

A derivation procedure for mandatory role and set-comparison constraints in fact-based modelling

Citation for published version (APA):

Bollen, P. W. L. (2011). *A derivation procedure for mandatory role and set-comparison constraints in fact-based modelling*. METEOR, Maastricht University School of Business and Economics. METEOR Research Memorandum No. 036 <https://doi.org/10.26481/umamet.2011036>

Document status and date:

Published: 01/01/2011

DOI:

[10.26481/umamet.2011036](https://doi.org/10.26481/umamet.2011036)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

P. Bollen

**A derivation procedure for
mandatory role and set-
comparison constraints in fact-
based modeling**

RM/11/036

METEOR

Maastricht University School of Business and Economics
Maastricht Research School of Economics
of Technology and Organization

P.O. Box 616
NL - 6200 MD Maastricht
The Netherlands

A derivation procedure for mandatory role and set-comparison constraints in fact-based modeling

Dr. 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
phone: 31-43-3883715
fax: 31-43-3884893

Abstract. In this paper we will address the conceptual schema design procedure (CSDP) in fact-based modeling. We will focus on the modeling procedure of ‘cook-book’ for deriving the set-comparison and mandatory role constraints. We will give an algorithm that can be applied by an analyst in an analyst-user dialogue in which all set-comparison constraints can be derived as a result of the acceptance or rejection of real-life user examples by the domain expert.

1 Introduction

In conceptual information modeling, a number of approaches have evolved over the past decades. In the mid-70’s two (families of) approaches to semantic information modeling were dominant. The first approach and most popular approach was entity-relationship (ER) modeling [1]. This approach has evolved and has been extended into numerous flavours of (Extended) Entity-Relationship ((E)-E-R) approaches [2, 3]. A second approach that emerged in the mid-seventies was the fact-based approach. The earliest incarnation of this approach was the ENALIM approach [4], that later evolved into binary NIAM [5] and subsequently into N-ary NIAM [6]. The more contemporary incarnations of these approaches are the UML class diagrams [7] for the ER family of approaches and ORM (-2) [8, 9] and CogNIAM [10, 11] for the fact-based approaches.

The difference between the ER and fact-based modeling approaches mainly lie in the fact that the fact-based modeling approach knows only one fact-encoding construct: the fact type, in contrast to the (extended) E-R approaches in which there are (at least) two fact encoding constructs: the attribute and the relationship [12]. Another major difference between the two approaches lies in the presence of a modeling methodology or procedure in the fact-based approach [6, 13], whereas such a procedure is absent in the other approaches.

In this article we will investigate the modeling methodology or conceptual schema design procedure in fact-based modeling. We will propose a precise specification of the conceptual schema design procedure (CSDP) [6, 8, 9] in which we will give a formal modeling procedure to derive set-comparison constraints [14] that

is a further specification of steps 5 and 6 of the CSDP [8]. This specification will consist of an algorithm that can be applied by an analyst in an analyst-user dialogue leading to a complete procedure-driven derivation of all set-comparison constraints in a given Universe of Discourse (UoD).

1.1 The running example: tax collection

In this article we will use as a running example, a simplified tax-collection universe of discourse. As a concrete example we will use the following ‘tax-collecting’ form from the (fictive) country ‘Utopia’. In the Utopian tax collection UoD citizens are requested to fill-in standardized tax collection forms. An accepted tax-form, however, has to comply to certain rules. In this article we will assume that these rules will be derived using an ‘inside-out’ approach [15] , which means that we will apply the explicit steps of the Conceptual Schema Design Procedure to derive these rules.

Utopian tax collection form 2011	
Social security number: 354678	
If you will become 27 years or older in this fiscal year please fill in part 1 otherwise part 2	
<p>PART 1</p> <p>Are you married: <input checked="" type="checkbox"/> yes <input type="checkbox"/> no</p> <p>If you have children please fill in the answer to the next question.</p> <p>How many children do you have ? : 3</p>	<p>PART 2</p> <p>Your current status is: <input type="checkbox"/> student <input type="checkbox"/> employee <input type="checkbox"/> unemployed</p> <p>Your gross income this year was:</p>

Fig 1. Example ‘Utopian’ tax collection form.

2 The conceptual schema design procedure in fact based modeling

The fact-based conceptual modeling methodology distinguishes itself from the (E)ER class of modeling approaches not only in the sense that it contains exactly one fact encoding construct instead of two, but also in the presence of a modeling methodology that can be used to create not only a syntactically correct model instance but also a semantically correct (conceptual) model for the Universe of Discourse. We will illustrate this procedure by introducing a running example from the domain of tax-collections. We will also give a formalization of the ‘well-known’ procedure for deriving uniqueness constraints in step 4 of the CSDP [8].

2.1 Overview of the conceptual schema procedure

The first complete publication of the conceptual schema design procedure was documented in [6]. The most recent version of the CSDP can be found in [8]. In table 1 we compare these versions of the CSDP.

Table 1: Comparison of steps in CSDP

CSDP Steps	Nijssen and Halpin (1989)	Halpin and Morgan (2008)
step 1	From examples to elementary facts	From examples to elementary facts
step 2	First draft of conceptual schema diagram	Draw fact types and populate
step 3	Trim schema and find derived fact types	Trim schema; note basic derivations
step 4	Add Uniqueness constraints	Add uniqueness constraints and check arity of fact types
step 5	Check arity of fact types	Add mandatory role constraints and check for logical derivations
step 6	Add entity type, mandatory role, subtype and occurrence frequency constraints	Add value, subset, equality, exclusion and subtype constraints
step 7	Check Entity identification schemes	Add other constraints and perform final checks
step 8	Add equality, exclusion, subset and other constraints	-
step 9	Check that the conceptual schema is consistent with the original examples, has no redundancy, and is complete	-

If we now inspect table 1 we can conclude that the first three steps in the old and new CSDP are identical, furthermore, the ‘new’ step 4 is a combination of the ‘old’ steps 4 and 5. The ‘new’ steps 5, 6 and 7 have a slightly different order in which the ‘sub-procedures’ are executed. The ‘old steps’ 7 and 9 are integrated into the ‘new’ step 7. The old step 8 is integrated into the ‘new’ steps 6 and 7.

2.2 A formal description of the ‘new’ step 4 in the CSDP

Once the first draft of a conceptual schema is created for a specific UoD, the analyst can elicit additional business rules from the domain user(s) by systematically confronting him/her (them) with new (combinations of) ‘real-life’ domain examples. The domain user only needs to confirm or reject the possibility that such a (combination of) examples can exist. In this section we will give an illustration of such an algorithm for the group of (intra fact-type) *uniqueness constraints*. Uniqueness constraints will constrain the occurrence of two or more fact instances in which a subset of the roles have *identical* value combinations. It is assumed that the basic information model consists of *elementary* fact types. This assumption has been embedded in the uniqueness constraint derivation algorithm 1.

Algorithm 1. Uniqueness constraints derivation

```

BEGIN UNIQUENESS(IBIM) [IBIM is the trimmed conceptual schema]
WHILE not last fact type
DO take a random sentence instance from a complex fact
type template for this fact type from the example
UoD:  $(a_1, \dots, a_N): FT_j \in IBIM$ 
Take the first role from this fact type
WHILE not last role in fact type
DO Create an example sentence where the instance
of role  $m$  is altered. Determine whether the
combination of this sentence with the first
sentence is allowed
IF the existence of such a sentence is allowed
together with  $(a_1, \dots, a_N)$ 
THEN add this sentence to the uniqueness
significant population
ELSE define a uniqueness constraint UC on
roles  $\{1, \dots, N\} \setminus m$ 
ENDIF
Go to the next role in fact type
ENDWHILE
Take next fact type
ENDWHILE
END

```

Implementations of this algorithm in an analyst-user knowledge extraction procedure can be found in [16]. After the uniqueness constraint derivation procedure has been applied the analyst can add the uniqueness constraints to the application's initial conceptual schema (see figure 2).

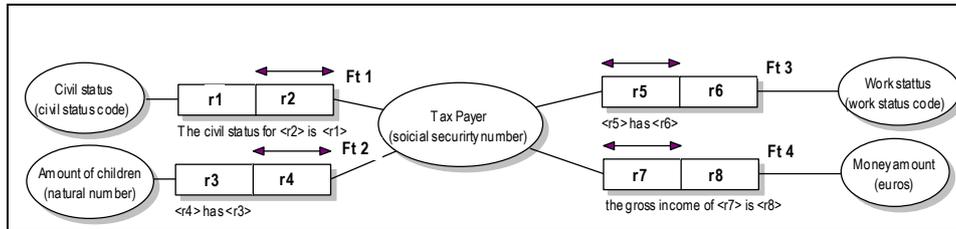


Fig 2. Tax collection conceptual schema including uniqueness constraints.

3 A formal derivation procedure for mandatory role and set-comparison constraints (the new steps 5 and 6)

In steps 5 and 6 of the CSDP we need to add the mandatory role, equality-, subset- and exclusion (set-comparison) constraints. A definition of set-comparison constraints, together with numerous example of these constraints can be found in section 6.4 of [8]. In this paper we will present a formal algorithm that can be used by an analyst to generate a minimal set of examples that in an analyst-user dialogue can be used to derive all set comparison constraints that govern the subject domain. The analysis is based exclusively on the acceptance or rejection of a set of (combinations) of real-life examples by the domain user in a dialogue with the (fact-based) business analyst.

Algorithm 2¹: Set comparison constraint derivation.

BEGIN SETCOMPARISON_on_2_rolecombinations
 let (R_1, \dots, R_N) and (R_{N+1}, \dots, R_{2n}) ² be the role combinations on which the set comparison should be performed.
 Let (a_1, \dots, a_M) be a sentence instance of the fact type (FT1) that contains roles (R_1, \dots, R_N) ($M \geq N$)

Let b_{N+1}, \dots, b_{2N+L} and g_{N+1}, \dots, g_{2N+L} be sentence instances of the fact type (FT2) that contains roles (R_{N+1}, \dots, R_{2n}) ($L \geq 0$).

Furthermore, let $IM := \{FT1, FT2\}$.

Create three user examples that reflect following fact type extensions (FT1+FT2):

$EXT_1(IM) : \{ (a_1, \dots, a_M) \}$

$EXT_2(IM) : \{ (a_1, \dots, a_M), (b_{N+1}, \dots, b_{2N+L}) \mid a_1 = b_{N+1} \wedge \dots \wedge a_N = b_{2N} \}$

$EXT_3(IM) : \{ (a_1, \dots, a_M), (b_{N+1}, \dots, b_{2N+L}), (g_{N+1}, \dots, g_{2N+L}) \mid$

$a_1 = b_{N+1} \wedge \dots \wedge a_N = b_{2N} \}$

Let the user determine which of these extensions refer to an allowed population state for the universe of discourse by showing (sets of) real-life examples that match these three extensions one at a time.

The following resulting set-comparison constraints will exist:

IF

¹ For a proof of the algorithm we refer to the appendix.

² $\{R_1, \dots, R_N\} \neq \{R_{N+1}, \dots, R_{2n}\}$

```

(∃ Popstate1(UoD) [Popstate1(UoD)= EXT1(IM)] ∧
 ∃ Popstate2(UoD) [Popstate2(UoD)= EXT2(IM)] ∧
 ¬∃ Popstate3(UoD) [Popstate3(UoD)= EXT3(IM)] )
THEN
(There is a subset constraint defined from role combination
 (RN+1, ..., R2n) to role combination (R1, ..., RN))
ELSE IF
(¬∃ Popstate1(UoD) [Popstate1(UoD)= EXT1(IM)] ∧
 ∃ Popstate2(UoD) [Popstate2(UoD)= EXT2(IM)] ∧
 ∃ Popstate3(UoD) [Popstate3(UoD)= EXT3(IM)] )
THEN
(There is a subset constraint defined from role
 combination (R1, ..., RN) to role combination
 (RN+1, ..., R2n) )
ELSE IF (¬∃ Popstate1(UoD) [Popstate1(UoD)= EXT1(IM)] ∧
 ∃ Popstate2(UoD) [Popstate2(UoD)= EXT2(IM)] ∧
 ¬∃ Popstate3(UoD) [Popstate3(UoD)= EXT3(IM)] )
THEN
(There is an equality constraint defined from role
 combination (R1, ..., RN) to role combination (RN+1,
 ..., R2n) )
ELSE IF (∃ Popstate1(UoD) [Popstate1(UoD)= EXT1(IM)]
 ∧ ¬∃ Popstate2(UoD) [Popstate2(UoD)= EXT2(IM)] ∧
 ¬∃ Popstate3(UoD) [Popstate3(UoD)= EXT3(IM)] )
THEN (There is an exclusion constraint defined from
 role combination (R1, ..., RN) to role
 combination (RN+1, ..., R2n) )
ENDIF
ENDIF
ENDIF
ENDIF
END

```

In figure 3 we have given a fact-based conceptual model plus the populations that represent the extensions 1, 2 and 3 from algorithm 2. We note that the algorithm also applies to 'nested' object types and to role combinations of nested object types and entity types.

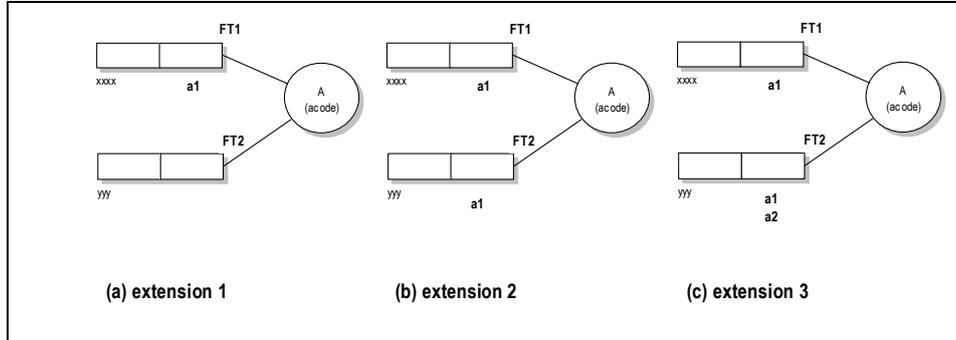


Fig 3. Graphical fact-based presentation of extensions 1, 2 and 3 from algorithm 2.

We can now summarize the algorithm as a decision-table in which a given combination of allowed existence or non-allowed existence of each of the example extensions as confirmed by the domain user in an analyst-user dialogue leads to the detection of (at most) one set-comparison constraint (see table 2). We note that for the other 4 possible outcomes of the algorithm no set-comparison constraints will be derived.

Table 2: Decision logic from the set-comparison constraint derivation algorithm 2.

	1	2	3	4
EXT1	Allowed	Not Allowed	Not Allowed	Allowed
EXT2	Allowed	Allowed	Allowed	Not Allowed
EXT3	Not Allowed	Allowed	Not Allowed	Not Allowed
Constraint type	Subset1 (FT2->FT1)	Subset2 (FT1->FT2)	equality	exclusion

The scope of Algorithm 2 is one pair of role(s) (combinations) in which in principle the same entity types (or nested object types or a combination of entity types and nested object types) are involved. In our tax collection example roles r2, r4, r5 and r7 are the roles played by the entity type *tax payer*. Hence, for each combination of two roles³ that potentially can have set-comparison constraints attached to them we must apply algorithm 2.

In this example of a conceptual schema we, therefore, need to apply algorithm 2 for the following pairs of roles: (r2, r4); (r2, r5); (r2, r7); (r4, r5); (r4, r7); (r5, r7). To illustrate the derivation algorithm we will show the application of the algorithm for one out of six pairs of roles (r2, r4) in the form of a constructed analyst-user dialogue in which 3 tangible domain examples are constructed that match the extensions 1, 2 and 3 in algorithm 2.

³ Note that the number of combinations of two roles that have to be analyzed are determined by the binomial coefficient which in this case equals: $4!/(2!*2!) = 6$.

Utopian tax collection form 2011	
Social security number: 354678	
If you will become 27 years or older in this fiscal year please fill in part 1 otherwise part 2	
<p>PART 1</p> <p>Are you married: <input checked="" type="checkbox"/> yes <input type="checkbox"/> no</p> <p>If you have children please fill in the answer to the next question.</p> <p>How many children do you have ?:</p>	<p>PART 2</p> <p>Your current status is: <input type="checkbox"/> student <input type="checkbox"/> employee <input type="checkbox"/> unemployed</p> <p>Your gross income this year was:</p>

Fig 4. Tax collection ‘real-life’ example based upon extension 1 of algorithm 2

We will consider the potential set-comparison constraint that we will derive for the roles r2 and r4. Extension 1 is supposed to be a sole instance of fact type 1: (married, 354678). The expression of this fact instance as a tangible user example is given in figure 4, the verbalization of this example is:

The civil status for tax payer 354678 is *married*

If we confront a domain user with this example and ask him/her to state whether this is an allowed example of communication or not, the answer is; yes, this is an allowed example of communication. We can now move on to the creation of a tangible example that embodies the example facts that are contained in extension 2 (as defined in algorithm 2). This constructed user example is given in figure 5.

Utopian tax collection form 2011	
Social security number: 354678	
If you will become 27 years or older in this fiscal year please fill in part 1 otherwise part 2	
<p>PART 1</p> <p>Are you married: <input checked="" type="checkbox"/> yes <input type="checkbox"/> no</p> <p>If you have children please fill in the answer to the next question.</p> <p>How many children do you have ? : 3</p>	<p>PART 2</p> <p>Your current status is: <input type="checkbox"/> student <input type="checkbox"/> employee <input type="checkbox"/> unemployed</p> <p>Your gross income this year was:</p>

Fig 5. Tax collection ‘real-life’ example based upon extension 2 of algorithm 2

Extension 2 represents an instance of fact type 1: (married, 354678) plus an instance of fact type 2 (3, 354678). The expression of these fact instances as a tangible user example is given in figure 4, the verbalization of this example is:

The civil status for tax payer 354678 is *married*
 Tax payer 354678 has 3 children

If we confront a (different) domain user with this example and ask him/her to state whether this is an allowed example of communication or not, the answer is: yes, this is an allowed example of communication. We can now verbalize the business rule that is exemplified by this acceptance of the example by the user as follows:

Br1: A taxpayer that has a marital status can have children.

Utopian tax collection form 2011	
Social security number: 354678	
If you will become 27 years or older in this fiscal year please fill in part 1 otherwise part 2	
PART 1 Are you married: <input checked="" type="checkbox"/> yes <input type="checkbox"/> no If you have children please fill in the answer to the next question. How many children do you have ?:3	PART 2 Your current status is: <input type="checkbox"/> student <input type="checkbox"/> employee <input type="checkbox"/> unemployed Your gross income this year was:

Utopian tax collection form 2011	
Social security number: 784567	
If you will become 27 years or older in this fiscal year please fill in part 1 otherwise part 2	
PART 1 Are you married: <input type="checkbox"/> yes <input type="checkbox"/> no If you have children please fill in the answer to the next question. How many children do you have ?: 3	PART 2 Your current status is: <input type="checkbox"/> student <input type="checkbox"/> employee <input type="checkbox"/> unemployed Your gross income this year was:

Fig 6. Tax collection ‘real-life’ example based upon extension 3 of algorithm 2

Extension 3 represents an instance of fact type 1: (married, 354678) plus two instances of fact type 2: (3, 354678) and (3, 784567). The expression of these fact instances as a tangible user example is given in figure 5, the verbalization of this example is:

The civil status for tax payer 354678 is *married*
 Tax payer 354678 has 3 children
 Tax payer 784567 has 3 children

If we confront a (different) domain user with this example and ask him/her to state whether this is an allowed example of communication or not, the answer is: NO, this is not an allowed example of communication. The reason that it is not an allowed example is that if the number of children are listed, a marital status must also be given (note that this is not the case for taxpayer 784567). We can now verbalize the business rule that is exemplified by this acceptance of the example by the user as follows:

Br2: A taxpayer that has children must have a marital status.

When we now apply the decision table (2) in terms of algorithm 2 we yield the following set comparison constraint that exists

There is a subset constraint defined from role combination (R_{N+1}, \dots, R_{2n}) to role combination (R_1, \dots, R_N)

Which in this situation means a subset constraint that is defined from role $r2$ to role $r4$ (see figure 7) exists. We will show the final result of applying algorithm 2 on all 6 role combinations in the following set of (additional) business rules to the fact types and uniqueness constraints in figure 2.

- Br1: A taxpayer that has a marital status can have children.*
- Br2: A taxpayer that has children must have a marital status.*
- Br3: A taxpayer that has a marital status cannot have a work status*
- Br4: A taxpayer that has a marital status cannot have a gross income*
- Br5: A taxpayer that has a gross income must have a work status*
- Br6: A taxpayer that has a work status must have a gross income*

The accompanying resulting conceptual schema including uniqueness and set comparison constraints is presented in figure 7.

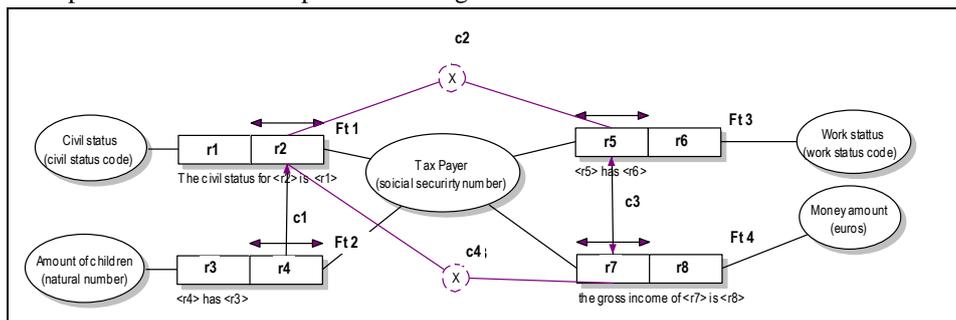


Fig 7. Conceptual schema including uniqueness- and set-comparison constraints.

We have now shown how a formal procedure will help us in extracting the domain-knowledge in the form of set-comparison constraints that can be phrased as semantic business rules by the domain user.

4 The derivation of mandatory role constraints

4.1 The mandatory role constraint to set-comparison constraint transformation

The mandatory role constraint, can be considered as a 'derived constraint' or a graphical shortcut. This means that a mandatory role constraint can always be replaced by one or more set comparison constraints, i.e. equality and subset constraints. This means that when we have applied the steps for detecting the set comparison constraints in the former section, we can map a subset of those constraints onto mandatory role constraints using the template in figure 8 and the following rules:

- Rule 1: Replace the equality constraints by two subset constraints (of different direction).
- Rule 2: For each role in which subset constraints from all other relevant roles 'end': Create an mandatory role constraint for that role. Delete the incoming subset constraints.
- Rule 3: Retransform (if necessary) the remaining subset constraints (e.g. into equality constraints).

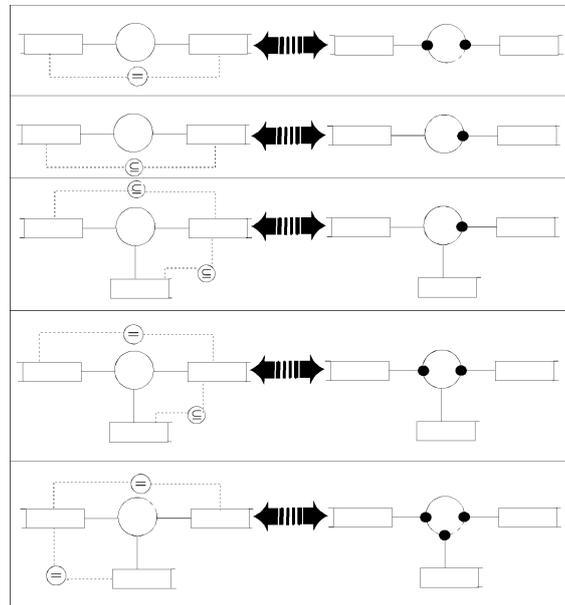


Fig 8. Subset constraint(s) versus mandatory role constraint(s)

4.2 The mandatory role constraint and set-comparison constraints and UoD integration transformations

Now we have introduced a formal modeling procedure to derive all set comparison constraints for a subject domain in an analyst-user dialogue in which the only user-input consists of accepting or rejecting (combination) of real-life examples we will investigate how we can capitalize on these analysis results as a starting point for the derivation of mandatory role constraints. If we take a ‘snapshot’ stance towards a conceptual schema, at a certain point in time, it can be shown that a mandatory role constraint in combination with the presence or absence of an entity type independence constraint [17] can be replaced by a set of set-comparison constraints. In [17] it was shown that in case a conceptual schema is adapted to cater for an extension of the Universe of Discourse (UoD), for example by integrating two conceptual schemas, that mandatory role constraints in the original conceptual schema might have to be replaced by set-comparison constraints in the integrated schema that capture the ‘semantics’ of the original mandatory role constraint. In figure 9a we have given a conceptual schema for UoD1.

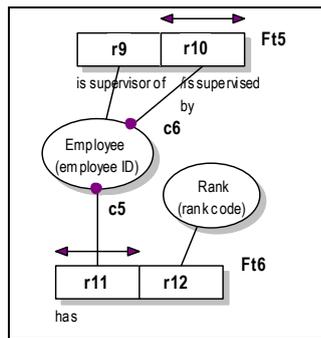


Fig 9. (a) Conceptual schema UoD1

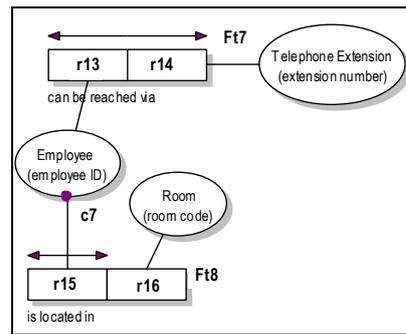


Fig 9.(b) Conceptual schema UoD2

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

*Each employee in the organization has exactly one rank.
Each employee has exactly one supervisor
Each supervisor is an employee.*

In UoD2 we have the following domain semantics:

*Each employee is located in exactly one room
For each employee that is located in a room zero, one or more (company) telephone extensions can exist.
A telephone extension can serve zero, one or more employees that are located in a room.*

In figure 9b we have presented a conceptual schema for Uod2. If we now want to integrate Uod1 and UoD2 we can merge the fact types and uniqueness constraints into one integrated conceptual schema. However, we can not do this for the mandatory role constraints, because in the integrated UoD the mandatory participation in some roles does no longer hold. We have summarized the semantics of the integrated domain (IUoD)as follows:

Each employee in the organization has exactly one rank.
Each employee has exactly one supervisor
Every supervisor is an employee.
An employee can be located in a room.

For each employee that is located in a room zero, one or more (company) telephone extensions can exist.

A telephone extension can serve zero, one or more employees that are located in a room.

In figure 10 we have given the conceptual schema for the integrated UoD (IUoD).

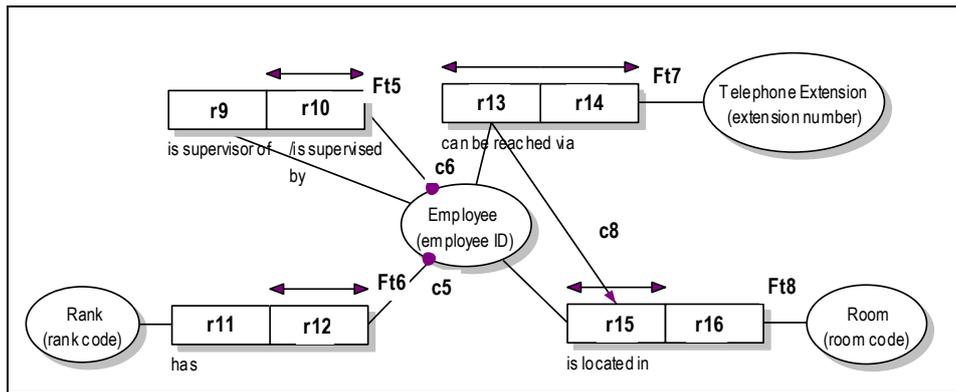


Fig 10. Conceptual schema IUoD

If we carefully analyze the conceptual schema of the integrated UoD in figure 10 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. From this we can conclude that the set-comparison constraint derivation procedure as was introduced in this paper can also be applied when two or more local schemas must be integrated to one global schema, i.e. the phenomenon of data federation [18]. By expressing the mandatory role constraints as (combinations of) set comparison constraints the process of data federation can focus on the derivation of set-comparison constraints by using algorithm 2 for the ‘new’ role combinations that govern the integrated UoD. After the additional set-comparison constraints have been derived and added to the conceptual schema, the inverse transformation can be applied in which the mandatory

role constraints that govern the federated UoD replace (some of the set-comparison) constraints.

5 Conclusion

In this paper we have shown how the fact-based modeling methodology can help a business analyst to explicitly derive all set-comparison and mandatory role constraints in an analyst-user dialogue, by exclusively confronting the domain expert with ‘real-life’ examples of communication. The specific format of these examples is defined in algorithm 2 which is a formalization of (parts of) steps 5 and 6 of the Conceptual Schema Design Procedure (CSDP). The only domain input for this procedure is the result of the acceptance or rejection of real-life user examples by the domain expert. Furthermore, we have shown how set-comparison constraints can be transformed into mandatory role constraints and the other way round, in this way algorithm 2 can also be used to derive mandatory role constraints.

References

1. Chen, P., *The Entity-Relationship Model : Toward a Unified View*. ACM Transactions on Database Systems, 1976. **1**(1): p. 9 - 36.
2. Teorey, T., D. Yang, and J. Fry, *A logical design methodology for relational databases using the extended E-R model*. ACM Computing Surveys, 1986. **18**(2): p. 197-222.
3. Hoffer, J., M. Prescott, and F. McFadden, *Modern database management*. 7th edition 2005: Pearson/Prentice-Hall.
4. Nijssen, G. *On the gross architecture for the next generation database management systems*. in *Information processing 1977*. 1977. IFIP.
5. Verheijen, G. and J. van Bekkum. *NIAM: An Information Analysis method*. in *IFIP TC-8 CRIS-I conference*. 1982. North-Holland, Amsterdam.
6. Nijssen, G. and T. Halpin, *Conceptual schema and relational database design: a fact oriented approach*1989, New-York: Prentice-hall. 337.
7. Booch, G., J. Rumbaugh, and I. Jacobson, *Unified Modeling Language User Guide*. 2nd ed. The Addison-Wesley Object Technology Series)2005: Addison-Wesley Professional. 496.
8. Halpin, T. and T. Morgan, *Information Modeling and Relational Databases; from conceptual analysis to logical design* 2nd ed2008, San-Francisco, California: Morgan-Kaufman.
9. Halpin, T., *Information Modeling and Relational Databases; from conceptual analysis to logical design*2001, San Francisco, California: Morgan Kaufmann.
10. Nijssen, M. and I. Lemmens. *Verbalization for Business Rules and Two Flavors of Verbalization for Fact Examples*. in *OTM workshops 2008*. 2008. Monterrey, Mexico: Springer Verlag.
11. Nijssen, M., I. Lemmens, and R. Mak, *Fact-Oriented Applied to Develop a Flexible Employment Benefits System*, in *ORM 2009*2009, Springer: Vilamoura, Portugal.

12. Bollen, P., *How to overcome pitfalls of (E)ER and UML in Knowledge Management education*. International Journal of Teaching and Case Studies, 2008. **1**(3): p. 200 - 223.
13. Nijssen, G., *Grondslagen van Bestuurlijke Informatiesystemen*1989, Slenaken: Nijssen Adviesbureau voor Informatica.
14. Leung, C. and G. Nijssen, *From a NIAM conceptual schema onto the optimal SQL relational database schema*. Australian Computer Journal, 1987. **19**(2): p. 69 - 75.
15. Bollen, P., *A Fact-Based Meta Model for Standardization Documents*. Lecture Notes in Computer Science, 2010. **6428**: p. 464-473.
16. Nijssen, G., *Universele Informatiekunde*1993, Beutenaken: PNA publishing.
17. Bollen, P., *A formal transformation from object-role models to UML class diagrams*, in *Seventh CAiSE/IFIP-WG8.1 international workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD '02)*, T. Halpin, K. Siau, and J. Krogstie, Editors. 2002: Toronto, Canada. p. 132-143.
18. Balsters, H. and T. Halpin, *Modeling Data Federations in ORM*, in *ORM 2007*2007, Springer: Vilamoura, Portugal.

Appendix:

Proof algorithm 2

```
WHILE still role combinations DO
IF
(  $\exists$  Popstate1(UoD) [Popstate1(UoD)= EXT1(IM)]  $\wedge$   $\exists$  Popstate2(UoD)
[Popstate2(UoD)= EXT2(IM)]  $\wedge$   $\neg \exists$  Popstate3(UoD) [Popstate3(UoD)=
EXT3(IM)] )
```

THEN

(There is a subset constraint defined from role combination (R_{N+1}, \dots, R_{2n}) to role combination (R_1, \dots, R_N))

{ Proof: Suppose no subset constraint from role combination (R_{N+1}, \dots, R_{2n}) to role combination (R_1, \dots, R_N) exists. In this case it follows that: \exists Popstate[x](UoD) [Popstate[x](UoD)= EXT_j(IM) | $\exists \alpha$ [$\alpha \in \{(a_{n+1}, \dots, a_{2n})\} \wedge \alpha \in \{(a_{n+1}, \dots, a_{2n})\}$]] implying that a population state EXT₃(IM) can be created such that \exists Popstate(UoD) [Popstate(UoD)= EXT₃(IM)]). This is a paradox. We have now proven that a subset constraint exists from role combination (R_{N+1}, \dots, R_{2n}) to role combination (R_1, \dots, R_N) □

ELSE IF

```
(  $\neg \exists$  Popstate1(UoD) [Popstate1(UoD)= EXT1(IM)]  $\wedge$ 
 $\exists$  Popstate2(UoD) [Popstate2(UoD)= EXT2(IM)]  $\wedge$   $\exists$  Popstate3(UoD)
[Popstate3(UoD)= EXT3(IM)] )
```

THEN

(There is a subset constraint defined from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}))

{ Proof: Suppose no subset constraint from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}) exists. In this case: $\exists \text{Popstate}[x](UoD) [\text{Popstate}[x](UoD) = \text{EXT}_j(\text{IM}) \mid \exists \alpha [\alpha \in \{(a_{n+1}, \dots, a_{2n})\} \wedge \alpha \notin \{(a_1, \dots, a_n)\}]$ implying that a population state $\text{EXT}_1(\text{IM})$ can be created such that $\exists \text{Popstate}(UoD) [\text{Popstate}(UoD) = \text{EXT}_1(\text{IM})]$). This is a paradox. We have now proven that a subset constraint exists from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}))}

ELSE IF $(\neg \exists \text{Popstate1}(UoD) [\text{Popstate1}(UoD) =$

$\text{EXT}_1(\text{IM})] \wedge \exists \text{Popstate2}(UoD) [\text{Popstate2}(UoD) = \text{EXT}_2(\text{IM})] \wedge$

$\neg \exists \text{Popstate3}(UoD) [\text{Popstate3}(UoD) = \text{EXT}_3(\text{IM})]$)

THEN

(There is an equality constraint defined from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}))

{ Proof: Suppose no equality constraint from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}) exists. In this case it follows that:

$\exists \text{Popstate}[x](UoD) [\text{Popstate}[x](UoD) = \text{EXT}_1(\text{IM}) \mid \exists \alpha [\alpha \in \{(a_{n+1}, \dots, a_{2n})\} \wedge \alpha \notin \{(a_1, \dots, a_n)\}]$] $\wedge \exists \text{Popstate}[y](UoD) [\text{Popstate}[y](UoD) = \text{EXT}_1(\text{IM}) \mid \exists \alpha [\alpha \notin \{(a_{n+1}, \dots, a_{2n})\} \wedge \alpha \in \{(a_1, \dots, a_n)\}]$] implying that can be created a population state $\text{EXT}_1(\text{IM})$ such that $\exists \text{Popstate}(UoD) [\text{Popstate}(UoD) = \text{EXT}_1(\text{IM})] \wedge \text{Popstate}(UoD) = \text{EXT}_3(\text{IM})]$) . This is a paradox. We have now proven that an equality constraint exists from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}))

ELSE IF $(\exists \text{Popstate1}(UoD) [\text{Popstate1}(UoD) =$

$\text{EXT}_1(\text{IM})] \wedge \neg \exists \text{Popstate2}(UoD) [\text{Popstate2}(UoD) =$

$\text{EXT}_2(\text{IM})] \wedge \neg \exists \text{Popstate3}(UoD) [\text{Popstate3}(UoD) =$

$\text{EXT}_3(\text{IM})]$)

THEN (There is an exclusion constraint defined from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}))

{ Proof: Suppose no exclusion constraint from role combination (R_1, \dots, R_N) to role combination (R_{N+1}, \dots, R_{2n}) exists. In this

