The Interplay of Mandatory Role and Set-Comparison Constraints

Peter Bollen

Department of Organization and Strategy
School of Business and Economics
Maastricht University
P.O. Box 616
6200 MD Maastricht, The Netherlands
p.bollen@maastrichtuniversity.nl

Abstract. In this paper we will focus on the interplay of mandatory role and set-comparison (equality-, subset- and exclusion-) constraints in fact based modeling. We will present an algorithm that can be used to derive mandatory role constraints in combination with non-implied set-comparison constraints as a result of the acceptance or rejection of real-life user examples by the domain expert.

Keywords: Mandatory role constraint, set-comparison constraint, implied constraint, CSDP.

1 Introduction

Fact-based modeling (FBM) is a conceptual modeling approach that is rooted in the field of conceptual modeling for database systems [11]. The features of FBM have inspired the development of a standard language for business rules (SBVR) in which the fact type construct and many other essential FBM modeling constructs have been incorporated [6]. FBM's *conceptual schema design procedure* (CSDP) [9, 13] can be considered one of the earliest business rule derivation technologies [12]. Some examples of contemporary incarnations of FBM are ORM [9], CogNiam [15] and FCO-IM [2].

In this paper we will investigate those steps in the CSDP that derive mandatory role and set-comparison constraints. We will present a mandatory role constraint derivation algorithm in tandem with the algorithm for deriving set-comparison constraints (as presented in [7: p. 332-333]). By combining the derivation procedure for these two constraint types we can avoid duplication in the process of business rule extraction by simultaneously deriving instances of both constraint types. Therefore, the main objective of this paper is to derive an algorithm that can be used to derive all mandatory role and set-comparison constraints in one procedure. This paper will lean heavily on the results that were presented in [7].

The paper is structured as follows. In section 2 we will define the mandatory role and set-comparison constraints. In section 3 we will investigate in what ways

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 328–337, 2012.

mandatory role and set-comparison constraints can be derived in a CSDP. In section 4 we will present an integrated algorithm for deriving mandatory role and set-comparison constraints. In section 5 we will consider the integration of conceptual schemas. Finally in section 6 conclusions will be given.

2 Definition of Mandatory Role and Set-Comparison Constraints

In the FBM defining literature a mandatory role is defined as follows: ‘A role is mandatory if and only if, for all states of the database, the role must be played by every member of the population of its object type; otherwise the role is optional’ [9: p. 162]. When it comes to defining mandatory role *constraints* in a conceptual schema, ORM-(2) allows to leave out explicit mandatory role constraints if a (primitive) object type plays only one role [9: p.164, 14: p.115]. In order to explicitly show that object types can ‘exist’, without having to play at least one fact role, ORM has defined the concept of *independent object type* as ‘a primitive object type whose fact roles (if any) are collectively optional’ [9: p. 291]. In Fully Communication Oriented Information Modeling (FCO-IM), independent object types can be modeled using existence postulating fact types [1]. In CogNiam (and some of its predecessors) [10, 13] independent object types are implied whenever all roles of the fact types in which they are involved have the status *optional*.

Set comparison constraints are population constraints that restrict the way in which the population of (a) role (combinations) relates to the population of another role (combination) [9: p. 224]. The three set comparison constraint type are the *subset, equality* and *exclusion* constraints. For an in-depth coverage of these constraint types we refer to [9: p.224-228].

3 The Derivation of Mandatory Role and Set-Comparison Constraints

In some knowledge domains the business rules might be provided in such a way that mandatory roles and set-comparison constraints can be easily added to a fact type diagram (e.g. the outside-in approach). If we for example consider the following fact types: *Person lives at Address* and *Person owns Car* and the following documented business rule: *For every Person an Address must be recorded*, we can see that this business rule maps straightforward to a mandatory role constraint defined on the *Person* role of the *Person lives at Address* fact type. In some exploratory knowledge domains, the constraints that govern the domain can only be detected by rigorously applying a conceptual schema design procedure in which the FBM analyst will meticulously present permutations of domain examples to the domain expert in order to derive instances of domain constraints. It is for these domains specifically that we need an integrated mandatory role and set-comparison constraint derivation algorithm.

In the ORM conceptual schema design procedure the mandatory role constraints are derived in step 5 and the set-comparison constraints are derived in step 6 [7, 9]. Some of the set-comparison constraints that will be derived in step 6 especially the subset and equality constraints, therefore will be ‘implied’ [8: p.6.8-6.26].

In the CogNIAM fact-based dialect [15] (and its predecessors NIAM2007 [10] and Kenniskunde [13]) the mandatory role constraints are derived after the set-comparison constraints. In this application sequence of constraint derivation sub-procedures, some mandatory role constraints can potentially be implied (by a configuration of set-comparison constraints).

3.1 Recap of the Procedure for Deriving Set-Comparison Constraints

In [7: p.332-333] we have given a fully specified derivation procedure in which in an analyst-user dialogue, exclusively based on the acceptance or rejection of 'real-life' domain examples or 'data use-cases' all set-comparison constraints that exists in an application domain can be detected. This algorithm is presented here as a decision-table in which for a given combination of allowed existence or non-allowed existence of each of the example extensions (at most) one set-comparison constraint will be derived (see table 1).

Table 1. Decision logic from the set-comparison constraint derivation algorithm [7, p.334][1]

	1	**2**	**3**	**4**
EXT1 pop(Ft1)= {a1} pop(Ft2)= empty	Allowed	Not Allowed	Not Allowed	Allowed
EXT2 pop(Ft1)= {a1} pop(Ft2)= {a1}	Allowed	Allowed	Allowed	Not Allowed
EXT3 pop(Ft1)= {a1} pop(Ft2)= {a1, a2}	Not Allowed	Allowed	Not Allowed	Not Allowed
Constraint type	Subset1 (FT2->FT1)	Subset2 (FT1->FT2)	equality	exclusion

The scope of the set-comparison algorithm in [7] is one pair of role(s) (combinations) in which in principle the same object type(s) (or nested object type(s) or a combination of (an) object type(s) and (a) nested object type(s)) are involved. It should be noted that the set-comparison constraint configuration that is derived by executing this algorithm for all possible pairs of role(s) combinations of a given object type (or nested object type(s) or a combination of (an) object type(s) and (a) nested object type(s)) is complete, and therefore contains all implied set-comparison constraints. In the example in figure 1 set comparison constraint *C3* is implied by the combination of subset constraints *C1* and *C2*. The application of the algorithm will always lead to the 'complete'[2] configuration of set comparison constraints as exemplified by the information model in figure 1(A).

---

[1] We note that for the other 4 (2^3-4) possible outcomes of the algorithm no set-comparison constraints will be derived.

[2] Complete means that all implied set-comparison constraints are derived.

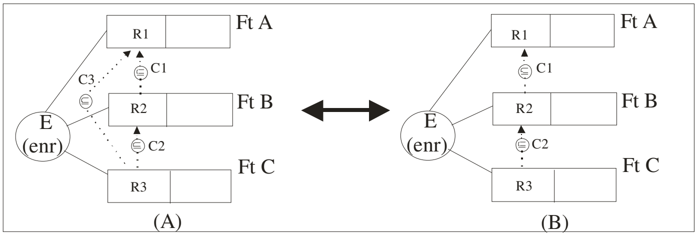

Fig. 1. Equivalent subset constraint configurations (example 1)

4 An Integrated Algorithm to Derive Mandatory Role and Set- Comparison Constraints

We will now take the set-comparison constraint derivation algorithm in [7] as a starting point for an integrated algorithm (see algorithm A on next page) in which we can 'derive' the mandatory role constraints by analyzing the subset and equality constraint configurations that are centered around a given object type. It is further assumed that the 'independence' status [9: p.219] of this object type is known. The integrated algorithm will contain the set-comparison derivation algorithm from [7]. The outcome of the first step in the algorithm will contain all set-comparison constraints between pairs of role(s) (combinations), and will therefore lead to a constraint configuration as is shown in figure 1(A) which is tabled in table 2(A).

Table 2. Tabular format set- comparison constraints

(A)

From \ To	R1	R2	R3
R1			
R2	⊆		
R3	⊆	⊆	

(B)

From \ To	R4	R5	R6	R7
R4		⊆		
R5	⊆			
R6	⊆	⊆		
R7	⊆	⊆		

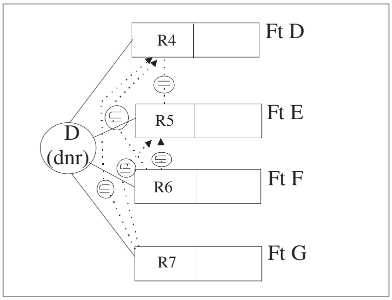

Fig. 2. Example subset constraint configuration (example 2)

Figure 2 will be replaced by the subset constraint entries: *from R2 to R1* and *from R1 to R2* in table 2(B). The next step in the algorithm is an inspection of the set-comparison constraint tables as exemplified in table 2. For each column in which all

Cells contain a subset constraint, a mandatory role will be defined on the column role In table 2(B) we have given the tabular format for the set-comparison constraints in example 2 (see figure 2). It should be noted that an *equality* constraint in the schema will be replaced by two *subset* constraints each opposite to the diagonal from the table. For example the equality constraint between roles R1 and R2 of the example in

('…*A* has a mandatory role *r* only if there is a subset constraint to *r* from each of *A*'s roles' [9: p. 226]) when the role playing object type is not independent.

Algorithm A

```
  BEGIN Algorithm derive set-comparison and mandatory role
      (fact type model,object type).
      WHILE still rolecombinations left
      DO Take next_2_role_combination
       Algorithm SETCOMPARISON_on_2_rolecombinations(see [7])
      ENDWHILE3
      create a tabular format by
      1. copying subset constraints 1-on-1 into the table
      2. copying exclusion constraints into exactly one
       of the applicable table cells.
      3. copying an equality constraint as 2 subset
   constraints, mirrored across the diagonal of the table
   4. remove all copied set-comparison constraints from the
information model
       take the first column from the table
       WHILE still columns left
       DO    Take next column. Check all entries in the column
       IF (all column entries = subset constraint AND object
             type IS NOT independent)
       THEN mandatory role defined for the role on the
       column. Remove subset constraints from column
       ENDIF
       ENDWHILE
       WHILE still constraint entries in table
       DO check next constraint
             IF constraint=subset
             THEN check for mirrored constraint
                    IF mirrored constraint exist
                    THEN add equality constraint in model
                     Remove both subset constraints fr table
                    ELSE add subset constraint in model
                     Remove sub-set constraint from table
                    ENDIF
             ELSE add exclusion constraint to model
                    Remove exclusion constraint from table
             ENDIF
       ENDWHILE
  Remove implied constraints4
  END
```

[3] We will refer to the bold italic segment of the algorithm as procedure SETCOMPOBJ(object type).

[4] In [8, section 6] an overview is given of a number of implied constraints for set-comparison and mandatory role constraint configurations. The algorithm can remove the implied constraints in for example the following cases: [(A is subset of C) AND (C is subset of B) → (A is subset of B)]; [(C is subset of A) AND (B is subset of C) → (B is subset of A)]; [(A is disjunct with B) AND (A = C) → (C is disjunct with B)].

After we have removed the content of the cells for the mandatory role columns we can map the remaining constraints back to the conceptual schema in which case the mirrored sub-set constraint pairs will be transformed back to equality constraints. The outcome of this algorithm applied on examples 1 and 2 will lead to the conceptual schemas as given in figure 3.

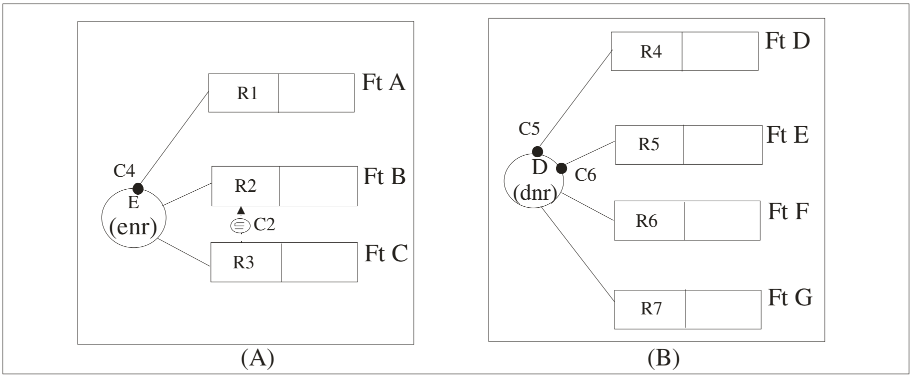

Fig. 3. Final mandatory role and set-comparison configurations for examples 1 (A) and 2 (B)

4.1 An Example of the Application of the Set-Comparison and Mandatory Role Derivation Algorithm

In example 3 (see figure 4) we need to investigate the mandatory role and set-comparison constraint configuration centered around the *TaxPayer* object type. The example in figure 4 contains the set-comparison constraints that have been derived using the set-comparison algorithm from [7].

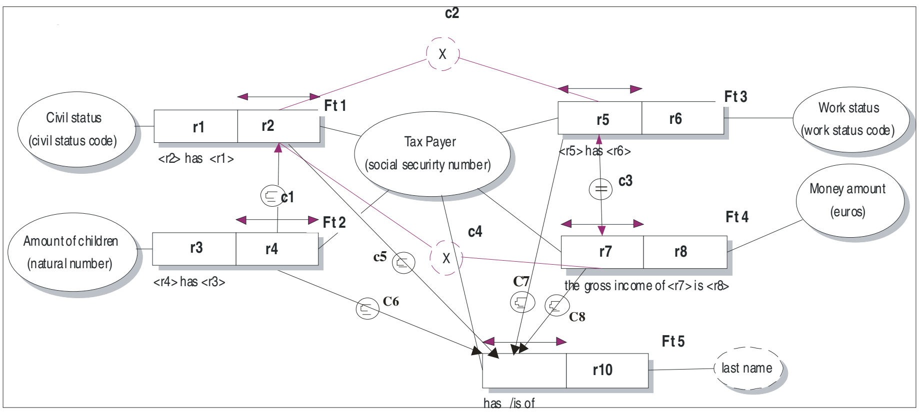

Fig. 4. Example 3 (based upon the example in [7])

Table 3. Transformation of tables in set-comparison and mandatory role algorithm

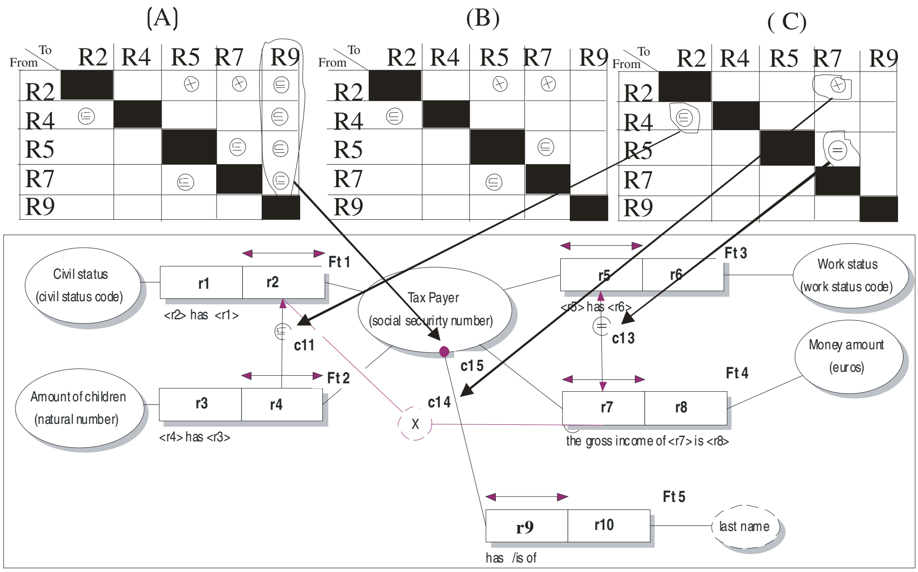

Fig. 5. In-between results and result of applying the algorithm derive set-comparison and mandatory role constraints

The constraint configuration in table 3(A) points at the existence of mandatory role constraint *c15* in figure 5. After removing the subset constraints in the mandatory role column (table 3(B)) and the collapse of 'opposite' subset constraints from table 3(B) into an equality constraint in table 3(C), and the removal of the implied exclusion constraint (from role *R2* to role *R5*) we can map the table entries in table 3(C) onto constraints *c11*, *c13* and *c14* of the conceptual schema, respectively.

5 The Integration of the Mandatory Role and Set-Comparison Constraints from Local Sub-schemas to a Global Schema

In [5: p.135] it was already shown that by integrating two conceptual schemas, (some of) the mandatory role constraints in the local conceptual schema(s) cannot directly be copied into the integrated schema that captures the global 'semantics' of the local mandatory role constraint. We therefore, propose to apply the set-comparison constraint derivation algorithm in [7] first when analyzing the 'local' UoD's. in figures 6 (A) and 6 (B) we have given the completed (local) conceptual schemas for UoD1 and UoD2. In figures 6(C) and 6(D) we have given the proto-conceptual schemas for UoD1 and UoD2.

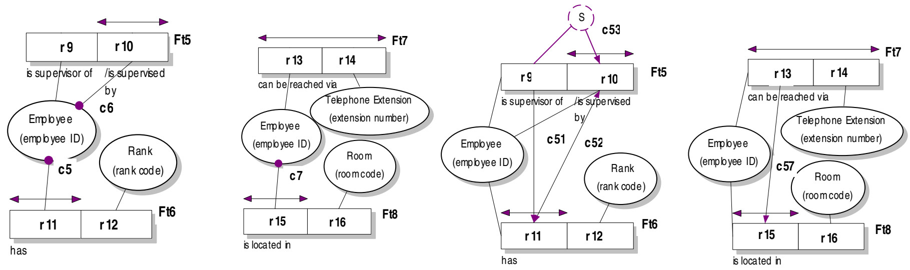

Fig. 6. (A). CS UoD1 **(B).** CS UoD2 **(C).** Proto CS UoD1 **(D).**Proto CS UoD2

In UoD1 we have fact types and constraints that represent the following (local) domain semantics:

Each employee has exactly one rank.
Each employee has exactly one supervisor
Each supervisor is an employee.

In UoD2 we have the following (local) domain semantics[5]:

Each employee is located in exactly one room
An employee can be reached via zero, one or more telephone extensions.

If we now want to integrate the completed conceptual sub-schemas for UoD1(figure 6(A)) and UoD2 (figure 6(B))we can merge the fact types and uniqueness constraints into one integrated conceptual schema. However, we can not merge the mandatory role constraints, because in the integrated (global) UoD the mandatory participation in some roles does no longer hold. It is because we have to consider the 'global characteristics' that emerge when we add 'individual views'. These characteristics can lead to new and/or adapted constraints that have to be added to/adapted in the model of the 'global view' [4: p. 338].

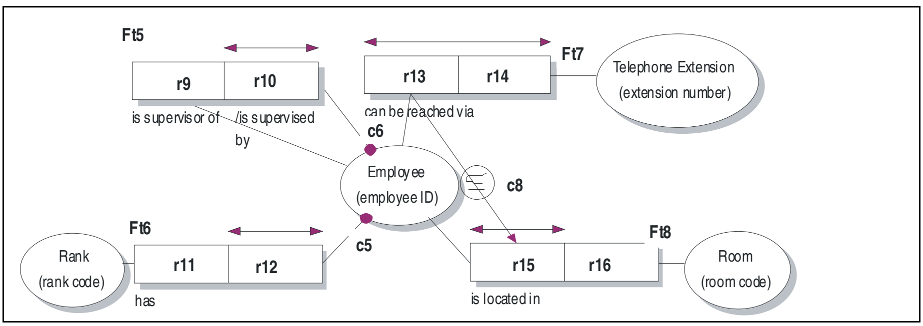

Fig. 7. Conceptual schema integrated UoD (IUoD)

[5] Note that in UoD2 only a subset of the 'global' employee population is relevant: those employees that are located in a room. This explains the existence of mandatory role constraint *c7* in figure 6(B).

We have summarized the semantics of the integrated domain (IUoD) as follows[6]:

Each employee has exactly one rank.
Each employee has exactly one supervisor
Each supervisor is an employee.
An employee can be located in a room.
An employee that is located in a room can be reached via
zero, one or more telephone extensions

If we carefully analyze the conceptual schema of the integrated UoD in figure 7 we notice that the result of the integration of the local schemas into a global schema with global semantics has had the following implications in terms of the mandatory role and set-comparison constraints: mandatory role constraint *c7* in the local schema has been replaced by subset constraint *c8* in the global schema (see figure 7). In this case it is recommended to apply the set-comparison derivation algorithm in [7] for the creation of the local proto conceptual schemas. In an integration step of two (or more) proto conceptual schemas the same algorithm can be applied on those new role combinations that have emerged from the federation of the local proto conceptual schemas into the new 'global' proto-conceptual schema (procedure beta in figure 8). The latter schema can now serve as input for procedure gamma (in figure 8), i.e. the application of the 2nd part of the set-comparison and mandatory role derivation algorithm (A) to map the set-comparison constraints to mandatory role and remaining (non-implied) set comparison constraints (see figure 8).

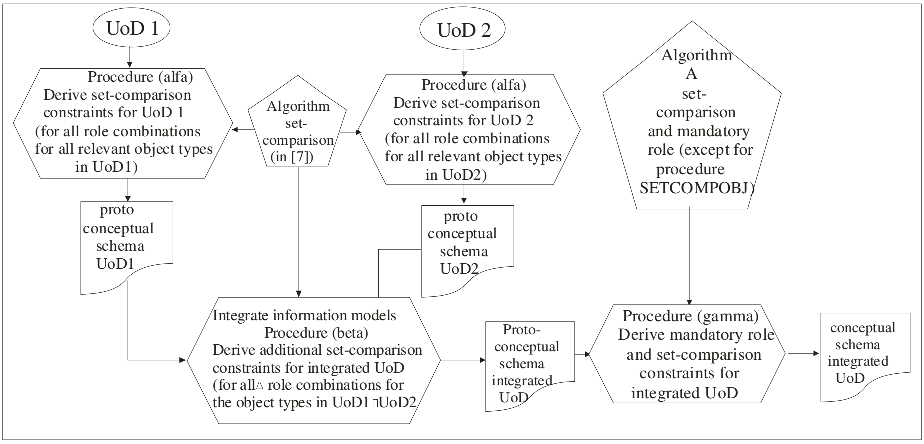

Fig. 8. The application of the mandatory role and set comparison algorithm for integrating (conceptual) sub-schemas

6 Conclusion

In this article we have shown that the algorithm for the 'example-based' derivation of set-comparison constraints can be used as a starting point for an integrated algorithm

[6] Note that in the integrated UoD not all employees have to be located in a room.

that will generate the mandatory role and the non-implied set-comparison constraints. We have also shown that for the integration of multiple 'local' conceptual schemas, the logic of the algorithm allows us to integrate those sub-schemas with a minimum of integration effort. By expressing the mandatory role constraints in the local (proto) schema as (combinations of) set comparison constraints the process of data federation can focus on the derivation of set-comparison constraints for the 'new' role combinations that govern the integrated UoD.

References

1. Bakema, G.P., Zwart, J.P., van der Lek, H.: Fully communication oriented NIAM. In: Nijssen, G., Sharp, J. (eds.) NIAM-ISDM 1994 Conference (1994)
2. Bakema, G.P., Zwart, J.P., van der Lek, H.: Fully communication oriented information modeling (2002), `http://www.fco-im.nl/pdfFiles/FCO-IM%20book.pdf`
3. Balsters, H., Halpin, T.: Modeling Data Federations in ORM. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 657–666. Springer, Heidelberg (2007)
4. Batini, C., Lenzerini, M., Navathe, S.: A comparative analysis of methodologies for database schema integration. ACM Computing Surveys 18(4), 323–336 (1986)
5. Bollen, P.: A formal transformation from object-role models to UML class diagrams. In: Halpin, T., Siau, K., Krogstie, J. (eds.) Seventh CAiSE/IFIP-WG8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2002), pp. 132–143 (2002)
6. Bollen, P.: SBVR: A Fact-Oriented OMG Standard. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 718–727. Springer, Heidelberg (2008)
7. Bollen, P.: A Derivation Procedure for Set-Comparison Constraints in Fact-Based Modeling. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM-WS 2011. LNCS, vol. 7046, pp. 329–338. Springer, Heidelberg (2011)
8. Halpin, T.: A logical analysis of information systems: static aspects of the data-oriented perspective. Ph.D thesis. Department of Computer Science. University of Queensland (1989)
9. Halpin, T., Morgan, T.: Information Modeling and Relational Databases; from conceptual analysis to logical design, 2nd edn. Morgan-Kaufman, San-Francisco (2008)
10. Lemmens, I.M.C., Nijssen, M., Nijssen, S.: A NIAM2007 Conceptual Analysis of the ISO and OMG MOF Four Layer Metadata Architectures. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 613–623. Springer, Heidelberg (2007)
11. Nijssen, G.: On the gross architecture for the next generation database management systems. In: Information Processing. IFIP (1977)
12. Nijssen, G.: Mini cookbook to prepare a conceptual schema or conceptualization procedure in fourteen major steps, University of Queensland, Australia (1982), `http://www.sjirnijssen.eu/documents.aspx`
13. Nijssen, G.: Kenniskunde 1A. PNA publishing, Heerlen (2001)(in Dutch)
14. Nijssen, G., Halpin, T.: Conceptual schema and relational database design: a fact oriented approach. Prentice-Hall (1989)
15. Nijssen, G., Le Cat, A.: Kennis gebaseerd werken: de manier om kennis produktief te maken. PNA publishing, Heerlen (2009) (in Dutch)