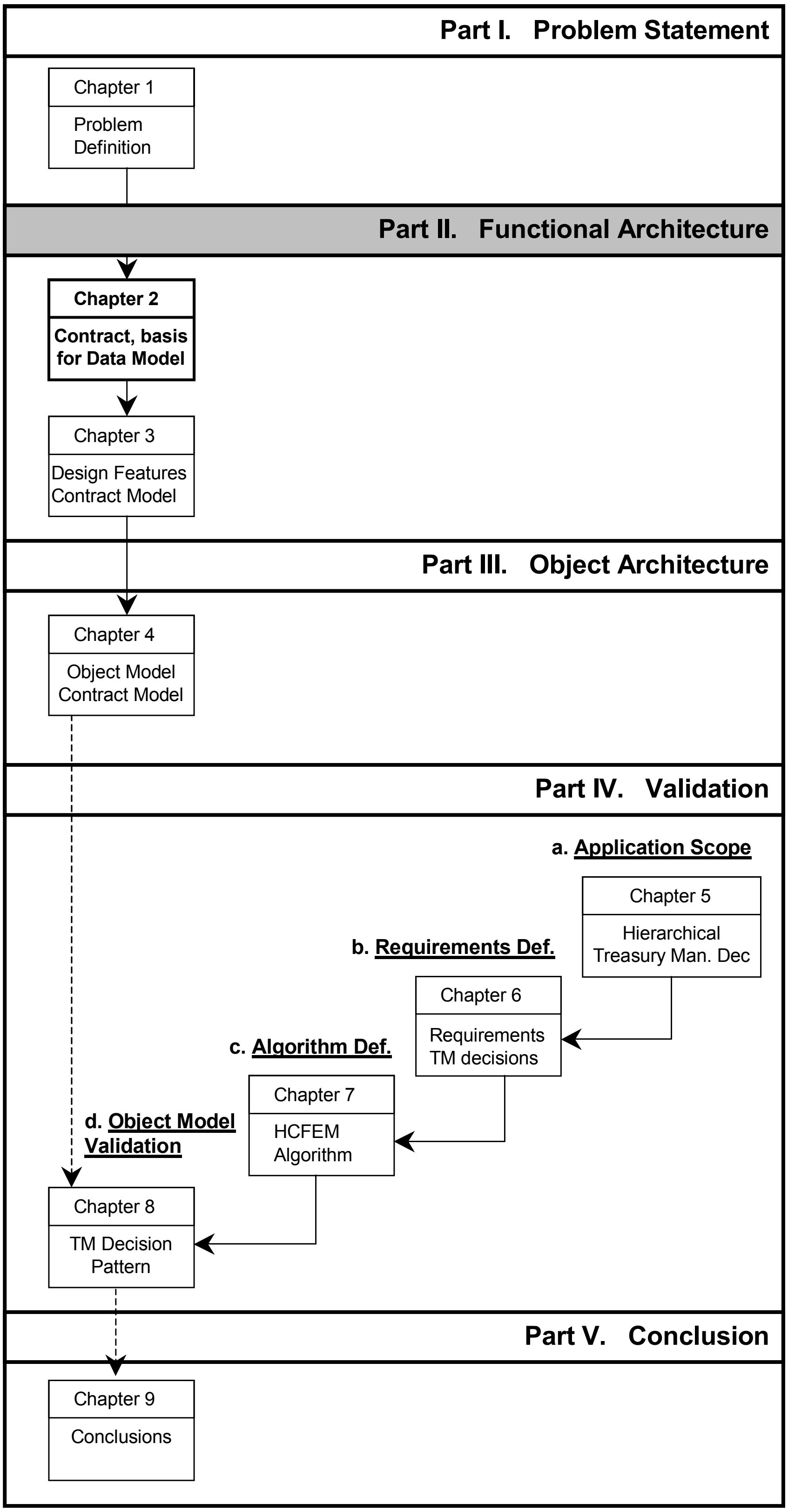
Chapter 2 is structured as follows. First, Section 2.2 presents a literature overview of accounting data model research. Then, Section 2.3 outlines functional requirements for BPI data storage. Following that, Section 2.4 presents a historical overview of how business information needs, currently serviced by double-entry bookkeeping data, have developed over time. Subsequently, Section 2.5 introduces a new approach to BPI data storage, i.e. data model definition based on the 'contract', the relevant aspect of reality in BPI data. Section 2.6 investigates the relationship between 'contracts' and 'organizations' from various viewpoints and the chapter concludes with some final remarks in Section 2.7.

2.2 Literature Review

In recent years, research efforts have concentrated on proposing improved accounting data models that can service a wider audience of information users with changing management information needs. The objectives pursued in these research initiatives are increased sharability of data, increased availability of data, improved evolvability of data and increased data integrity (Everest, 1974).

---

[8] In the remaining part of this chapter, 'business process instance' is abbreviated as 'BPI'.


Part I. Problem Statement
Chapter 1
Problem Definition
Part II. Functional Architecture
Chapter 2
Contract, basis for Data Model
Chapter 3
Design Features Contract Model
Part III. Object Architecture
Chapter 4
Object Model Contract Model
Part IV. Validation
a. Application Scope
Chapter 5
Hierarchical Treasury Man. Dec
b. Requirements Def.
Chapter 6
Requirements TM decisions
c. Algorithm Def.
Chapter 7
HCFEM Algorithm
d. Object Model Validation
Chapter 8
TM Decision Pattern
Part V. Conclusion
Chapter 9
Conclusions

Accounting data models can be discussed at various levels, not all of which are relevant to scientific research. Based on the ANSI/SPARC model[9], database management systems consist of three schemas: the internal schema, the conceptual schema and the external schema. Ullman (1980) notes that these three levels are identical to the three levels of database abstraction: the physical database, the conceptual database and the views. The conceptual database deals with representations as symbols of events in the real world and is considered to be central to the other two schemas. The internal schema sets the required database table definitions and the external schema defines the user interface. Therefore, the most fundamental part of an accounting information system is the conceptual schema and thus the subject addressed in accounting data model research. Researchers of accounting data models do not have to address difficulties related to the internal or external schemas since these are merely questions raised at a system-specific implementation level. Once the discussion of the conceptual schema is established, only the routine work of designing and coding the model is left. The research of accounting information systems is independent of these software development activities.

Another classification of accounting data models is a typology according to database capabilities (Murthy and Wiggins, 1993). Following this approach, three different types of models can be distinguished: implementation models, semantic models and object-oriented models.

- *Implementation Models*. Implementation models are models that are technologically specific. They can only be implemented using a specific DBMS (Data Base Management System). These models are now considered old-fashioned because state-of-the-art DBMS support the definition of semantic models and object-oriented models. With respect to the classification system described above, the implementation model is the internal schema.
- *Semantic Models*. Semantic models are models that focus on the incorporation of the semantics of the application domain into the database schema. According to the classification system described above, the semantic models is the conceptual schema.
- *Object-Oriented Models*. Object-oriented models are models that incorporate both structural and behavioural aspects in the model. According to the classification system described above, object-oriented models also come under the category of conceptual schemas, requiring an internal schema for actual implementation. Object-oriented modelling has many advantages in the areas of the development, operation and maintenance of information systems.

In the field of (accounting) information system design, object-oriented models are considered to offer the most interesting area of research as they are readily incorporated into *standard* business information systems that can be deployed in real-life customer implementations.

The American School and the German School predominantly conducted research in the area of accounting data models, proposing research results that are still useful today, even though the two schools carried out their research projects independently. A summary of the outcomes of the research initiatives of both schools is outlined below.

---

[9] ANSI/SPARC (American National Standards Institute / Standards Planning and Requirements Committee) is a body that determines and evaluates standards associated with information technology. A three-layered database schema is advocated by the Data Base System Study Group, which is a subordinate body of ANSI/SPARC. The central schema of the three is called the conceptual schema and faithfully expresses events in the real world. The other two schemas, the external schema and the internal schema, lie beside the conceptual schema.

Sorter (1969) dominated the research ideology of the American School through his 'events theory' of accounting. This theory stresses the importance of events that change the balance sheet of an organization according to the financial reporting methodology, and was proposed as an alternative to the 'value theory' of accounting, the more traditional approach. An overview of the research results of the American School is presented by Murthy and Wiggins (1993), Sakagami (1995), Weber and Weissenberger (1997), Dunn and McCarthy (1997) and Verdaasdonk (1998). The most significant semantic model of the American School is McCarthy's (1979, 1982) REA model. The REA model is a framework describing the real world in terms of resources, events, agents and the relationships between them, and is able to fulfil all kinds of (financial) reporting functionalities. The REA model is visualized in Figure 2-1.

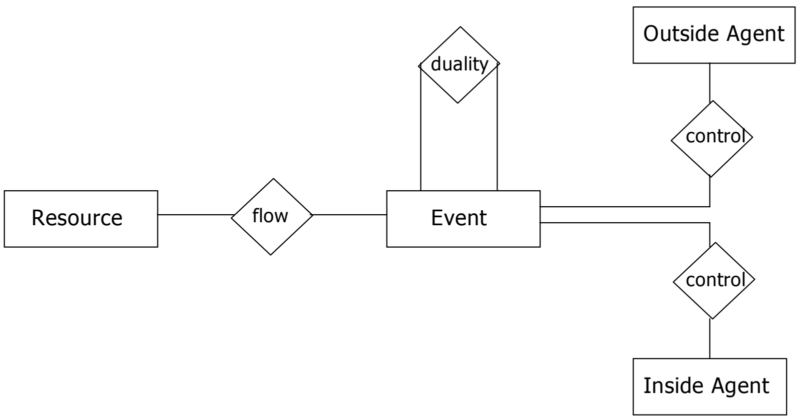


Figure 2-1. Components of the REA model (source: McCarthy, 1982)

The components of the REA model are defined as follows: *Resources* are defined as '*objects that are scarce and have utility and are under the control of an enterprise*' (Ijiri, 1975, pp. 51-52). *Events* are defined as '*a class of phenomena which reflect changes in scarce means resulting from production, exchange, consumption and distribution*' (Yu, 1976, p. 256). Events trigger a resources state-change. This is expressed by the *flow* relationship, defined between resources and events. *Agents* are defined as '*persons and agencies who participate in the economic events of the enterprise*' (McCarthy, 1982, p. 563). Each event is always characterized by two agents, one internal agent (also called a 'unit') and one external agent. According to the REA philosophy, an event can always be considered from two viewpoints - a 'give' and a 'take' event (Geerts and McCarthy, 1997). The 'give' relationship of one agent is the 'take' relationship of the other agent involved in an event. Since every 'give' event is ultimately always related to at least one 'take' event, the relationship between them is called a *duality*.

The REA model has a number of drawbacks in practice. McCarthy (1982) explains that the REA model is meant to replace the general ledger. Verdaasdonk (1998) explains that as a consequence of this aim only historical reporting functionality was catered for. The REA model does not feature the accommodation of future-oriented data. Sakagami (1995) remarks that the REA model is not suitable for real-life customer implementation because the model itself is an entity-relationship model, whose number of REA components (i.e. resources, events and agents) can grow exponentially. This drawback was later recognized and an object-oriented version of the REA model was created (see Dunn and McCarthy, 1997; Geerts and McCarthy, 1997; and McCarthy, 1995). Geerts (1997) claims that the REA model lacks 'reusability' and 'extendibility'. The REA model has been further developed in the extended REA model and a REA domain ontology (Geerts and McCarthy, 2000, 2002). The REA domain ontology overcomes much of the original REA model's scope limitations but has yet to be designed further into object-oriented analysis models that can be implemented and deployed as the data models of real-life ERP system implementations.

The second category of important research initiatives on improved data models is from the German School. The most important contribution of researchers of the German School is the requirement for a total separation of the data area (called 'Grundrechnung') from the application area (called 'Sonderrechnungen') (Schmalenbach, 1948, 1956). In comparison to McCarthy's REA model, 'Grundrechnung' does not address aspects from the financial accounting domain since it is a data area only. Therefore, the 'Grundrechnung' is not intended to replace the general ledger in the way McCarthy's REA model does. Riebel (1994) developed the following recording principles on storing accounting data in 'Grundrechnung' based on Schmalenbach (1956):

- *Data recording principle 1: No heterogeneous classification or summarizing of elements needed separately for applications*
- *Data recording principle 2: No arbitrary division and allocation of accounting data*
- *Data recording principle 3: Entries are to be recorded at the lowest level possible in the hierarchy, without introducing arbitrary allocations*
- *Data recording principle 4: Characterization with all attributes of interest and importance.*

Sinzig (1983, 1994) elaborated upon the ideas of these data recording principles and has proposed a semantic accounting data model using the 'Grundrechnung' principles from Schmalenbach and Riebel. He developed a relational data model to accommodate data to service an application on German-based cost accounting (i.e. 'Einzelkosten- und Deckungsbeitragsrechnung', an example of a 'Sonderrechnung', see above). Sinzig named his data model (the 'Grundrechnung') a '*purpose-neutral database*', because it was oriented at accommodating data to service a broad range of applications. In its initial version, his data model was designed for one organization only (Riebel and Sinzig, 1981; Sinzig, 1983). This approach could bring the model's generality into question. Later, Riebel et al. (1992) and Sinzig (1994) concluded in publication that this model had achieved an acceptable degree of 'purpose neutrality' so that it could be used in the R2 ERP system of ERP vendor SAP AG. An overview of the research initiatives of the German School is presented by Verdaasdonk (1998). Research by Weber and Weissenberger (1997) indicated that Riebel's approach has not resulted in applications in practice.

Section 1.3.2 of Chapter 1 investigated from the literature and then illustrated in a sample ERP system whether current ERP systems have adopted solutions as proposed in scientific literature to resolve the need for a shared data environment[10]. The following findings were formulated:

- No features of the (extended) REA model have been adopted in the data models of current ERP systems
- Data models of current ERP systems do not comply with 'Grundrechnung' data recording principles

[10] The ERP system 'SSA Baan ERP' of SSA GT. was chosen as representative sample ERP system in this research.

- ERP vendors have not designed and implemented an own application-independent data storage method in the architecture of their ERP data models

Additional information needs could result in the need for modification of the data model.
A suitable, generic accounting data model to overcome these drawbacks has yet to be defined. This new data model should enable data accommodation to service the information needs of all the various types of ERP system users from one shared data environment, and should be extendible so that it can be used satisfactorily in large-scale implementations. Authors like McCarthy (1982) and Verdaasdonk (1998) prescribe that objective data should be provided and that the design pattern of such a data model should be close to aspects of reality so that no restrictions are adopted in the data model. The requirements of such a generic accounting data model acting as a sole data source of ERP systems will be systematically defined in the next section.

2.3 Requirements of Business Process Instance Data Recording

In this section, the requirements for an improved data recording approach to the accounting information of BPIs are defined. These requirements are defined on the basis of literature analysis, discussion with ERP architects and investigation of accounting data structures in current ERP systems (see Section 1.3.2.2 of Chapter 1). The overall objective of accounting data models is '*to exclude value as far as possible from stored data and to make processing of information possible in various ways*' (Sakagami, 1995). The requirements that are defined in this section contribute to the attainment of this goal.

Requirement 1: Objectivity of business process instance data
Sakagami's remark that 'value' has to be excluded as far as possible from data defined in the accounting data model relates to the requirement of only storing 'value free' or 'objective' data (Sakagami, 1995). The 'value free' data objective can be explained using the 'events accounting' approach as presented by Sorter (1969). He explains that '*the purpose of accounting is to provide information about relevant economic events that might be useful in a variety of possible decision models…Instead of producing input values for unknown and perhaps unknowable decision models directly, accounting provides information about relevant economic events that allow individual users to generate their own input values for their own individual decision models*' (Sorter, 1969, p. 13). In accordance with this view, 'value free' data are defined here as data that can be used in a variety of decision models, even in decision models that are not yet known at the moment the BPI data are stored. Authors like McCarthy (1979), Verdaasdonk (1998) and Everest and Weber (1977) refer to the fact that no double-entry bookkeeping artefacts such as debit/credit, general ledger account classification, etc., should be adopted in the data model.

The scope and nature of 'objective' data is illustrated in an example, 'purchase contract to buy a new machine'.

Purchase price: USD 1,000,000. Purchase contract issue date, 10 October 1998; start of machine use, 1 December 1998; end of machine use, 1 December 2003. At this date, the machine may be returned. According to the purchasing manager, the machine value is USD 250,000 at disposal date. However, the official catalogue representing the industry standard consensus indicates a machine salvage value of USD 100,000 at disposal date.

Situation 2-1. Illustration: different types of 'data'

The following types of data are considered 'objective':

- *Pure objective data*. These data are objective as they are not or cannot be influenced by people or situations. For example, the purchase contract issue date (because it is a copy of the system's date)
- *Inter-subjective data*. This type of data is typical to contracts. Its characteristic is that all contract participants generally agree on the content. As this type is not one-sided, involving several participants generally agreeing on the content, it becomes formalized, qualifying as 'objective data'. Examples are the purchase price, the start and end date of machine use, etc.
- *Data formulated by authorized people or organizations*. If people or organizations are authorized to formulate an opinion (data), this type of data becomes 'objective data' as a consequence. An example of objective data formulated by authorized persons is the USD 250,000 salvage value formulated by the purchase manager. Normally, this type of data should be considered as subjective data as it is a one-sided opinion based upon unknown criteria and objectives. However, if this person's vision is accepted (e.g. by the company board), this person is 'authorized' to formulate a one-sided opinion and the data has to be considered as 'objective'. To contrast this with subjective data, if the machine operator defined the salvage value as USD 50,000, this data would be considered one-sided and subjective because the machine operator is unlikely to be authorized to formulate machine-salvage values. His opinion, therefore, is not and should not be stored in the data model. An example of objective data formulated by authorized organizations is the USD 100,000 machine salvage value as described in the official catalogue on industry standards (issued by an authorized organization).

The justification for only recording *objective* data is twofold. First, only objective data can be used to support *multiple* information requests. The dominance of one application over the others can be avoided only by including accounting data that can be measured objectively. All other data needed has to be derived from these data. This is why the semantics used in financial information system design should exclude specific application artefacts. Second, only objective data allow the servicing of *changing* information needs, i.e. if the information requests change, data can be reused to service the revised information requirement since they are neutral and not tailored to solve only the initial information requirement. In this dissertation, Sorter's viewpoint is clearly applied in the role of accounting as a data storage method: '*accounting provides information about relevant economic events that allow individual users to generate their own input values for their own individual decision models*' (Sorter, 1969, p. 13). This can only be achieved when available BPI data are objective and reusable. Many authors have highlighted the importance of data neutrality. Schmalenbach (1948, 1956) maintained a clear separation between 'Grundrechnung' (the data domain) and 'Sonderrechnungen' (the application domain). Sinzig and Riebel described a data storage approach applicable for multiple purposes they called '*Purpose-Neutral Database*' (Riebel and Sinzig, 1981; Sinzig, 1983, 1994).

This requirement overcomes the disadvantages of BPI data storage in current ERP systems as described in Section 1.3.2.2 of Chapter 1, Finding 3 ('Every finished activity of the BPI leads to BPI state, domain-specific data storage. This BPI state data is stored in isolation and not as a sub-categorization of a BPI').

Requirement 2: Data on a *single* business process instance to be defined in coherence

Besides consistently storing data on individual finished activities, information on *integration* between data of finished activities of the same BPI should also be maintained in the data model. Without this information, it is often impossible to group all data on one BPI together. In accordance with Requirement 1, data on the integration of finished activities of a BPI should be described in a 'value free' and 'objective' way so that multiple information requests can utilise it. No single set of information needs should be dominant in the definition of this integration.

This requirement overcomes the disadvantages of BPI data storage in current ERP systems as described in Section 1.3.2.2 of Chapter 1, Finding 2 ('There is no equivalent of the BPI at data level. Data on a single activity are persistently stored but the activity *network* (i.e. the BPI) itself is lost at data level') and Finding 3 ('Every finished activity of the BPI leads to BPI state, domain-specific data storage. This BPI state data recording is stored in isolation and not as a sub-categorization of a BPI').

Requirement 3: Data between *different* business process instances to be defined in coherence

Data on different BPIs should be defined in such a way as to enable them to service multiple information needs using data from different business BPIs, even when these information needs are unknown at the moment of BPI data storage. In keeping with Requirement 1, 'objective' and 'value free' integration between data of different BPIs should be accommodated to guarantee the capability of servicing multiple information needs. No single set of information requests may be dominant in this integration definition.

This requirement overcomes the disadvantage of BPI data storage in current ERP systems as described in Section 1.3.2.2 of Chapter 1, Finding 4 ('If data on a single BPI are not stored in a structured manner, then a structured data organization approach across BPIs cannot be expected').

Requirement 4: Data defined on business process instances are of an *ex post* and *ex ante* type

Existing accounting data models predominantly focus only on the availability of *ex post* data. The nature of double-entry bookkeeping is data storage of past occurrences. The objective of the REA model is to replace the general ledger, thus only recording *ex post* data on economic occurrences (McCarthy, 1979, 1982). Johnson and Kaplan (1987) and Riebel (1994) refer to restrictions when *ex ante* information is not stored. There is a balanced need to have both *ex post* and *ex ante* data available. There should not be a boundary between *ex post* and *ex ante* data. Instead, the concept could be visualised as a smooth transition from *ex ante* data to *ex post* data with *ex ante* data continuously rolling forward.

Requirement 5: Data are required on *internal* as well as *external* business process instances

Existing information systems often only support the formal storage of event data of the organization in relation to the outside world. Data pertaining to internal agreements are often not formalized or recorded as such. Managing internal BPIs can only occur when data on these BPIs are stored. For larger organizations in particular, it is becoming increasingly important that internal participants be made responsible and accountable for what they promise in the execution of an external obligation. It would be convenient for the data recording of internal BPIs to follow the same storage methods as those used for external BPIs.

The five requirements as described above should enable the definition of a data model that allows the accommodation of data to service changing information needs without making modifications to the data model. In Chapters 3 and 4, a data model will be designed which complies with these requirements. This data model should later be validated in real-life circumstances. On the one hand, this validation should provide insights into the question of whether it can handle large volumes of data specific to real-life customer implementations (this question will not be investigated in this research project). On the other hand, this validation should help understand to what extent the data stored in the data model could really service new and/or changing information needs. This part of the validation is the subject of Chapters 5 to 8.

The remaining part of this chapter is organized as follows. The evolution of business information needs over time, currently supported by double-entry bookkeeping data, is investigated in Section 2.4. On the one hand, this investigation illustrates the suitability of double-entry bookkeeping data to service today's business information needs. On the other hand, it will be used to derive the relevant aspect of reality, i.e. the recurring pattern, that can be found in data of different BPI types that trigger journal entries. This recurring pattern will then be used in Section 2.5 and subsequently to outline an alternative data storage method.

2.4 Historical overview of the evolution of business information needs currently supported by double-entry bookkeeping data

For years now, authors such as Everest and Weber (1977), McCarthy (1979) and Verdaasdonk (1998) have been insisting that accounting 'artefacts' associated with journals and ledgers like debit/credit separation, classification schemes, specific allocations, etc., need to be excluded from data models because these artefacts are only mechanisms introduced to facilitate the manual storage and transmission of data. McCarthy (1982) explains that, as such, they are not essential aspects of an accounting information system.

McCarthy (1982) states that an improved method to define data models capable of supporting existing and new information needs should contain aspects of reality that occur within the defined scope in which the information needs are to be supported. One aspect relevant for adoption in the data model design is the recurring pattern in the data to be stored. There are various possible ways to extract the recurring pattern from BPI data. For instance, case studies could be researched for representative customer implementation to understand which different types of BPIs occur, and the recurring pattern in the data could then be investigated. Another approach would be to investigate which types of business processes occur in ERP systems and derive the recurring pattern from the instance data. A different approach was followed in this research. Since the first two proposed methods require very elaborate research activity, data of BPIs currently served by double-entry bookkeeping data were investigated. According to the literature (McCarthy, 1979), double-entry bookkeeping is still the most frequently used data model to date, having survived for centuries. This should explain why this approach is likely to lead to useful results.

In 1494, in Italy, a special data storage method called 'the double-entry bookkeeping system' was invented by Paccioli. This formal data storage method has survived over the centuries and became the standard source for financial information used all over the world. Investigating the economic and social context in which this system was founded is therefore of particular interest. The double-entry bookkeeping system was initially used in commerce-driven Renaissance Italy, where merchants felt the need to store transactions with their trade relations in an orderly manner in order to remember their mutual obligations (Yamey, 2000). At that time, business activities only consisted of purchasing and selling goods between trade participants. There was no value-creation or value-addition process. In addition to a storage function, the double-entry bookkeeping system also facilitated ordering and manipulating data. Its objective was to be a standard for BPI data storage that could facilitate profit calculation in a uniform and straightforward way for the local government. To attain the objective of profit determination, data on trade transactions were only stored selectively in 'categories', which were later called 'general ledger accounts' (Geijsbeek based on Paccioli, 1974). 'Selective storage' means that only the relevant financial data were recorded. A transaction line was the description of an exchange that took place with external business relations. The financial data on these exchanges were recorded in operational supply and demand contracts. Internal processes like employee data storage, internal transportation, value-adding activities etc. were not recorded because they were not considered relevant to the business. As a consequence, the data recorded at that point was not a full description of the business event, it was the storage of *selective* financial data that could be used to keep track of an external obligation and to support profit determination. The double-entry bookkeeping information system only focused on the storage of business events, of the organization in interaction with its environment. Its objective was to accommodate data for **external control purposes** only. This information was provided satisfactorily for several centuries. However, organizations became more complex over time. With the first industrial revolution, there was a shift from trade-only organizations towards production organizations with the appearance of a corresponding awareness of value-creation and value-addition. Over time, more knowledge became available on internal processes, such as production processes and financial processes. Information to service **internal planning and control** requests became required (Kaplan, 1984). The need to accommodate data to support these processes was obvious. The variety of formal agreements that were used in these processes became wider and employment contracts, machine purchase contracts, maintenance contracts etc. also became important. The increase in financial knowledge, especially in the area of financial management, has led to the introduction of a series of new financial contracts, e.g. different types of loan contracts (mortgage loan contracts, bonds, etc.), hedging instruments (option contracts, forward rate agreements, etc.) and complex payment instruments (trade notes, letters of credit, etc.).

Organizations were already familiar with Paccioli's formalized data storage method. It has to be re-emphasized that the double-entry data storage method was created to accommodate financial data for *external control* purposes in *simple trade organizations* only. The original objective was to have a systematic approach to 'remember' obligations with external parties. However, this system started to be used to store data on complex financial processes and internal processes as well, something that double-entry bookkeeping, unable to accommodate all the data required for information requests in each of these application areas, is not equipped to do. Specific data such as data on risk elements that can be involved in financial processes or data on throughput, lead time, etc., which characterize production processes, cannot be provided. With the introduction of new, complex financial contracts (e.g. option contracts), obligations with external participants also became more complex and dependent on particular situations. The obligation is described in the contract itself and data describing the various aspects of the obligation goes well beyond the level of detail currently available when only the financial outcome of BPIs are stored in general ledger accounts. Internal processes detailing the activities performed by internal participants focus on value-creation and value-addition. When these processes are serviced with double-entry bookkeeping data, only the *actual* difference in value over the value chain can be tracked. Information on activities only becomes available once they are finished and only the end result is stored. No information is available on the question of whether the activities are executed as outlined in the conditions or not. However, this information is required to support basic information requests like searching for the cause of differences using variance analysis. As opposed to the data requirements of external information requests where the financial result is often sufficient, using double-entry bookkeeping data to capture data on internal business processes results in recording financial data alone. Data on rights and obligations between the internal (contract) participants is lost. Data requirements to support internal processes differ significantly from those needed to support external business processes – more data are needed to support internal processes.

McCarthy (1980, p. 628) formulates the following limitations of double-entry bookkeeping (see also Belkaoui, 1992, p. 110, Hollander et al., 1996, pp. 49-54 and Verdaasdonk, 1998):

- *Its dimensions are limited*. Most accounting measurements are expressed in monetary terms, a practice that precludes maintenance and the use of productivity, performance, reliability and other multidimensional data
- *Its classification schemas are not always appropriate*. The chart of accounts for a particular enterprise represents all the categories into which information concerning

economic affairs may be placed. This will often lead to data being left out or classified in a manner that conceals its nature from non-accountants
- *Its aggregation level for information is too high*. Accounting data is used by a broad variety of decision-makers, each needing different degrees of quality, aggregation and focus depending on personalities, decision styles and conceptual structures. Therefore, information concerning economic events and objects should be kept at the most elementary form possible to be aggregated by specific users
- *Its degree of integration with other functional areas of an enterprise is too restricted*. Information concerning the same set of phenomena will often be maintained separately by accountants and non-accountants, thus leading to inconsistency as well as information gaps and overlaps.

The disadvantages of double-entry bookkeeping data as described by McCarthy are directly related to the fact that these data are used today to service information needs that go well beyond the scope and characteristics of those that were initially the focus when this data recording method was invented. The evolution in the functional spectrum of information needs requires a more suitable and complete data recording method. This data storage method should be capable of meeting the current and future information needs of a range of different user types.

2.5 Why contracts are a useful aspect of reality to design the data model

In this dissertation, we want to propose an improved data model for BPI data that can hold data to service existing, new and changing information needs. In Chapter 1, Section 1.2, two approaches that could be followed to solve the question of how to provide suitable data in a dynamic environment with changing information needs were discussed. First, existing data models could simply be extended, based on the additional data requirements of new information needs (discussed in Section 1.2.1). Second, the design of a data model could be focused on, holding all required data to service all existing and new information needs within a given scope of applications (described in Section 1.2.2). In Section 1.2.3, we explained why this approach will be followed in this research.

When choosing to design an improved data model holding data to service all existing and new information needs in a particular domain, authors like McCarthy (1979), Everest and Weber (1977) and Verdaasdonk (1998) prescribe that the data model should be designed according to the relevant aspects of reality[11]. It should not hold any application artefacts such as the 'general ledger', 'debit/credit' etc., because these would restrict the ability to provide data to support other types of applications. This prerequisite is the main difference between data models built to support a limited and frozen set of information needs, determined at given moment, and a data model designed to support existing and new information needs within a chosen scope.

The above recommendation has been observed in the design of the data model presented in this research. Two questions need to be answered before designing the data model. Question 1: What is the domain for which the data model needs to hold information? Question 2: What are the relevant aspects of reality occurring in the chosen domain, so that the data model can be designed on their basis?

---

[11] With 'aspect of reality', we mean phenomena which occur in the real world, as opposed to application artefacts or abstractions made for IT implementation reasons. With 'relevant', we mean those aspects of reality which are recurring, and which are useful in the context of data model design.

The answer to question 1 can be derived from the objectives of this research. Section 1.4 explained the desire to provide an improved data model that could serve as a data source for an ERP system. The data model has therefore been designed to hold data to service accounting information needs in the scope of an ERP system.

Question 2 can now be redefined as follows: ‘What are the relevant aspects of reality that occur in the application scope of an ERP system?’ Of all the information requirements that are typically supported by an ERP system, the focus of this research is on those ERP information requirements that use accounting information in the broad sense of the word. In current ERP systems, this type of information requirement is typically accommodated by double-entry bookkeeping data (see e.g. McCarthy, 1979; illustrated for a sample ERP system in Section 1.3.2.2 of Chapter 1). To define the relevant aspect of reality, Table 2-1 outlines a summary of the BPI types for which data are currently stored in double-entry bookkeeping as described in Section 2.4.

Table 2-1. Business process instance types for which data are currently stored in double-entry bookkeeping

	Scope	Subject	Stored data components
1.	Between Organization and Business Partners	Sales/procurement transactions, exchange of operational resources (trading finished products)	Financial summary, ownership indication, no details on quantity and price, etc.
2.	Between Internal Participants	Value-creation/Value-addition processes. Exchange of finished activity against value	Information on ‘value’ in financial terms, no details on lead-time, throughput, etc.
3.	Between Organization and Financial Partners	Exchange of various financial resources against each other[12]	Information on outcome in financial terms (value of exchange, no details on risk, currency rates, etc.

Table 2-1 shows that the following aspects of reality can be found in the BPI types mentioned. BPI types can be distinguished between the organization and the external environment (business partners or financial partners) on the one hand, and BPI types defined internally in the organization on the other. With BPI types between the organization and the external environment, the underlying trigger of the event can be perceived as an engagement or agreement on a *resource exchange*, together with its *execution*. This *resource exchange* relates to operational resources (goods or services) where the opposing party is a business partner (e.g. a customer or a supplier). It relates to financial resources (e.g. a foreign currency amount, a loan, etc.) when the opposing party is a financial partner (e.g. a bank or an insurance company). The *execution* relates to the delivery, receipt, invoicing, payment, collection, etc. All participants are encompassed by the organization for internal BPI types. The aspect of reality is the agreement or engagement on an *exchange of value-adding activities* (e.g. manufacturing or administrative in nature) and its *execution*. The fundamental differences between the support of these transactions with double-entry bookkeeping compared to the approach proposed in this thesis are twofold. First, double-entry bookkeeping only stores partial information (i.e. instead of storing the complete BPI data from agreement to execution, double-entry bookkeeping only stores the financial result in a general ledger account, while data on terms and conditions are lost). Second, data covering the *entire*

[12] Two examples are given of this type of *financial resource exchange*. Example 1: exchange of money against a time-phased payback facility (i.e. a loan). Only the total financial value is stored via double-entry bookkeeping, no information is available on aspects such as total payback time, frequency of installments, installment amount, interest rates used, etc. Example 2: exchange of a hedged risk against a certain amount to be paid. Double-entry bookkeeping only stores the total amount paid for the hedging operation. No information is stored on the nature of the risk, duration for which the risk is hedged, etc.

business BPI (i.e. the context and integration of data on the agreement on resource exchange and its execution in several steps afterwards) are lost.

The recurring aspect of reality in the chosen scope of applications (accounting information for ERP information requests) is found to be the *exchange of resources or value-adding activities and their execution*. The trigger of this resource exchange is the *engagement* or the *agreement* on the terms and conditions of this exchange. This recurring pattern is accordingly 'labelled' as 'the contract' in this dissertation and its execution – according to the contract's terms and conditions – of the resource exchange, 'the fulfilment'. It should be noted that these definitions have been chosen in the context of the research goals of this dissertation. A contract is defined in this dissertation as '*a collection of contract clauses, where a contract clause is the expression of an exchange of resources or value-adding activities executed by one or more observances of the contract clause terms*'. Although the contract is defined in several professions, such as law and accounting, the contract has not been used in those contexts since the focus here is on designing a data model.

The choice outlined above requires some clarification. To begin with, in legal science one of the key features of the contract is its legal enforceability. The very fact that the contract is recognized here as the recurring pattern, also capable of expressing internal agreements on value-adding activity exchanges, is an illustration of the fact that the use of the contract in a legal sense is not envisaged, since legal enforceability is impossible for internal contracts. However, the definition of contracts in legal science does play an important role when defining the essential data elements to be captured by the data model to record full information on the contract. Examples include the legal definition of aspects such as contract terms, abrogation terms, contracting parties, etc. These topics have been discussed thoroughly in legal science and are therefore used in Chapter 3 when outlining the design features pertaining to these data components.

The contribution of accounting science with respect to the definition and use of contracts can be discussed similarly. In accounting, contracts are discussed in the context of agency theory and property rights. Agency theory discusses how best to organize relationships in which one party (the principal) determines the work that another party (the agent) undertakes (Eisenhardt, 1985). This theory argues that under conditions of uncertainty and incomplete information, two agency problems arise, namely adverse selection and moral hazard. Adverse selection is the situation where the principal cannot ascertain whether the agent accurately represents the ability to perform the task he or she is being paid for. Jensen and Mackling (1976) explain that fixed wage contracts are not optimal in this situation. Moral hazard is where the principal cannot be sure if the agent has exerted maximum effort (Eisenhardt, 1989). In this research, contracts are used in the context of the aspects of reality relevant to designing the data model. The agency theory focuses on another question, namely, how are contracts used internally between a principal and an agent? The body of knowledge of accounting on contracts does not therefore immediately provide anything in the context of this research.

From a definition viewpoint, a distinction should be drawn between the application of contracts as an aspect of reality in designing the contract data model on the one hand, and the use of contracts internally and externally to express mutual commitments on the other. When contracts are considered as aspects of reality in designing the contract data model, reference is made to the fact that contracts were found to be the recurring pattern of resource or value-adding activity exchanges within the chosen application scope, around which the data model has to be designed. The emphasis of this thesis is on this aspect (see Research Objective 1, Section 1.4 of Chapter 1). Aside from this, the question of how contracts are used within the organization internally and between organizations to express commitments could be considered, as an activity of using the data of these recorded contracts. This approach is useful as algorithms or applications can be described as series of definitions of relationships between different types of contracts where the focus is on describing the data flows ('the type of data used in an application'). This approach is termed 'Contract-based Accounting' here. In this thesis, an example of the 'Contract-based Accounting' approach has been provided in the context of validation of the completeness of the data stored in the contract data model. The algorithm used to calculate the treasury management decision alternatives in a generic way for all treasury management decisions (see Chapter 7) is defined as the definition of relationships between different types of contracts, and is therefore an example of 'Contract-based Accounting'.

When defining a data model on the basis of a limited set of requirements frozen at a given moment, and when extending the data model when additional information requirements need to be supported, see Section 1.2.1 of Chapter 1, the scope for which the data model holds data can typically be described in terms of an application (e.g. credit limit check), a functional domain (e.g. financial accounting in Belgium) or a sub-domain of a functional domain (e.g. accounts receivable requirements according to Belgian accounting regulations). The scope of information needs that can be supported on the basis of the data stored in a data model, which is not designed on the basis of specific functional requirements but on the basis of aspects of reality (as discussed in Section 1.2.2 of Chapter 1), cannot be described similarly. Since the objective in this approach is to support existing and new information needs within a certain domain, and the aspects of reality upon which the data model is designed are chosen from within this domain, the scope within which the data model's data can be perceived as accommodating sufficient information should not be articulated in terms of an 'application' or a 'functional (sub) domain' because it is quite likely that the scope for which data can be provided is much larger. It is, for instance, possible that a new application domain will come into existence in the future though sufficient information will already be available in the data model because there are no new requirements for data availability. It is therefore proposed that the scope within which the data model holds data be described on the basis of the characteristics of the data stored. Applied to the contract data model, the scope of information requirement for which the contract data model can provide data is described as: 'all the information requirements that make use of data that could be derived from data on the agreed exchange of resources or value-adding activities, and the execution of this agreement'. This does not necessarily coincide with a functional domain or an application, as illustrated below.

In the contract data model, it is the objective in principle to provide data only through contracts. It cannot be denied, however, that some information requirements, such as focusing on the requirements of users in the scope of external control applications, prescribe that data should be presented in general ledger account format (e.g. to generate the annual report)[13].Double-entry bookkeeping data storage has been explicitly abandoned in the conceptual solution proposed to exclude known restrictions (e.g. see McCarthy, 1980, p. 628). However, at implementation level, it is possible to solve the problem by selecting from the contract data and presenting and/or storing these data in general ledger accounts in a separate, derived data model to hold accounting data in the format as prescribed by legal, external control-oriented information requests. This could even be a preferred implementation approach for system performance reasons. This approach is acceptable so long as full contract

---

[13] It is recognized that some business events, such as valuation (depreciation, revaluation of fixed assets, currencies, etc.), and year-end transactions, e.g. allocation of profit, provisions etc., are only defined via general ledger accounts and not via contracts. This is the application scope of users in the application domain of Financial Accounting. In the proposed conceptual solution of the contract data model, the scope of stored data is limited to storing data on the 'operational' business events relating to resource exchanges. This would not include the Financial Accounting events, which are unrelated to operational business events. In actual implementations, one should think of a combined solution of contracts and general ledger accounts to cover all types of business events.

data continue to remain available in the contract data model, the shared data environment to be used by all users.

The contract data model can generate objective data (see Requirement 1, Section 2.3) and is not designed to support one specific set of requirements (so-called purpose-neutrality). However, the actual scope of characteristics such as ‘purpose-neutrality’ and ‘objective data’ is not absolute and has to be considered from the perspective of the goals of accounting data model research. The objective is not to have a perennial data model, originally designed to hold data to support business information requirements, but one that can be deployed equally well in very different application domains, such as aircraft or machine control. Accounting data model research concerns itself with increasing the reusability and sharability of data for future information needs that remain within the chosen domain scope. Only within this context can purpose-neutrality be of issue.

2.6 How contracts relate to organizations

Twenty years ago, the boundaries between organizations tended to remain relatively stable for extended periods. However, besides frequently redesigning their organizational structures, today’s organizations increasingly opt for mergers and acquisitions as straightforward strategic choices in achieving overall objectives and to increase efficiency. The usability of stored data in this dynamic environment of on-going organizational change remains important, for example in supporting information requests like long-term trend analysis. Current information systems cannot cope effectively with the new tendency for shifting organizational boundaries. This again relates to the design approach used for the data model chosen (i.e. designed on the basis of frozen requirements at given moment). Since flexibility in data provision is fundamentally important in a dynamic environment, it is the aim of this section to investigate how the chosen aspect of reality in data model design (i.e. ‘the contract’) relates to the organization from various perspectives. This investigation has been carried out to determine whether ‘contracts’ are not only a useful aspect of reality at the level of BPI data but also when taking the aspect of ‘organization’ into account.

2.6.1 Contracts are the formal information base of organizations

Contracts start playing a role in organizations at the beginning of their life cycle, when they are founded. Organizations are founded following procedures detailed by law. Authorized persons always formally store the agreements made between the founders in a foundation contract. From that moment on, the organization is able to act in its own name as a legal entity. After its foundation, every transaction involving the organization with external participants creates rights and obligations for all participants. These rights and obligations are formalized in contracts defined between business partners. Under the proposed contract-based data storage method, these contracts are the starting point from which each information process is derived. As ‘contracts’ are the formal store of business agreements where execution is legally enforceable, an organization’s portfolio of legally enforceable contracts can be regarded as manifesting its legal ownership rights. The fundamental reason why the concept of ‘contracts’ is greatly intertwined with organizations is that a formal administrative organization can be defined on the basis of formal contracts. When the boundaries between legal entities change as a consequence of mergers or acquisitions, the organizational changes themselves are also the subject of explicit, formal contract administration. The execution of long-term contracts (e.g. multiple-year construction contracts) can still be ongoing in a situation where the original legal entity that created the contract was subject to a merger or acquisition resulting in a legal transition of the contractual ownership. Traditional data storage approaches like double-entry bookkeeping cannot cope effectively with this kind of change. However, once full contract administration is in place, this kind of construction contract can be brought under the auspices of merger or acquisition contracts, and through the details of this relationship, it is automatically clear at all times which organization is the legal owner of the construction contract. These arguments further justify choosing 'contracts' as the central, transparent, data organization method for recording BPI data.

2.6.2 Contracts in relation to participants inside and outside organizations

Contracts can also be described from the perspective of the different possible types of participants and their relationships with organizations. Organizations operate in an economic environment. Potential contract participants operate in the same environment. They can be natural persons or legal entities (e.g. organizations). The contracts that have to be stored in the shared data model are all possible combinations of agreements between organizations and different types of persons (natural persons or legal entities) in an environment within which organizations interact in an economic sense. The different possibilities are visualized in Figure 2-2.

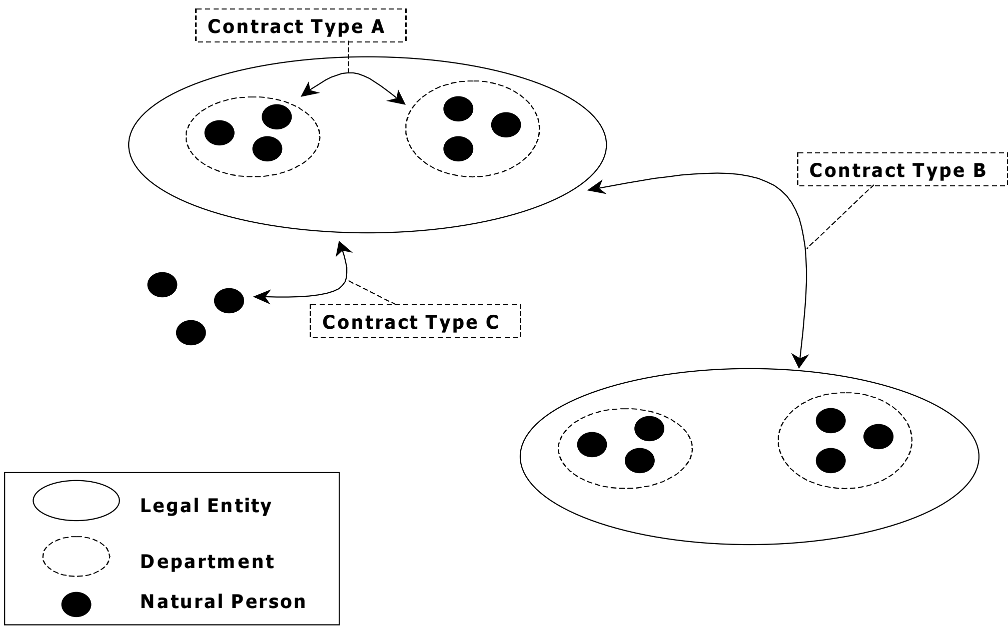


Figure 2-2. Contract definition within and between organizations

The different possibilities of contract definition as visualized in Figure 2-2, are:

- *Contracts defined within the organization itself (contracts of Type A)*. These contracts represent agreements made between an organization's internal departments. Examples include the formal agreement (contract) between the Financial Department and the Production Department to spend USD 10 million per quarter on operational expenses. One or more natural persons represent an internal department; e.g. the Chief Financial Officer represents the Financial Department. Under current business practices, this type of agreement is often not formalized or recorded as such and therefore not perceived as a contract. The use of contracts internally in organizations can be envisaged as a mechanism for the organization of the internal administration in an explicit way.

However, though internal agreements can be 'formalized', they are not legally enforceable as such. Further elaboration of the relevance of storing and maintaining this type of contract in the data model is presented in the next section.

- *Contracts defined by the interaction between the organization and other organization(s) in their environment (contracts of Type B)*. These contracts represent formal agreements between the organization and other (external) organizations. For example, a contract between the organization and a bank (organization) to obtain a new loan for the amount of USD 50 million. Because of the legal rights and obligations involved in these exchanges, this type of contract is usually formalized and stored.
- *Contracts defined as the result of interaction between the organization and other natural persons in their environment (contracts of Type C)*. These contracts represent formal agreements between the organization and external natural persons. An employment contract between an organization and an employee is an example of this type of contract. The legal nature of these agreements is again the main reason why they are often formalized and stored as a matter of ordinary business practice. However, this does not necessarily imply that employment contracts are always stored and maintained in the central data model.

These three different types of contracts comprise the relevant scope of possible agreements, the data from which can be used for decision-making. Traditional data storage methods such as double-entry bookkeeping typically focus on the storage of data pertaining to Type B contracts.

There are 3 types of contract data stored in the data model: *Type A* on contracts within an organization, *Type B* on contracts between organizations and *Type C* on contracts between natural persons and an organization. In broader terms, this implies that all contracts where the organization is one of the contracting parties should be recorded in the data model. There is a fourth type of contract as yet undiscussed, namely contracts made in the organization's operating environment where the organization is not one of the contracting parties. Data on this type of contract are not stored in the data model.

2.6.3 Contract enforceability and the importance of internal contracts

Contracts are proposed as the generalized data organization paradigm here because the 'contract principle' was found to be the recurring pattern of transaction features in the data of the relevant scope of BPIs. Since they can be defined within the organization itself or between the organization and participants in the external environment, as described above, the following questions arise. Can both contract types accommodate the same level of information and how are internal contracts to be applied when stored in information systems? Webster's Dictionary describes a contract as '*a binding agreement between two or more persons or parties, especially one that is legally enforceable*'. *Binding* contracts can be verbal, or legitimately recorded. However, since 'contracts' are considered here as a data organization paradigm to be used as the data source in an information system, only *stored* contracts are relevant. Besides being stored, contracts between the organization and external participants have the added characteristic of being legally enforceable. Every contractee has the option of demanding in court the enforced execution of the contract in the event of the failure of one of the other contractees to perform its terms. Stored contracts, which are legally enforceable, make up the formal information base of an organization from the viewpoint as described in Section 2.7.1.

Of importance is knowing what the advantage of defining and maintaining internal contracts explicitly could be, given the fact that in most organizations today, internal contracts are not stored as such. A reason for this could be the fact that the execution of internal contracts cannot be enforced before a court in the same way as external contracts. There are internal agreements that are irrelevant to record as no internal value exchange takes place within them, e.g. an agreement between two internal managers to organize a videoconference meeting every Monday at 10. However, when organizations consistently store internal contract data on value exchanges, they could have an information data model capable of accommodating data to service a much broader scope of information needs. The value of storing internal contracts is illustrated by two examples. Firstly, internal contracts could be defined by the agreements made between internal participants on their minimum expected performance, in relation to certain events. Contracts could thus be a means of ensuring accountability for performance. While not being legally enforceable, were data on these internal contracts stored, they could be used to ultimately support reward by bonus recommendation or facilitate the opposite, e.g. automatic resignation on poor performance. In contrast to the example on the agreement to organize a videoconference meeting as described earlier in this paragraph, internal contracts in such cases do concern a resource exchange. They detail the circumstances of the exchange of the resources of internal participants (i.e. agreed hours of working) against salary. Secondly, internal contracts could be defined as logical extensions of external contracts. Imagine a large contract with an external organization with penalties for late or bad execution defined by the external organization. Participants in the internal organization have to commit themselves to the terms in the external contract to execute the (external) contract successfully and predictably, thus avoiding possible claims. A way of handling this situation from an information perspective is to translate the external contract commitments into internal participant commitments, defined in one or more internal contracts. Should an internal participant not executing his or her part of the internal contract as agreed, he or she can be held accountable (e.g. loss of bonus as explained above) and the impact on the external contract becomes immediately apparent. These two examples illustrate that despite not being legally enforceable, internal contracts can contribute significantly to the predictable execution of external contracts in those situations where external contracts can be translated into internal contracts. However, from a functional perspective, this requires the possibility of relating contracts to each other. Once organizations start deploying internal contracts, the agency theory concept as discussed in Section 2.6.2 becomes applicable. For vertically defined contracts (see Section 2.7.4.1), there is a situation with a principal (e.g. a higher-level manager) and an agent. Situations of adverse selection and moral hazard do occur in the envisaged situations and the detail of data stored in the contract can anticipate a preferred course of action. For instance, performance bonuses can be defined in cases of moral hazard. Since in this dissertation the emphasis is on the registration aspects of contracts (how and when), agency theory is only illustrated as a supportive theory in the deployment of internal contracts but not considered further in the remainder of the thesis.

2.6.4 Relationships between contracts: The Contract Portfolio

2.6.4.1 Generic characteristics of the Contract Portfolio

The semantic means by which contracts are related to each other can be understood by exploring how contracts start playing a role from the moment of an organization's inception, followed by how it operates subsequently. The formal foundation of a new organization is stored in the foundation contract, the memorandum of association. This foundation contract is a formal letter of intent between the founders and the shareholders and serves as a means of communicating to the outside world the rights and obligations of these participants and the newly founded organization. The foundation contract is afterwards further detailed in a business plan detailing at a high level what the activities and potential markets of the organization are and how the plan can be financially realized. This contract details the rights and obligations of shareholders and creditors. The business plan is periodically updated and is valid over a strategic period of time (often 3 to 5 years). This business plan is further detailed in a ‘master plan’ covering the planning of activities for the coming year. This plan details the rights and obligations of the managers involved; it is a contract between, for instance, the production manager, the sales manager and the financial manager detailing what operational and financial resources are needed and what products and services will be offered over the next year and what the corresponding revenue and profit targets are. The execution of the master plan is further detailed in execution contracts like purchase and sales orders, hedging contracts, credit line agreements etc. This is visualized in Figure 2-3:

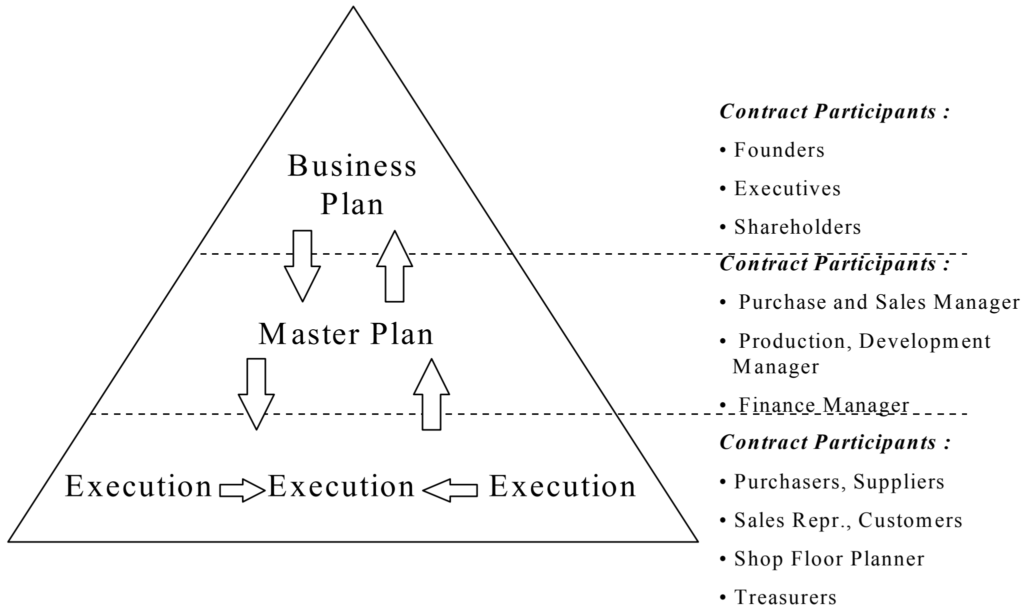


Figure 2-3. Relationship between contracts in an Organization

Figure 2-3 illustrates that contracts within organizations are *hierarchically* related. Vertical arrows indicate a hierarchical dependency between contracts. The master plan is derived from the business plan, but in the event that the master plan (contract) indicates a bottleneck, this can lead to a request to alter the business plan (contract). In Figure 2-2, three planning levels are envisaged. However, in real-life customer situations, the number of planning levels can vary based on the nature of the business.

In addition to hierarchical relationships, the figure also illustrates that contracts defined at the same hierarchical level can be dependent on each other. This is *horizontal dependency*, illustrated by a number of examples.

- *Budgeting*. The sales budget (contract) influences the production budget (contract) and the procurement budget (contract). A change in the sales budget can require a change in the other two budgets. In other words, these three budgets (contracts) are mutually dependent.
- *Financial contracts connected with operations contracts*. Financial contracts and operational contracts can be mutually dependent. For example, a purchase contract can trigger a letter of credit contract that defines the payment of this contract. A sales contract can trigger a hedging contract to cover the risk of foreign currency fluctuations. If the number of operations contracts changes, the number of financial contracts should change accordingly, or vice versa.

This type of relationship is important because financial contracts often depend on operations contracts. As explained in the examples above, operational contracts and financial contracts are mutually dependent. However, in current business practice, financial contracts and operational contracts are often defined and agreed upon more or less separately. For example, financial departments often plan the volume of exposure in a particular currency once a year and afterwards hedge the risk for that amount, without relation to the actual operational contracts maintained.

2.6.4.2 Example: Identification of Financial Contracts in the total Contract Portfolio

The contract portfolio consists of contracts belonging to different functional domains, e.g. operations contracts, human resources contracts, financial contracts, etc. The financial contracts are a subset of the contracts in the contract portfolio. The various contracts are linked to each other through horizontal and vertical relationships, and the network of contracts (in other words, the structure of the contract portfolio) is maintained. The position of financial contracts at various levels in the contract portfolio is illustrated and visualized in Figure 2-3 as an example of the contracts in the contract portfolio. Some examples of financial contracts at different levels are described below.

- *Business Plan*. A part of the business plan deals with financial contracts. These financial contracts explain how the operational and commercial parts of the business plan will be financed. Examples of financial contracts at this level include a capital-raising agreement, a long-term bank loan agreement, bonds, etc.
- *Master Plan*. Financial contracts defined at the master plan level, define how the financing of activities for the next year is to be outlined. Examples include credit line agreements, letters of credit and bills of exchange, hedging contracts to cover currency risks, etc.
- *Execution*. The operational financial plans are executed at this level. Financial contracts defined at this level are mainly fulfilments that detail how payment and collections of financial resources take place. Some examples include bank drafts, cheques, bills of exchange, overnight investments, etc.

This is visualized in Figure 2-4:

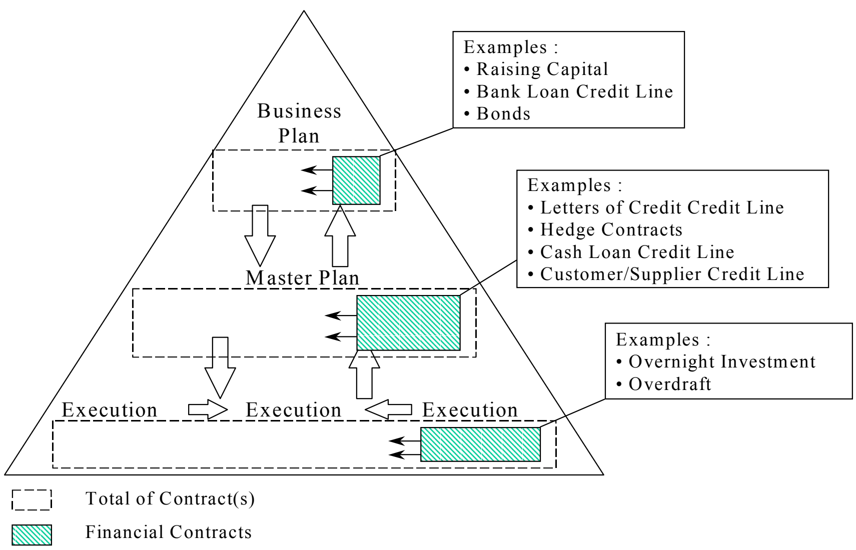
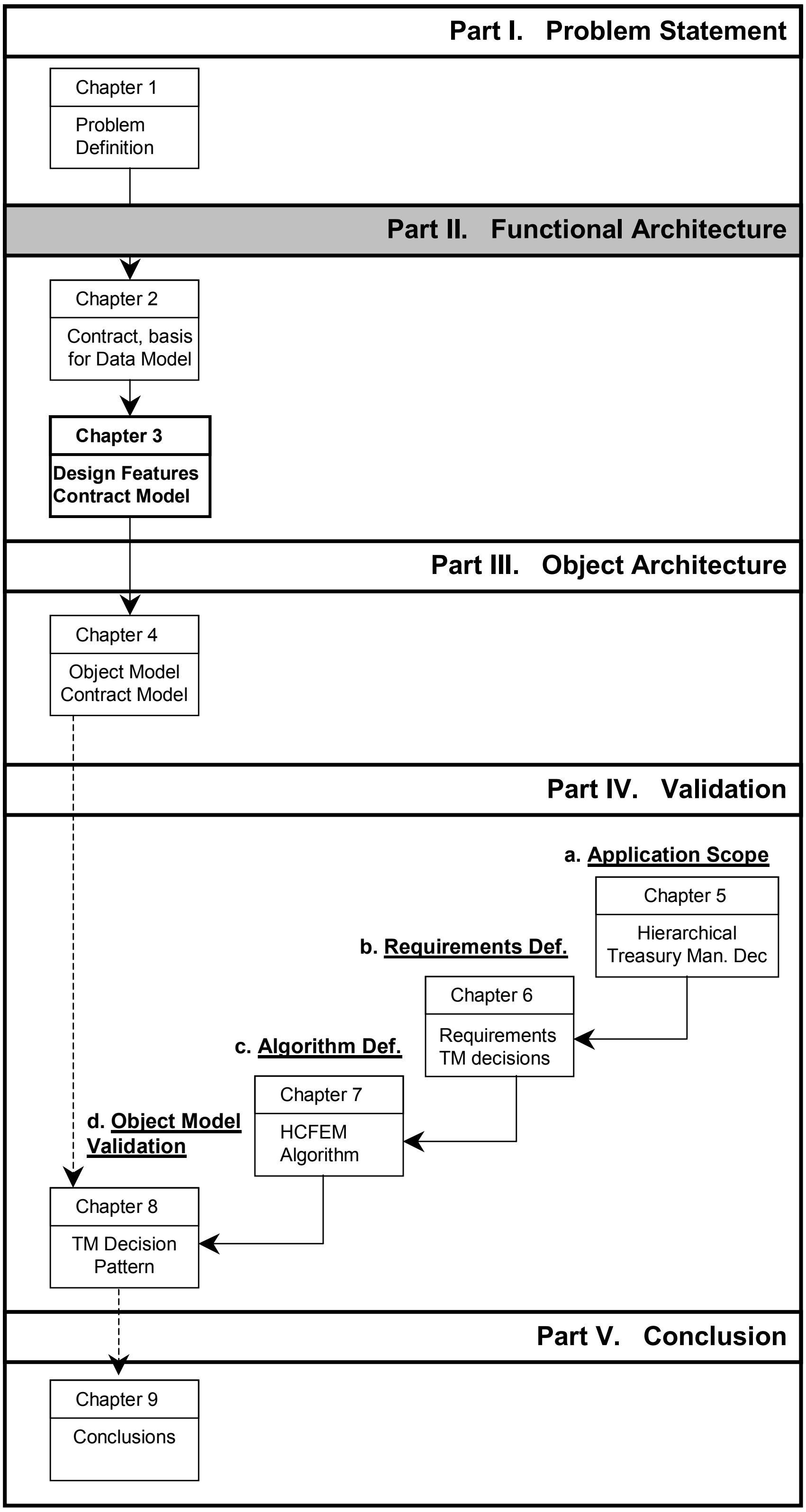


Figure 2-4. Position of Financial Contracts in the Contract Portfolio

Figure 2-4 illustrates that financial contracts can be defined at different levels. Financial contracts can be related to other contracts at the same level. These relationships express a mutual dependency between the contracts (see also Section 2.7.1). Financial contracts can also be related to other financial contracts at different levels. These relationships visualize that lower-level contracts are derived from higher-level contracts as explained in the previous section.

2.7 Summary

Double-entry bookkeeping data is still the most important data source in organizations despite the fact that attention has long been drawn by scientists to its limitations. Accounting data models providing 'value-free' BPI data have been suggested by the American School (e.g. McCarthy's REA model) and the German School (e.g. Schmalenbach's 'Grundrechnung') but have not been adopted in the data model architectures of today's ERP systems. Authors like McCarthy (1982), Everest and Weber (1977) and Verdaasdonk (1998) have noted that data models appropriate for ERP systems should be designed following a data organization pattern close to aspects of reality. In this chapter, the recurring pattern characterizing transaction features of various BPI types was found to be the exchange of resources as stored in a formal agreement or contract. 'Contracts' were therefore chosen as the proposed data organization paradigm close to aspects of reality, and the activity of administrating contracts was introduced as 'Contract-based accounting'. The relationship between contracts and organizations is important, as on-going data accommodation has to be ensured even when the boundaries between organizations change. Various aspects of this relationship were described and the importance of also storing internal contracts was highlighted. Furthermore, it was explained that contracts are defined in organizations by highly structured means. In this context, an analogy with a 'contract portfolio' was drawn. A subsequent step in the design of a suitable data model for ERP implementation is the definition of the essential data characteristics of the contract data model. These essential data characteristics relate to data components that should be adopted in the data model as objective and value-free and derived from contract features. This will be the subject of investigation in Chapter 3.


Part I. Problem Statement
Chapter 1
Problem Definition
Part II. Functional Architecture
Chapter 2
Contract, basis for Data Model
Chapter 3
Design Features Contract Model
Part III. Object Architecture
Chapter 4
Object Model Contract Model
Part IV. Validation
a. Application Scope
Chapter 5
Hierarchical Treasury Man. Dec
b. Requirements Def.
Chapter 6
Requirements TM decisions
c. Algorithm Def.
Chapter 7
HCFEM Algorithm
d. Object Model Validation
Chapter 8
TM Decision Pattern
Part V. Conclusion
Chapter 9
Conclusions