Innovative information systems modeling and development with FCO-IM

Guido Bakema (guido.bakema@han.nl)
Jan Pieter Zwart (janpieter.zwart@han.nl)
Research and Competence group Data Architectures & Metadata Management
Academy of Communication and Information Technology
HAN University, The Netherlands
June 2006

Note by the authors: the article is strongly based on the first chapter of the FCO-IM book [9]

Abstract

This article presents the context in which the innovative information modeling method Fully Communication Oriented Information Modeling (FCO-IM) should be placed. First is defined what is meant by the term information system. Next is discussed the process of information systems development and how FCO-IM fits in it. Finally the basic principles of FCO-IM are set out.

1 Information systems and information systems development

Information plays a dominant role in society nowadays. The competitive position of companies and the proper functioning of non-profit institutes depend strongly on the ability to provide management, employees, customers and government in time with the information they require. Whereas but a few decades ago all information was processed by hand, today information handling and exchange is automated to a very large extent. This is known as automation of information services. Commercial companies took the lead here, multinationals and large businesses (big production companies and banking and insurance companies, …) first. With the advent of the personal computer small and medium-sized business companies followed. Non-profit institutes (civil offices, hospitals, universities, …) initially lagged behind in this trend, but in the meantime most non-profit institutes have automated their information systems to a considerable degree.

At first this automation was done by computer departments in large business companies. But it was not long before an entirely new branch of industry consisting of automation experts developed, providing goods and services for the purpose of automating the information processing of their customers. This branch consists of computer system suppliers, software houses, system houses, consultancy bureaus and the like. In all industrialized (or better: automated) countries this branch of industry accounts for a considerable percentage of the gross national product. Either standard products are used as ready-to-wear solutions to automation problems, or software and hardware are tailored to the specific needs. In the latter case the communication processes in the organization are charted (or even redefined), the information that is being exchanged by these communication processes is analyzed, and complete automated information systems are designed, built and implemented. Another possibility is that support is provided to organizations to perform the development of their desired information systems, or the improvement of their existing information systems, themselves.

To make matters more concrete we give a few examples of information systems:

- invoice, order and stockkeeping systems for production companies;
- information systems for financial transactions (transfers) in banks;
- information systems for registering insurance policies and dispatching claims;

- booking systems for travel agencies and flight reservation systems for airlines;
- information systems for processing orders in mail-order firms;
- systems for registering inhabitants and their relevant data at civil administrations;
- student administration and monitoring (mark recording) systems in universities;
- systems for patient administration and all relevant data about them in hospitals.

An ‘information system’ can be defined as a subsystem of an organization, with the purpose to support the organization as efficiently as possible with respect to information processing. Such a subsystem in principle also includes the users of the information system in a narrow sense (those who record information in, maintain the information in and withdraw information from the information system) and in a wider sense (those who supply or consume that information). Aside: suppliers and consumers of information need not be human beings: they may be machines as well, for example machines in automated production processes (in that case those mechanical users in the wider sense are usually - but not necessarily - also users in the narrow sense). However, the term ‘information system’ is mostly used in a more restricted sense, namely without including the users. This was already manifest in the second sentense in this paragraph, where we wrote ‘users of the information system’. Obviously we should not be overly consistent on this point. The exact meaning will be clear from the context.

Initially, in the process of automation of information services the emphasis was strongly on the automation (computerization) side of the matter: “We want to automate our administration of orders, invoices and stock.” By now, automation by computer systems is taken for granted everywhere and computers are able to communicate with each other without too grave technical difficulties or security problems. In addition, relational database management systems (RDBMS’s) and front-end tools such as form and report generators (for Windows and Web) enable automation experts to meet the requirements on the automation side of affairs without too much trouble. Fully automated metadata driven *application generators* [15][21], cube generators [18] and sophisticated OLAP tools [19] are available nowadays. So, presently there is an increasing interest in the real informatization aspects:

1. Which *business processes* do we need for good management and which information processes would support these business processes best? (*business process modeling*).
2. Which *information flows* are involved and what information items play a role in these flows ? (*information analysis*).
3. What is the *structure of this information* and what are the *constraints*? (*information modeling*).
4. Which *derived, probably aggregated, information* would provide our company business intelligence? (*data warehousing*).

Business consultants try to answer the first question, whereas information analysts and data-architects try to answer the others. This illustrates a shift in emphasis from information systems development from a technical automation point of view towards an emphasis on the the information flow and the information itself. Nowadays there is more emphasis on these questions that are the hart of the matter. Allthough the three letter buzzwords are still all over the place, this is the direction it goes!

Although the demand for technical and automation experts (network specialists, application programmers, risk and security experts, …) is still increasing, nowadays information experts (information analysts, data architects, …) are wanted even more. In the highly industrialized/automated countries, after a decline during recent 5 years, the number of qualified information experts graduating from universities or from commercial trainings is again insufficient to fill the need on the job market and analytical work can not be outsourced as easy as application programming and the like. Not all organizations realize this in full extend yet. Also and also in university education and commercial trainings there is still an overemphasis on technical aspects and application programming skills, compared to analytical competences and innovative integrated tool support [13][14][15][16][17][19][21]. In other words: there is a hidden (educational and practical) demand for such competences that is expected to become more and more manifest.

2 Fact oriented information systems development

We can view information systems from three different *perspectives*:

1. The *information perspective*, which concerns the *information* that is of interest in the communication processes that are to be supported by the information system. The analysis of this relevant information determines to a very large extent the structure and the integrity aspects of a database design;
2. The *process perspective*, which concerns the *processes* that act on this information (input, output, maintenance and derivation processes). Which processes should the information system offer and what exactly should they do?
3. The *behavioral perspective*, concerning the events (such as choosing a menu option) that trigger processes to start. This perspective is especially important in the case of real-time systems.

An *information systems development method (ISDM)* is a set of coherent techniques and procedures that can be deployed in the development of information systems. ISDM's can be distinguished, among other things, by the relative emphasis they place upon the three different perspectives.

Today the insight is predominant that in information systems development the main emphasis is on the information oriented perspective (data oriented information systems development). In the first place, data structures are generally more stable than processes or triggers: information (meaning types of facts here) taking part in communication processes is usually less prone to changes than the way in which this information is processed. In the second place, the process oriented and behavior oriented perspectives can only be specified precisely in terms of the information that is to be processed. Finally, desired changes in the process oriented and behavior oriented perspectives can generally be carried out much easier with modern tools than changes in the information oriented perspective. Therefore, the problem: how to realize changes in the database schema of an operational database without too much trouble, is still a research issue nowadays (known as the *delta problem*). All this underlines the importance of meticulous information modeling.

We defined an information system as a subsystem of an organization with the purpose to support the organization as efficiently as possible with respect to information processes. The shift in emphasis towards full emphasis on the information itself is very manifest in the following extended definition of an information system:

> An information system is a subsystem of an organization for supporting the organization as efficiently as possible in its communication processes, i.e.:
> - *record the facts* involved in the communication;
> - *maintain these facts*;
> - *derive other relevant facts* from these facts;
> - *retrieve these facts* when desired;
> - *present the stored and derived facts* as desired.

This definition highlights the main focus of the innovative fact oriented approach: concentrate on the *facts* that are relevant to the organization concerned. Let's zoom in on this fact communicating way of looking at information systems modeling and development a bit further. Information experts (information analysts, modelers, data-architects, …) must be able to determine systematically the information requirements of domain experts (administrators, managers, technicians, etc.) and lay them down in a *conceptual fact based information model*. Furthermore, if required, they must be able to simply translate this conceptual information model into a *physical data model*: nowadays in general a highly normalized or dimensional data model (*relational, cube, …)* that can be implemented on a physical relational platform (RDBMS, Cube processor, …). The word 'simply' is used deliberately, because such translation (conversion) towards relational and even *dimensional models* is in high extend automated nowadays [13][14][16][17], as is relational application development (for Windows and Web) [15][21]. Why do we speak of translating towards a relational data model? Because the bulk of all currently developed information systems is (and for coming years will be) implemented on relational platforms! And if this is not the case, then the relational platform is in 99% of cases used as intermediate platform (think of *multidimensional cubes*, generated from *relational stars* or *snowflakes* [16][18][19]).

This section ends with a remark with respect to the popular distinction between *data* and *information*: in all communication processes we are dealing with facts that mean something to the participants in the communication, in a context they collectively share and understand. In addition to this aspect of meaning, facts also have an aspect of form, manifest in what we will call: *representations of facts* (namely as spoken words, or in the form of written sentenses, tables, graphs, schema's and the like, recorded on paper, visible on viewing screens or stored in digital form on data carriers). By *information* we will mean this duality of meaning and form. When just the aspect of form is meant - that is when only the representations of the facts are considered - the term *data* is generally used. But when people use the terms *data analysis* and *information analysis* they always mean the same.

3 Information systems development versus software engineering

The examples of automated information systems, presented in section 1, have something in common: they are all examples of information systems that have been automated for the very reason that huge amounts of information have to be processed. In such a situation automation is bound to pay off. It was certainly no coincidence that the automation of information processing started with automating administrative processes and the bulk of information systems design and development is still and will probably always be related to such so-called *data intensive fields of application*. We mean by this, that the amount of information is vast, not only at the *instance level* (a lot of facts), but especially at the *type level* (many different *sorts* of facts: *fact types*). This is usually the case in *administrative information systems* and *management information systems* (datawarehouses, decision support systems, …). However, *technical information systems* also often concern data intensive fields of application. Consider for example the large amounts of data of various sorts required in running modern, almost completely automated, power stations.

It is especially in such data intensive areas of application that *Fact Oriented Information Modeling (FOM)* methods, like *Natural Language based Information Modeling (NIAM)* [2a][6], *Object Role Modeling (ORM)* [2b] and the newest variant: *Fully Communication Oriented Information Modeling (FCO-IM)* [3][4][5][7][8][9][11], can be applied fruitfully. Of course, FCO-IM can be used in fields of application that have little data intensity as well, but this powerful method of information modeling has proved to be pre-eminently useful where there is a large number of fact types. Such areas of application are usually (but not always) of little algorithmic complexity. The reverse (algorithmically complex systems often have simple information structures) is also true, but to a much lesser extent. As a consequence of this phenomenon, the two different disciplines of *software engineering* and *information systems development* are seldom applied in an integrated way (see figure 1):

- *Software engineering* concentrates on the analysis and design of complex algorithms, which may or may not be data intensive at the instance level, but in general have little data intensity at the type level (do not concern many fact types). These data structures, together with the algorithms that are defined on them, are nowadays usually implemented using object oriented (OO) techniques.
- *Information systems development* is primarely concerned with developing information systems with data structures that are intensive at the type level (many fact types), but which are often simple from an algorithmic point of view. Such information systems are excellently suitable for relational implementation and data manipulation with SQL.

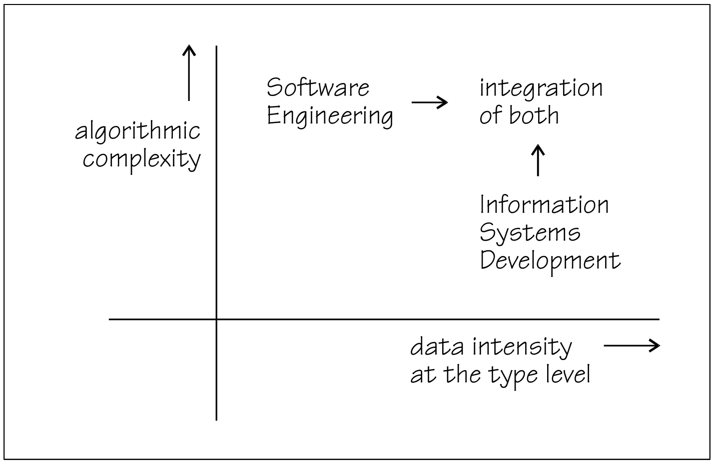

Figure 1:Information Systems Development versus Software Engineering

In fields of application with both algorithmic complexity and data intensity, of course both disciplines are deployed. In such a case relational interfaces are frequently developed, which feed OO data structures, and vice versa. The algorithmic conversion of an FCO-IM information model to OO schemas (Class diagrams) is not more difficult than to relational schemas and has recently been implemented as well [14][16][17].

4 FCO-IM placed in the information systems development life cycle

FCO-IM is about *information analysis* and *information modeling*. Let's clarify these terms a more precise and place FCO-IM in the context of the complete information system life cycle. For this an idealized linear process for designing, building and relationally implementing a new automated information system will do. Although this is an oversimplification (in practice, the process is less straightforward and partly cyclic in nature), as well as a limitation (think of a re-engineering project, or designing a datawarehouse that can be fed from several existing source systems), for the moment it provides a sufficient basis for our line of thought.

The interpretation of the term information analysis varies. Some people use it to designate the activity of roughly charting information flows in an organization. Others understand it as giving an overall specification of the information sets that make up those flows. Still others count both activities in. We would rather situate these activities in the *definition phase*, which should yield what is called a *global functional model* of the nascent information system. Such a global functional model specifies input and output functions of the information system globally in terms of the accompanying input and output information sets. The main objective in the definition phase, then, is in our view not so much a detailed analysis, but rather a summary overview of information flows and especially *information sets*. For this purpose various diagramming techniques are being used in practice, such as *data flow diagramming* or even better: *document flow diagramming*.

In most accepted phasings of the *information system life cycle* (see figure 2 for a commonly used phasing with *milestone products*), this definition phase follows the *planning phase*, in which a project plan is made for developing the information system. Before that, there may have been an *orientation phase* (or *information planning phase* or *feasability study*, although these terms are no precise synonyms), in which is investigated and decided whether or not information systems are to be developed, and if so which ones, and with what priority and in what financial and organizational context: feasibility study report.

Phases	Milestone products
Orientation	Feasibility study
Planning	Project plan
Definition	Global functional model
Analysis	Conceptual information model
Design	Technical IS design report
Building (and test)	Information system
Introduction	User guide + Introduction plan
Use and Maintenance	Additional documentation

Figure 2: ISD phasing and milestone products

Information analysis takes place in the *analysis phase*, which follows the definition phase. First, the analysts, together with domain experts, determine the relevant information sets much more precisely than in the definition phase (exactly which kinds of facts are involved?).

Next, these information sets are subjected to an analysis process yielding an *FCO-IM information model*. This latter analysis process, in the restricted sense of drawing up an information model, is also called *information modeling*. The activity of the precise determination of the relevant information sets that provide a suitable starting point for the strict information modeling, should certainly not be considered to be less important or easier to carry out than the strict information modeling itself. On the contrary, it is in fact a very important and difficult first step that is best learned in a larger practical project (a case study) than in a cursory manner. However, not until one has acquired sufficient skills in strict information modeling in smaller cases, one has reached the stage to apply this in larger cases drawn from practice. This stresses the difficulty of determining information flows and information sets in detail.

In the FCO-IM course book [9] and the accompanying exercise and case book with [12] is treated in detail how to draw up *FCO-IM information models* (also called *information grammars*) and how subsequently to derive a relational logical database schema from such information grammar. This derivation - maybe via an intermediate *Entity-Relationship model* (more familiar, use script generators from Entity-Relationship tools, …) - takes place in the *logical design* part of the design phase of the life cycle. After that follows the *physical* part of the design phase (also called *technical design*) that describes how the database schema will be implemented on the chosen platform and for the applications. This is strongly dependant from the chosen implementation platform (hardware, software) and how applications are to be developed (maybe generated). See figure 3.

Phase	Focus on	Aims
Conceptual analysis	Domain area	User examples, FCO-IM model
Logical design	Logical Data Model	Entity-Relationship model and/or Logical Relational Schema
Physical design	Implementation Platform	Physical Database Schema Application design

Figure 3: Analysis and design phases, focus and aims

After the analysis and design phases follows the *building (and test) phase*, that provides the information system and developed applications provide the automated information system (in the strict sense). Then at last the important *introduction phase* (the introduction plan also comprises the schooling of end users), and finally the *use and maintenance phase*, in which it is vital to document every shortcoming and implemented change or addition well.

5 Basic principles of FCO-IM

Fully Communication Oriented Information Modeling (FCO-IM) is founded on a number of basic principles, subdivided into modeling principles and methodical principles.

The FCO-IM modeling principles:

1. *100 % conceptuality*: In FCO-IM all conceptual aspects of the Universe of Discourse (UoD) that the information system should support must be modeled, and nothing but those aspects.
2. *Fully communication orientation*: FCO-IM does not aim to model the UoD itself, but the relevant communication about the UoD.
3. *Validation*: Domain experts must be able to validate the correctness of the modeling of their communication in an FCO-IM information grammar.
4. *Genericity*: It must be possible to represent both FCO-IM information grammars and relational schemas (and intermediate forms, like Entity-Relationship diagrams), using the same FCO-IM diagramming technique.
5. *Preserving hard and soft semantics*: FCO-IM must be able to model the soft semantics as well, and preserve them completely as a supplement to relational database schemas.

Ad 1. The first principle has been adopted as a standard starting point for information modeling in the international scientific literature [1]. It means that implementation elements (for example elements having to do with the technical realization) do not belong to the conceptual information model. Neither does the way the data is represented externally (on screens, in reports etc.). However, *all* conceptual aspects (i.e. aspects necessary for understanding) of the Universe of Discourse (i.e. the Object System, in short: 'reality' at hand) must be included explicitly in the conceptual information model. FCO-IM completely meets this requirement of 100 % conceptuality - even recursive identification is covered [20] -, although a few subtle modeling constructs, like the FCO-IM set type [6][7], are not covered in the course book, because they are mainly of theoretical interest and better avoided in practice. Other modeling constructs, like so-called disconnected and overlapping object type expressions, although considered of practical importance by some high level FCO-IM information modelers, are not yet treated in the course book, because they are more recent research results [22] and as a consequence not covered by the FCO-IM modeling tool CaseTalk yet.

Ad 2. It is not possible to record actual objects from reality in information systems (nor properties of such objects or relationships between such objects). We can only register alphanumerical (or maybe other) representations of relevant facts about objects from reality. After all information systems are ment to support the organizations communication processes. Therefore FCO-IM is based on analyzing the representations of those facts (in whatever form they are available) than reality itself (or attributes or relationships between objects or classes of objects in reality). The communication is fully analyzes instead. This is not a novel insight either, as it was previously formulated as foundation for NIAM (Natural language Information Analysis Method) [2a], in which FCO-IM has its roots. This insight led to the important methodological principle of *verbalizing representative examples of facts* that appear in *concrete examples of information sets* (*data use cases*) used in the communication processes, thus obtaining *declarative fact stating sentences* in natural language.

In FCO-IM, these natural language sentences are not only used as the starting point to arrive at a conceptual information model, but they are also completely and in a redundancy free way to be stored in the FCO-IM repository. Therefore: *fully communication oriented.*

Ad 3. Figure 3 shows schematically the perspectives of domain experts (their concrete reality and verbalizations of representative facts that can be part of the communication about this reality) and of information analysts (the same verbalizations and the FCO-IM information grammar he/she is to draw up). The dialogue between information analyst and domain experts (interviews of the latter by the analyst to arrive at an FCO-IM information grammar) takes place at the middle level: that of exemplary verbalizations. The analyst also gains some insight into the way domain experts view their reality during these interviews.

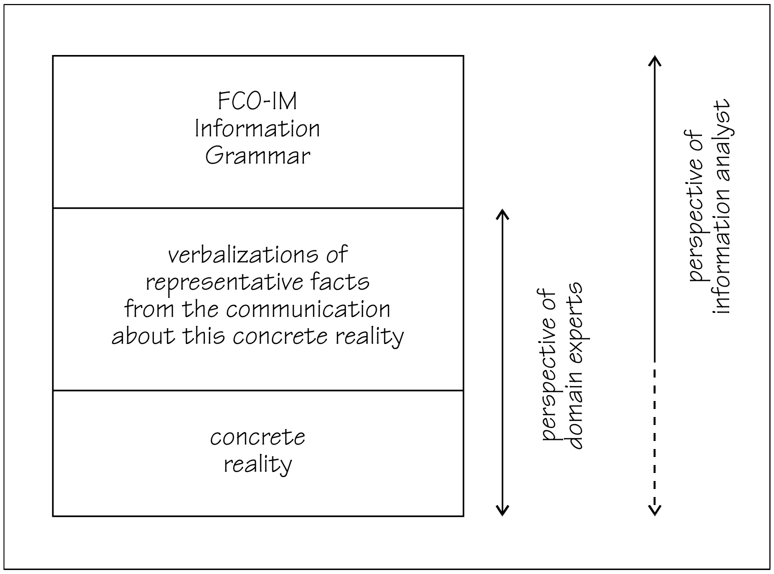

Figure 3: perspectives in FCO-IM

During and after the modeling process, analysts must be able to regenerate sentence representations of exemplary facts that are modeled and present these - together with basic constraints expressing uniqueness, mandatoryness, value restrictions and so on - to the domain experts. Only if they approve these sentences, i.e. if they confirm that they did (or could) express their relevant information in this way, then the analyst did a good job, otherwise he/she did not. In FCO-IM such fact stating sentences plus constraint formulations can be regenerated by using a simple substitution algorithm. This algorithm is implemented in the FCO-IM tool CaseTalk [13] in such a way that it can be activated at any time during the modeling process. As a consequence, the domain experts can *validate* the (intermediate) results at any desired moment.

Ad 4. This principle implies that an FCO-IM information grammar diagram can be transformed step by step into a redundancy free relational schema (even a relational schema in guaranteed Boyce Codd Normal Form (BCNF)), which can subsequently be implemented in an RDBMS. This transformation process has been automated as well and, using the FCO-IM modeling tool CaseTalk, can be followed graphically. In fact, the transformation is carried out in the repository (itself a relational database) of the FCO-IM modelingtool CaseTalk, based on stepwise changes in the population of this repository. The graphical representation is nothing more or less than an external manifestation of the contents of this generic repository, running synchronously along with its changes.

A very valuable aspect of the transformation process is the fact that an intermediate result of the transformation of an FCO-IM information grammar towards a relational schema is an FCO-IM information grammar that is equivalent with a BCNF Entity-Relationship model (after all an Entity-Relationship model is nothing else than a relational schema that is made redundancy free at the type level).

Ad 5. This one may be considered as derivable from the others, because the first four principles together enable us to derive a redundancy free relational schema with corresponding integrity rules (*not null* constraints, primary keys, foreign keys, and the like: the *hard semantics* of the relational database: structure and integrity aspects) from an FCO-IM information grammar in a fully transparent way.

Moreover, we can completely take verbalizations of the recorded data in natural language (the *soft semantics* of the database: user defined naming and predicates: verbalizations) along in this process.

Especially this last point is a unique characteristic of FCO-IM and the FCO-IM modeling tool CaseTalk, in which this transformation to a relational schema with corresponding soft semantics is implemented.

We mentioned earlier that FCO-IM stands in the NIAM tradition. This is not only true with respect to its modeling constructs and diagramming technique (way of modeling), but also with respect to its operational procedure (way of working), which we can summarize in four basic methodical principles:

The four basic methodical principles of FCO-IM are:

1. ***Collect/ construct example documents***: Collect or construct example information sets, i.e. sets of significant fact (i.e. covering all fact types at least once) facts used in the communication.
2. ***Interview in own jargon***: Domain experts must be interviewed in their own jargon.
3. ***Verbalize example facts***: Domain experts are required to supply verbalizations, in natural language, of representative examples of facts that are of interest in their communication.
4. ***Decide at the instance level***: Modeling decisions must be based on concrete (i.e. instance level) examples.
5. ***Work methodically***: The whole modeling process must be based in a stepwise instruction set (do this, than do that, …).

In the FCO-IM course book [9] and the accompanying exercise and cases book [12] the operational procedure according to these basic principles is presented and illustrated comprehensively. The accompanying instructions are systematically elaborated, until the complete FCO-IM procedure has been treated. This is of course always done starting from concrete examples (i.e. small cases).

References and literature

In chronological order:

[1] Concepts and Terminology for the Conceptual Schema and the Information Base, J.J. van Griethuysen, Publ. Nr. ISO/TC97/SC5-N695, 1982.

[2a] Conceptual Schema and Relational Database Design (a fact oriented approach), G.M. Nijssen, T.A. Halpin, Prentice-Hall, Sydney Australia, 1989.

[2b] Conceptual Schema and Relational Database Design (a fact oriented approach), revised edition, T.A. Halpin, Prentice-Hall, Sydney Australia, 1995.

[3] Objects are in fact facts, Harm van der Lek, Guido Bakema, internal publication BSO (Atos Origin) and HAN University, Den Haag/Nijmegen The Netherlands, 1991.

[4] The unification of object types and fact types, H. van der Lek G.P. Bakema, J.P.C. Zwart, NIAM-ISDM international conference 1993, Utrecht The Netherlands, 1993.

[5] On the structure of an Information Grammar, H. van der Lek, NIAM-ISDM international conference 1993, Utrecht The Netherlands, 1993.

[6] Information Modelling in data intensive domains, A.H.M. ter Hofstede, master thesis Radboud University, Nijmegen The Netherlands, 1993.

[7] Fully Communication Oriented NIAM, G.P. Bakema, J.P.C. Zwart, H. van der Lek, NIAM-ISDM international conference, Albuquerque USA, 1994.

[8] Fully Communication Oriented Information Modeling and Relational Database Design, Guido Bakema, lecture notes HAN University and Radboud University, Nijmegen The Netherlands, 1994.

[9] Fully Communication Oriented Information Modeling, Guido Bakema, Jan Pieter Zwart, Harm van der Lek, Dutch course book (1996) translated in English (pdf-file), HAN University, Arnhem/Nijmegen the Netherlands 1999.

[10] FCO-IM Casetool, Ascaris Software, first fully functional FCO-IM modeling tool with integrated user guide, Amsterdam 1996/1997.

[11] Where Category Theory meets FCO-IM, M. Jansen, E. ter Horst, master thesis Radboud University, Nijmegen The Netherlands, 1997.

[12] Fully Communication Oriented Information Modeling: exercises and cases, Marco Engelbart, Rob Arntz, Jorg Janssen, Dutch course book (2000) translated in English (pdf-file), Arnhem/Nijmegen the Netherlands, 2001.
[13] *CaseTalk, state of the art FCO-IM modeling tool with integrated user guide, BCP Software (www.CaseTalk.com), Utrecht The Netherlands, 2002.*
[14] StarBridge, getting from FCO-IM to Dimensional Model, Elton Manoku, master thesis HAN University, Arnhem/Nijmegen The Netherlands, 2002.
[15] CASE Tool Interoperability, Chris Scholten, master thesis HAN University, Arnhem/Nijmegen the Netherlands, 2003.
[16] Integrated Tool Support for Datawarehouse Design, Journal of Conceptual Modeling, Elton Manoku, Guido Bakema, Internet Magazine, 2003.
[17] FCO-IM Bridge, integrated (ER Bridge, UML Bridge, Star Bridge) model-to-model transformation tool + user guides, HAN University and BCP Software, Arnhem/Utrecht The Netherlands, 2004.
[18] Aspects of aggegates in Datawarehouses and Multidimensional Databases', Abdallah Eteleeb, master thesis HAN University, Arnhem/Nijmegen the Netherlands 2004.
[19] Management Information System for ArboTotaal, Entela Shehu, master thesis HAN University, Arnhem/Nijmegen the Netherlands, 2004.
[20] Development of a Database Server for Quaestor Project Data', Azizah F.N., master thesis HAN University, Arnhem/Nijmegen the Netherlands, 2005.
[21] Imagine, Web application generator for the SQL Server platform + user guide, Eddy Luursema, Arnoud van Bers, HAN University, Arnhem The Netherlands, 2005.
[22] Even more Fully Communication Oriented Information Modeling, Jan Pieter Zwart, Guido Bakema, to be published soon.
