On the consistency of Cardinality Constraints in UML Modeling

F. Boufares, H. Bennaceur and A. Osmani

e-mail : {boufares, bennaceu, ao}@lipn.univ-paris13.fr
LIPN-UMR 7030 du C.N.R.S.
Université Paris 13, Institut Galilée
99, avenue J. B. Clément,
F-93430 Villetaneuse FRANCE.

Abstract. At a conceptual level data modeling consists in providing a structured form of relevant information and to accompany *structures* with *constraints* in order to capture more semantics. Cardinality constraints are among the most popular classes of constraints in database models. While each constraint class is now well understood, little is done about their interaction since possible conflicts among them may appear. The global coherence of these constraints must be considered before creating the physical corresponding database. In order to help in database design, our aim is then to propose a tool for reasoning about a set of cardinality constraints. We will treat the global coherence using mathematical programming technique. The analyses and the detection of invalid sub-schemas will be done using Fourier-Motzkin elimination.

Key Words : Conceptual modeling, UML Approach, Constraints, Cardinality constraints, Valid Conceptual schema, Linear Programming, Fourier-Motzkin Elimination.

1 Introduction

Many of systems which have to be modeled have persistent objects, which means that they can be stored in database for later retrieval. Most often, relational database, an object oriented database or a hybrid object/relational database are used for persistent storage. The UML (Booch98) (Arlow02) as well as Entity-Relationship (ER) approaches (Chen76, Smith77, Engels93, Elmasri94) are well-suited to model conceptual and logical database schemas.

Nowadays, applications become more and more complex. They involve a large amount of data coming from different sources. Designers are highly prone to make many mistakes to structure this data and to accompany data definition with constraints. For instance, conflicts among these constraints are not easy to detect. Database designers use CASE (Computer-Aided Software Engineering) tools to create a conceptual schema for their database applications.

These tools are based on UML or ER approaches. They are used interactively by database designers to create a conceptual schema for their database applications.

Many tools use ER-diagrams or UML's class diagrams to develop graphically the schema and then automatically convert it into a target database schema in the language of a specific database management system (DBMS), or in specific programming languages. *Power AMC* [] which is a Sybase product, *Oracle Designer* (Oracle00), *Objecteering* (softeam02) *and Rational Rose* (Rose02) are examples of CASE tools.

In most of them, cardinality constraints are embedded. In ER modeling, the consistency problem has been considered by several authors (Boufars02, Hartmann00, Thalheim92, Lenzerini90). Unfortunately, little is done about these constraints using UML approach. Efficient algorithms for consistency checking are still missing. Most of actual tools do not offer *intelligent consistency checking routines* to detect especially *conflicts* among the constraints considered at the same time. In this work, our aim is to stress some characteristics for a design tool. In fact, it is necessary to check the validity of a conceptual schema before the automatic translation in a target one. Consistency mistakes must be detected in the beginning of the software development life cycle. Our approach emphasizes the importance of the constraints' verifications in general and that of cardinality ones in particular.

Our aim is to stress some characteristics for a design tool. In fact, it is necessary to check the validity of a conceptual schema before the automatic translation in a target one. Consistency mistakes must be detected in the beginning of the software development life cycle. Our approach emphasizes the importance of the constraints' verifications in general and that of cardinality ones in particular.

In this work we will propose a constraint propagation algorithm based on Fourrier-Motzkin elimination (Williams76) technique to detect and to localize unconsistancy in UML diagrams. We consider binary relationships in convex domains.

This paper is organized as follows. In the second Section we briefly describe the UML approach and give an over view of constraints in database modeling. A discussion about cardinality constraint satisfaction is elaborated. In Section 3 we define the linear constraint system corresponding to a conceptual schema. The analysis of invalid schemas is developed in Section 4 and 5 using Fourier-Motzkin elimination which is a mathematical programming technique for solving linear constraint systems then we draw conclusion and indications to further research.

2 UML Modeling

UML become a standard for modelling objet-oriented systems. Except the object constraint language (OCL)(Warmer98, OMG-OCL02), its notations are diagrammatic. Constraints are essentially textual, stylised form of first order predicate logic which is part of UML standard (OMG97) intended to be used in conjunction with the UML for building consistant diagrams.

OLC gives a language to express constraint. But it still difficult to propose an unified framework to reason about constraint propagation in UML diagrams. The main problem is the hetegenous form of contraints and the abstraction level of some diagrams (some notations are informal). Several works gives propositions dealing with constraints (IBM00, Schmidt01, Jackson00, Gil99).

Conceptual modeling within UML concerns designing an application by means of *elements*. These *elements* are organized, with respect to their *properties*, into sets called *Classes*. Two or several *elements* are respectively related by means of *binary* or *n-ary relationships*. Relationships that have the same shape are organized into sets called *Classes*.

A common *property* of class is called its *attribute*. At a conceptual level, classes and attributes are designated by *names*. Roughly speaking, an *UML class diagram* of an application consists of the set of these inter related names. The graphical representation of classes is called the *UML class-diagram*. If a class participates twice or more in the same relationship type R then R is called *recursive*. Figure 1 shows a simple example of a class Diagram.

In order to capture more semantics of real world objects, the diagrams are usually accompanied with some constraint declarations. If we consider the class diagram, examples of constraints are *key constraints*(or object identifiers in UML) and *cardinality* ones (OMG-OCL02). Key constraint concerns only classes independently from any other concept, whereas cardinality constraint deals with a class and the corresponding relationship one. The key constraint declaration consists in stressing some attributes of a class C as key attributes in C. It means that there are no distinct objects in C having the same values for the key attributes. The participation of a Class C in a relationship R consists in determining a cardinality constraint. Declaring such constraint consists in adding a label to the schema which is composed of a range or a set of values.

Example 1. The running example: Management of a National Research Center)

The following class diagram describes the management of laboratories belonging to the national center of research. Each laboratory is composed of teams. Each one belongs to only one laboratory which members (researchers) compose teams. Teams in laboratories have to establish contacts with other laboratories. Each team must have at least 3 contacts. Laboratories may be not contacted (figure 1).

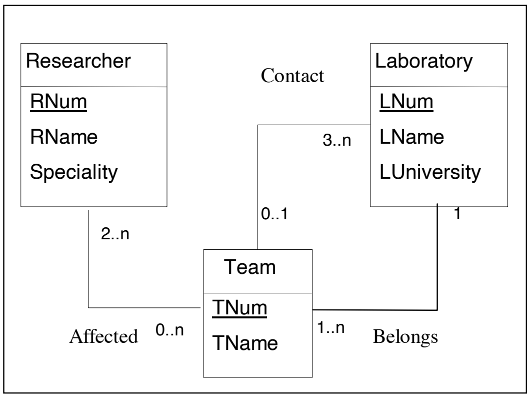


Fig. 1. UML class diagram: National Research Center Management

3 Cardinality Constraints

In many modeling situations, it is important for the designer to state how many objects may be connected across an instance of an association. This "*how many*" is called the cardinality of an association's role, and is written as an expression "*min..max*" or "*singleValue*" (when the min value is equal to the max one)[1]. If a class C is involved in a relationship R playing the role L, and the cardinality of the side of L is (min..max) then any object of Class C playing role L may participate in at least *min* and at most *max* relationships of R.

Cardinality constraints are among the most popular classes of constraints in database models. In the literature, other kinds of constraints have been proposed (Behm193, Rochfeld93, Dullea99, Hartmann00, Gil99, IBM00, Jackson00). Constraints may be written as free-form text, or more precisely defined using UML's Object Constraint Language (OCL) (Demuth99, Gabay01, Warmer98, Kleppe00, Richters99, Akerst01).

We are interested with cardinality constraints (or multiplicities contraints in UML) to be studied in the sequel (Boufarès02). In this section, we will present the cardinality constraints in UML, then we propose an algorithm to check the consistancy able to detect and to localize unconsistancies in the model in the case of the constraints with convex domains.

Let c_i $i = \overline{1..n}$ be a set of model classes $\mathcal{C}$, $s = \prec c_1, \ldots, c_l \succ$ role names of association defined by function

$$roles(as) = \prec l_1, \ldots, l_m \succ \;\; m \geq 2$$

[1] The Object group Management document: Response to the UML 2.0 OCL RfP published in June 2002 gives more details about constraints types.

where all role names must be distinct, i.e. $\forall i, j \in \{1, \dots l\} : i \neq j \Rightarrow l_i \neq l_j$

An association specifies the possible existence of links between objets of associated classes. The multiplicities or cardinalities is the number of links that an object can be part of (OMG-OCL02). Convex cardinalities constraints are the constraints of the form :

- $a..b$ where $a,\ b \in \mathcal{N}$ and $0 \leq a < b$;
- a where $a \in \mathcal{N}$;
- $a..*$ or $*$ where the symbol $*$ means greater and unknow right limit.

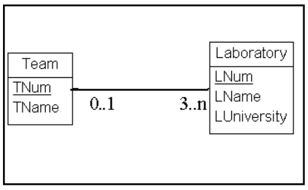
In the rest of the paper, to make easier the formulas forms, we will represent the cardinaly a by $a..a$. In the cardinality $a..*$ form, * means an unknow greater number, we will representes this cardinality by $a..b$.

4 Cardinality constraint satisfaction

Let I be an instance (a database) of a conceptual schema S. We say I satisfies the convex cardinality constraint of a relationship R over the class $\mathcal{C}$ with the role L, iff:

for every object o of class of $\mathcal{C}$, the number of objects of R connected to o must verify: $a \leq n \leq b$

For instance, figure ?? shows that for each team at least two researchers must appear in the database to satisfy the cardinality constraint $2..n$.



Teams	
TNum	TName
T1	LCR
T2	OCAD

Laboratories		
LNum	LName	Luniversity
L1	LIPN	UP13
L2	LIP6	UP6
L3	LRI	UP11
L4	LICP	UCP

Contacts	
TNum	LNum
T1	L2
T1	L3
T1	L4
T2	L ...*

Fig. 2. (a) UML schema for contact management (b) An instance for the Contacts' management database

Let e be the number of elements of E, and r the number of elements of R. In order to satisfy the cardinality constraint C, the linear constraints $a \times e \leq r \leq b \times e$ (where $e > 0$ and $r > 0$, $e,\ r \in \mathcal{N}$) must hold.
if $a = b$ then the cardinality constraint C is satisfied if the linear constraint $r - a \times e = 0$ holds.

Remark 1. Notice that the values 0 and $*$ for the minimum (a) and the maximum (b) cardinalities do not represent constraints.
A revoir si cette remarque restera

5 Conceptual Schema Linear Constraint System

Given a conceptual schema S, we associate with it a normalized linear constraint system Σ_S whose unknowns and constraints are defined as follows:

- one unknown e for each class E of S; and one unknown r for each relationship R of S (e and r represent the number of objects);
- for each cardinality constraint C denoted $a..b$, such that $a < b$ is associated the linear inequality, $a \times e - r \leq 0$ if $a > 0$ and the linear inequality $r - b \times e \leq 0$ if $b < \infty$;
- for each cardinality constraint C - denoted a- is associated the linear equality : $r - a \times e = 0$.

Notice that all the constraints of the normalized linear constraint system Σ_S are binary and can be expressed by $r - a \times e = 0$ or $a \times r - b \times e \leq 0$ where $a \geq 1$ and $b \geq 1$.

Example 2. If we consider the conceptual schema of the national research centre management project (figures 1 and 2) then the linear constraint system Σ_S will be:

$x_1 - x_4 = 0$	$2 \times x_2 - x_4 \leq 0$	$x_2 - x_5 = 0$	$x_3 - x_5 \leq 0$
$x_5 - 7 \times x_3 \leq 0$	$3 \times x_2 - x_6 = 0$	$x_6 - x_3 \times 0$	

Where:

- x_1, x_2, and x_3 are the number of objects of the classes *Researcher*, *Team* and *Laboratory*, respectively and
- x_4, x_5, and x_6 are the number of elements of the relationship *Affected*, *Belongs* and *Contact*, respectively.

The following result shows the equivalency between the validity of a conceptual schema S and the consistency of the associated linear constraint system Σ_S.

Theorem 1. *A conceptual schema S is valid, with respect to the cardinality constraints, iff the associated linear constraint system Σ_S has a solution.*

Proof : Let $E_1, E_2, \ldots, E_n$ be the classes of S and $R_1, R_2, \ldots, R_m$ be the relationships of S. S is valid, with respect to the cardinality constraints, iff there exists an instance (a database) I of S verifying all the cardinality constraints of S.
Let $C = a..b$ be a cardinality constraint of S between E_i and R_j and let $I(E_i)$ and $I(R_j)$ be the number of objects respectively of E_i and R_j in I. By definition we have:

$$a \times I(E_i) \leq I(R_j) \leq b \times I(E_i)$$

Then the linear constraint of Σ_S associated to the cardinality constraint C holds. Thus, $I(E_1), \ldots, I(E_n)$ and $I(R_1), \ldots, I(R_m)$ is a solution of the linear constraint system $Sigma_S$.
If we consider that $X = (x_1, x_2, \ldots, x_n, \ldots, x_{n+m})$ is a solution of Σ_S then, by definition, there exists an instance I of S such that $I(E_1) = x_1, \ldots, I(R_m) = x_{n+m}$ such that I verify all the cardinality constraints of S. $\diamond$

6 Solving Σ_S

In general the problem of solving integer linear constraint system is NP-complete, but as our study is restricted only to cardinality constraints then the associated integer constraint systems are homogeneous. Thus these systems can be solved as linear constraint systems with real variables in polynomial complexity ; as the variables of Σ_S are positive, the system Σ_S has a solution iff the relaxed system Σ_S with real variables has a solution too.

The method we adopt to solve the system Σ_S, is known as *Fourier-Motzkin elimination*(FM)(Williams93). This method is adequate to solve and detect inconsistencies at the same time.
All methods of solving linear programs (Simplex, Interior points, Fourier-Motzkin, etc.) may be applied to solve Σ_S (Papadimitriou82, Schrijver86, Nemhauser88, Williams93). The particularity of FM algorithm is the ability to detect and to localize the unconsistency. We have developped an extension of this algorithm to solve the system Σ_S.

Algorithm 1 *UML-FM algorithm*

Input : *system SS associated to a Conceptual-Schema S*
Output :
begin
For each constraint of the form y a x = 0 Substitute y by a x and remove this constraint ;
V ← the list of variables of SS ;
Sol ← True ;
While V is not empty and Sol do
Choose a variable x from V ;
V ← V - x ;
CP ← the list of constraints of SS where x occurs whith a positive coefficient ;
CN ← the list of constraints of SS where x occurs whith a negative coefficient ;
If CP and CN are not empty then
Transform in each constraint of CP the coefficient a of x to 1 by dividing all the coefficients of the constraint by a. ;
Transform in each constraint of CN the coefficient a of x to 1 by dividing all the coefficients of the constraint by a ;
Generate and Add to SS new inequalities by combining each constraint of CP with each constraint of CN (by adding the terms of the two constraints) ;
If there exists a new inequality having non positive coefficient then
Remove this constraint from $\frac{\Sigma}{S}$ *else Sol* ← $false$;
EndIf
EndIf
Remove from Σ_S *all the constraints in CP and in CN ;*
EndWile
If Not Sol Then SS has no solution ; S is Not valid ;
Else SS has an integer solution ; S is Valid EndIf ;
EndAlgorithm

Example 3. Let us consider the conceptual schema given in the figure 1. The following steps illustrate the instanciation of the system Σ_S using the UML-FM algorithm:

Step 0 (1.0) $x_4 - x_1 \leq 0$ (2.0) $x_2 - x_4 = 0$ (3.0) $x_5 - x_2 = 0$
(4.0) $x_3 - x_5 \leq 0$ (5.0) $3 \times x_2 - x_6 \leq 0$ (6.0) $x_6 - x_3 \leq 0$
Step 1 (1.1) $x_2 - x_1 \leq 0$ (2.1) $x_3 - x_2 \leq 0$
(3.1) $3 \times x_2 - x_6 \leq 0$ (4.1) $x_6 - x_3 \leq 0$
Step 2 (1.2) $x_2 - x_1 \leq 0$ (2.2) $x_3 - x_2 \leq 0$ (3.2) $3 \times x_2 - x_6 \leq 0$
(4.2) $x_6 - x_3 \leq 0$ (5.2)=(1+2) $x_3 - x_1 \leq 0$ (6.2=(3+2) $3 \times x_3 - x_6 \leq 0$
Step 3 Choose a variable, for example x_3 then generate new inequalities:
(1.3) $x_6 - x_3 \leq 0$ (2.3) $x_3 - x_1 \leq 0$ (3.3) $3 \times x_3 - x_6 \leq 0$
(4.3) $x_6 - x_1 \leq 0$ (5.3)=(1+2) $x_6 \leq 0 \rightarrow$ sol = false because $x_6 > 0$

The analysis of the validity of an Conceptual schema consists in checking the global coherence of all the constraints, considered at the same time. In fact, our aim is to detect conflicts among constraints, we have to study the global coherence of an Conceptual schema. This way, we can discover that the corresponding database will be partially empty.

The non existence of a solution can be explained by backtracking the resolution process then it is possible to identify the cardinality constraints which induce conflicts in the Conceptual schema.
For instance, in the running example the inequality $(2/3) \times x_6 \leq 0$ occurs when the variable x_3 is chosen. Thus the cardinality between x_6 (which indicates the relationship type name *Contact*) and x_3 (which indicates the entity type name *Laboratory*) have to be modified, else by going up the steps we have to consider the variable x_2 (which indicates the entity type name *Team*) and then the set of cardinalities that must be examined is located in the ER-sub-schema (*Laboratory- Contact- Team).* By going up the process we have to consider the variable x_5 (x_5=x_2 ; which indicates the relationship type name *Belongs*) and then the set of cardinalities that must be examined is located in the ER-sub-schema (*Laboratory- Contact- Team- Belongs).*

7 Conclusion

We have shown that possible conflicts among cardinality constraints may appear in an UML class-diagram (a conceptual schema). We proposed to associate a normalized linear constraint system SS to each conceptual schema S.
Conflicts among constraints are then detected by resolving the system SS corresponding to the conceptual schema studied using integer programming techniques. The method we adopt is known as Fourier-Motzkin elimination [Boufars2002], it allows to detect and locate at the same time invalid sub-schemas.
There are many problems that should be investigated in the future to extend

this work. It is the case of studying the extended cardinality constraints by integrating complex constraint cardinalities (disjunction of basic constraints, ranges and enumerated sets).

A CASE tool based on the ER-model is realized. This tool, developed with C++, generates SQL2 orders for an input conceptual Conceptual schema. The analysis of the validity of an Conceptual schema consists in checking, on the one hand, the *syntax* according to the meta-model we have defined in [Boufarès2001, 2002] and on the other hand, the *semantic* aspects in which we stress the importance of the global constraints' verifications in general and that of cardinality ones in particular.

Conflicts among constraints are detected by resolving the inequality system corresponding to the Conceptual schema studied using integer programming techniques. The method we adopt is known as Fourier-Motzkin elimination [Williams1993], it allows to detect and locate at the same time invalid sub-schemas.

There are many problems that should be investigated in the future to extend this work. It is the case of studying the extended cardinality constraints by integrating complex constraint cardinalities (disjunction of basic constraints, ranges and enumerated sets).
A new unit will be developed which allows schema transformation and integration between an ER-conceptual schema and an UML class diagram[Gabay2001]

8 Bibliography

[Arlow2002] Jim Arlow, Ila Neustadt UML and the Unified Process: Practical Object-Oriented Analysis and Design Addison Wesley Professional; ISBN: 0201770601; 1st edition (January 16, 2002)
[Buooch1998] G. Booch, J. Rumbaugh and I. Jacobson The Unified Modeling Language User Guide, Addison-Wesley ISBN 0-201-57168-4
[Chen1976] P. P. Chen, "The Entity-Relationship Model -Towards a Unified View of Data", volume 1, ACM Transactions On Database Systems, March 1976, pages 9-36.
[Smith1977] J. M. Smith and D. C. P. Smith, Database abstractions: Aggregation and Generalization, ACM Transactions On Database Systems" , volume 2, N°2, June1977, pages105-133.
[Thalheim1989] B. Thalheim, "The Higher-Order Entity-Relationship Model and (DB)2", MFDBS'89 : 2nd Symposium on Mathematical Fundamentals of Database Systems, LNCS 364, Visegrad, Hungary, June 26-301989, pages 382-397.
[Lenzerini1990] M. Lenzerini and P. Nobili, "On the satisfiability of dependency constraints in Entity-Relationship schemata", IS' 90, Information Systems, volume 15, N° 4, 1990, pages 453-461.
[Thalheim1992] B. Thalheim, "Fundamentals of Cardinality Constraints", ER'92:

Entity-Relationship Approach, 11th International Conference on the Entity-Relationship Approach, LNCS 645, Karlsruhe, Germany, October 7-9 1992, pages 7-23.
[Behm1993] J. B. Behm and T. J. Teorey, "Relative Constraints in ER Data Models", ER'93 : Entity-Relationship Approach, 12th International Conference on the Entity-Relationship Approach, LNCS 823, Arlington, Texas, USA, December1993 , pages 46-59.
[Engels1993] G. Engels, M. Gogolla, U. Hohenstein, K. Hulsmann, P. Lohr-Richter, G. Saake and H. D. Ehrich, "Conceptual modelling of database applications using an extended ER model", Data and Knowledge Engineering, volume 9, 1993, pages 157-204.
[Rochfeld1993] A. Rochfeld and P. Negros, "Relationship of relationships and other inter-relationship links in E-R model", Data and Knowledge Engineering, volume 9 , 1993, pages 205-221.
[Elmasri1994] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems", The Benjamin Cummings Publishing Company, Inc., Second Edition 1994 .
[Rauh96] O. Rauh and E. Stickel, "Standard Transformations for the Normalization of ER Schemata", IS'96, Information Systems, volume 21, N° 2, 1996, pages 187-208.
[Hartmann1998] S. Hartmann, "On the Consistency of Int-cardinality Constraints", ER'98 : Conceptual Modeling, 17th International Conference on Conceptual Modeling, LNCS 1507, Singapore, November 1998, pages 150-163.
[Demu99] B. Demuth and H. Hussmann, "Using UML/OCL Constraints for Relational Database Design", UML'99 : The Unified Modeling Language, LNCS 1723, Fort Collins, CO, USA, October 28-30, 1999, Proceedings, pages 598-613.
[Dullea99] J. Dullea and Il-Yeol Song, "A Taxonomy of Recursive Relationships and Their Structural Validity in ER Modeling", ER'99, Conceptual Modeling, 18th International Conference on Conceptual Modeling, LNCS 1728", Paris, France, November 1999 , pages 384-398.
[Soutou99] Ch. Soutou, "Objet-relationnel sous oracle8, Modélisation avec UML", Eyrolles, Paris 1999.
[Lellahi00] S.K. Lellahi and F. Boufarès, "Meta-Model and loose Semantics for ER-Model", Rapport Technique LIPN N° 2000-3 , mai 2000.
[Muller00] UML..................
[Hartmann2000] S. Hartmann, "On Interactions of Cardinality Constraints, Key, and Functional Dependencies", FoIKS'2000 : Foundations of Information and Knowledge Systems, First International Symposium, LNCS 1762, Burg, Germany, February 2000, pages 136-155.
[Oracle00] Oracle Technology Network : http://otn.oracle.com
[Boufarès01] F. Boufarès and N. Kraïem, " A new tool to analyze ER-schemas ", The second Asia-Pacific Conference on Quality Software (APAQS 2001), Hong Kong, China, December 10-11, 2001. pp 302-307
[Christensen01] H. L. Christensen & al. "Operations for Conceptual Schema Manipulation Definitions and Semantics", ADBIS 2001, Advances in Databases

and Information Systems, 5^{th} East European Conference, Vilnius, Lithuania, September 2001, LNCS 2151, Proceedings.

[Gabay01] J. Gabay, "Merise et UML pour la modélisation des systèmes d'information", Dunod, Paris 2001.

[Gil99] J. Gil, J. Howse, and S. Kent Constraint Diagrams: A Step Beyond UML In Proceedings of TOOLS USA'99. IEEE Computer Society Press, December 1999.

[IBM00] IBM. Object constraint Language. http://www-4.ibm.com/software/ad/standards/ocl.html, 2000

[Jackson00] D. Jackson, I.Schechter and I.Shyakhter Alcoa: the Alloy constraint analyser. In Proceedings of International Conference on Software Engeneering, Limerick.

[Soutou02] Ch. Soutou, "De UML à SQL, Conception de bases de données", Eyrolles, Paris, Juin 2002.

[Boufarès02] F. Boufarès, H. Bennaceur, A. Abdellatif and F. Sahli " Oracle Designer and the Global Coherence of a Conceptual Data Model ", 9^{th} European Conference on IT Evaluation (ECITE'2002), Paris, France, July 15-16, 2002.

[Papadimitriou82] C. H. Papadimitriou and K. Steiglitz, "Combinatorial Optimization", Prentice-Hall, Englewood Cliffs, N. J. 1982 ; pp 129-133.

[Schrijver86] A. Schriver "Theory of linear and integer programming", Edition Wiley.

[Nemhauser88] G. L. Nemhauser & L. A. Wolsey, "Integer and Combinatorial Optimization" John Wiley and Sons, USA, 1988.

[OMG1997] UML 1.1 Specification OMG document ad970802-ad970809

[OMG-OCL02] UML 2.0 Specification OMG document ad02-05-09

[Warmer98] J. Warmer and A. Kleppe The Object Constraint Language: Pricise Modeling with UML, Addison-Wesley, 1998 [Williams1976] H. Williams. Fourier-Motzkin elimination extension to integer programming problems. Journal of Combinatorial Theory, 21:118–123, 1976

[Williams93] H. P. Williams, "Model Solving in Mathematical Programming" John Wiley and Sons Ltd, England, 1993.