# The FCO-IM Analysis Process of Fact Stating Sentences

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 2007

*Abstract*

In the Fact Oriented Modeling (FOM) method Fully Communication Oriented Information Modeling (FCO-IM), full emphasis is given - i.e.: to a far greater extent than other FOM methods do - to the analysis process of fact stating sentences (fact expressions). The founding principle of FCO-IM, namely that "all objects are in fact facts", strongly contributed to the full understanding, workability and implementability of the FOM analysis process of fact stating sentences verbalized by domain experts and expert users. The introduction of object type expressions to deal with objects as objectifications (or nominalizations) of their existence postulating fact stating sentences proved to be the master key unlocking doors which so far could only be opened using inelegant brute force. Even doors already known to hide interesting things (such as recursive identification) that were still closed to operational analysts, could be opened surprisingly easily by this master key. After a short introduction of the main FOM concepts and the main steps for drawing up an FCO-IM information model, the FCO-IM analysis process of fact stating sentences will be discussed in depth.

*Key words*

information modeling, data use case, semantically equivalent transformation, subtyping, generalization

## 1. Introduction

The process of building an information model (information grammar) for some Universe of Discourse (UoD) and of deriving a database schema from this model is more or less the same in all Fact Oriented Modeling (FOM) methods. With respect to Fully Communication Oriented Information Modeling (FCO-IM), this process is explained and trained in the FCO-IM text books [1][2]. Globally it consists of the following steps:

1. Collect or construct a significant set of concrete example documents (data use cases).
2. Verbalize the data on these data use cases in the form of fact stating sentences (fact expressions).
3. Analyze these verbalizations by adding type level information (classification and qualification).
4. Add basic constraints (value constraints, uniqueness constraints, totality constraints, …).
5. Carry out elementarity tests and take action (split fact types) where necessary.
6. Carry out semantically equivalent transformations (nominalization, subtyping, and generalization), if desired.
7. Add other more complex constraints, if any.
8. Derive a Relational model (database schema) from the information model.

This paper zooms in on step 3, the analysis process of fact stating sentences, because this is the very essence of the modeling process. In FCO-IM this process is much more powerful than in other FOM methods (such as NIAM [3] and ORM [4] [1]). In section 2 the first four steps will be presented. In section 3 diagrammatical aspects will be discussed, with some attention to step 4 as well. Step 5 will be skipped, although these very simple FOM tests rendered normalization completely superfluous (see [1][2],[3][4]). In sections 4, 5 and 6 the power of the analysis process of fact stating sentences will be elaborated on further and related to steps 6 and 7. Step 8 is extensively discussed in [8] and fully automated in the FCO-IM tool CaseTalk[3] [8].

[1] NIAM: Natural language based Information Analysis Method; ORM: Object Role Modeling.

[2] Free download FCO-IM text book: http://www.casetalk.com/php/index.php?FCO-IM%20English%20Book

[3] CaseTalk is the state of the art FCO-IM modeling tool, BCP Software 2002-2007 (www.CaseTalk.com).

## 2. The first four steps

For step 1, it is not so important how the analyst obtains the data use cases, i.e. which techniques are used to collect or construct them, it is just the result that counts: a significant set of concrete example documents that can be used as a starting point for step 2. The analyst might start modeling relevant business processes, zoom in on the information processed and collect concrete example documents from users involved and/or construct example documents by interviewing expert users. Figure 1 shows an example document that served as a demonstration data use case in many presentations and articles [5][6], which is quite suitable for further elaboration. This data use case shows information about floors and rooms and will be extended in the following sections to demonstrate the power of the FCO-IM analysis process of fact stating sentences.

| Floor | Emergency Exits | Room | Seats in Room | Equipment |
|---|---|---|---|---|
| 1 | 2 | 1.1 | 20 | PC<br>BM |
| | | 1.2 | 30 | -- |
| 2 | 0 | 2.1 | 20 | PC |
| 3 | 2 | 3.2a | -- | --- |

**Full names of facilities:**
PC: personal computer, BM: beamer, WB: whiteboard

*Figure 1: A data use case*

During step 2, at least one instance of each type of fact found on the example document(s) is verbalized, i.e. expressed in terms of predicative fact stating sentences. See figure 2. These facts and - as a consequence - the predicative fact stating sentences are preferably *elementary*, i.e. as small as possible, but not further splittable without loss of information. Steps 2 and 3 cannot be clearly separated, because verbalizing implies that facts are classified to belong to certain fact types, after which at least one fact of each class is then verbalized. Each class (fact type), once identified, is usually given a meaningful name (type level information) at once as well (first qualification). During step 3 this process of classification and qualification (ClaQua) is continued.

Floor:
"Floor 1 exists."

Emergency Exits:
"Floor 1 has 2 emergency exits."

Seats in Room:
"Room 1.1 has 20 seats."

Equipment:
"Room 1.1 is equipped with the facility PC."

Facility Full Name:
"The full name of the facility PC is: personal computer."

Room:
"Room 1.1 exists."

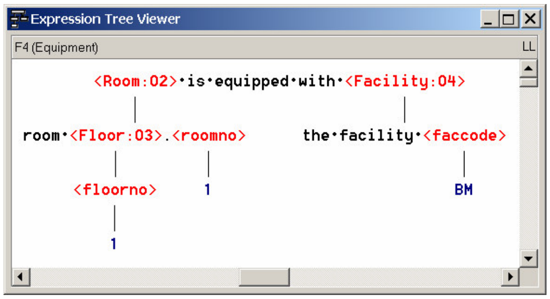
*Figure 2: Elementary fact expressions*

It is not the primary objective of FCO-IM to make a diagram (or a set of diagrams), as is the case in modeling techniques such as Entity-Relationship Modeling and in even more classical FOM techniques, but aims to ClaQua the fact expressions resulting from the verbalization process. As a consequence, an FCO-IM modeling tool is not (and cannot be) a 'drag-and-drop' tool, but instead a tool that offers a sophisticated and powerful but nevertheless intuitive interface for carrying out the ClaQua process, essentially just like it would be done by hand, as shown in figure 3 (but then without having to type things twice). Figure 3 clearly illustrates that the ClaQua process essentially adds more UoD-related type level information to the instance level fact stating sentences. This means at the instance level: indicating *predicates* (the non-underlined sentence parts), *object expressions* (double underlined) and *labels* (values, single underlined). At the type level it means: indicating to which *fact type expression* (FTE: predicate plus typed blanks) each fact expression belongs (F1, F2, etc.), to which *object type expression* (analogous to FTE) each object expression belongs (O1, O2, etc.), and adding names for object types and label types), as was done earlier for fact types.

```
Floor:
F1:  "Floor 1 exists."
          floorno

Emergency Exits:
F2:  "Floor 1 has 2 emergency exits."
       Floor:O1   number
       'Floor 1'
           floorno

Seats in Room:
F3: "Room 1.1 has 20 seats."
       Room:O2     number
      'room 1.1'
    Floor:O3  roomno
             '1'
          floorno

Equipment:
F4: "Room 1.1 is equipped with the facility PC."
       Room:O2                          Facility:O4
        match                                     'PC'
                                                faccode

Facility Full Name:
F5: "The full name of the facility PC is: personal computer."
                             Facility:O4             facname
                               match

Room:
F6: "Room 1.1 exists."
     Floor:O3  roomno
      Match
```

*Figure 3: The full ClaQua process illustrated*

The resulting metadata (FCO-IM information model) is to be stored in a repository. The FCO-IM modeling tool CaseTalk [7] offers such a ClaQua interface and stores the resulting metadata automatically in an FCO-IM tool repository [8], from which the analysis tree can be (re)produced for each fact expression at any time. See figure 4 for the analysis tree of the fact expression that belongs to fact type expression F4 (Equipment) from figure 3.



```
<Room:02>·is·equipped·with·<Facility:04>

room·<Floor:03>.<roomno>        the·facility·<faccode>

<floorno>     1                             BM

1
```

*Figure 4: The CaseTalk Expression Tree Viewer*

## 3. Diagrammatization

Human information modelers prefer to see things not just in textual form or as contents of a repository, but in a diagrammatic form as well. An FCO-IM information diagram (or a set of diagrams belonging to one FCO-IM information model), is a "self-synchronizing auto-presentation" of the contents of the FCO-IM tool repository. As soon as a fact expression has been analyzed, the analysis result should be presented graphically as well. However, this graphical presentation mainly serves the purpose of being nice for the analyst to see, to reflect on during the ClaQua process, and to present the final results. Figure 5 (produced by the FCO-IM modeling tool CaseTalk) shows the same instance data and metadata as figures 2 and 3, but in graphical form.

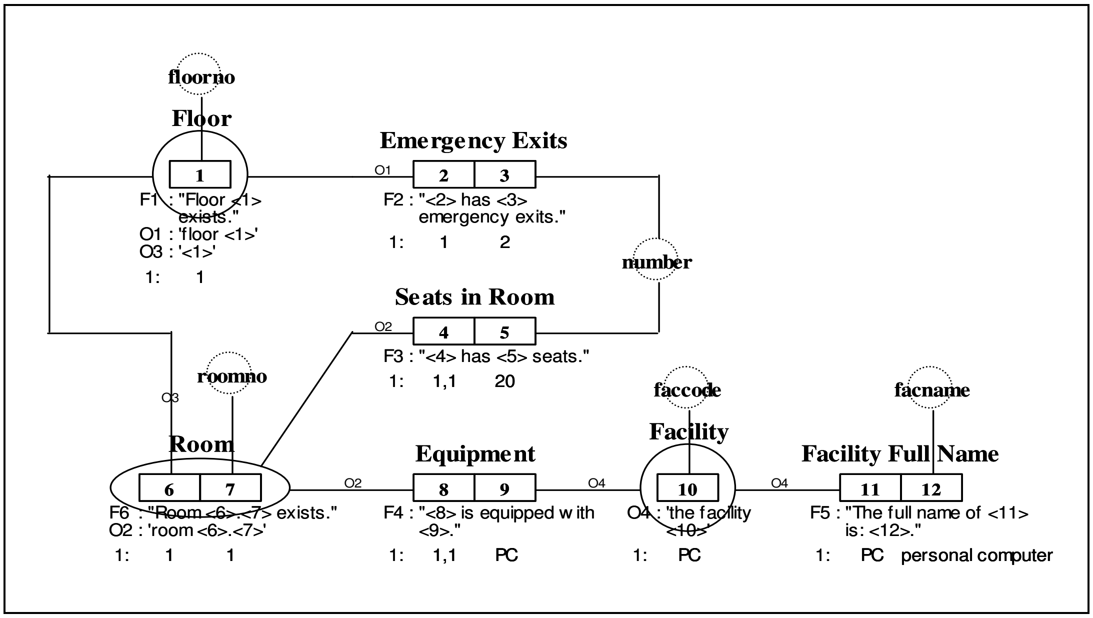


*Figure 5: Preliminary FCO-IM information diagram*

Figure 5 clearly illustrates the most important principle of FCO-IM, namely that ***all objects are in fact facts*** [9]. This principle strongly contributed to the full understanding, workability and implementability of the analysis process of fact stating sentences verbalized by domain experts and expert users. All object type expressions are considered to be nominalizations of existence postulating fact stating sentences: F1 → O1, O3 and F6 → O2, even in case the corresponding existence postulating fact stating sentence is not present (because not considered relevant): ? → 04. These object type expressions O1, O2, O3 and O4 in their turn are substituted into fact type expressions: O1 in F2, O3 in F6, O2 in F3 and F4, O4 in F4 and F5.

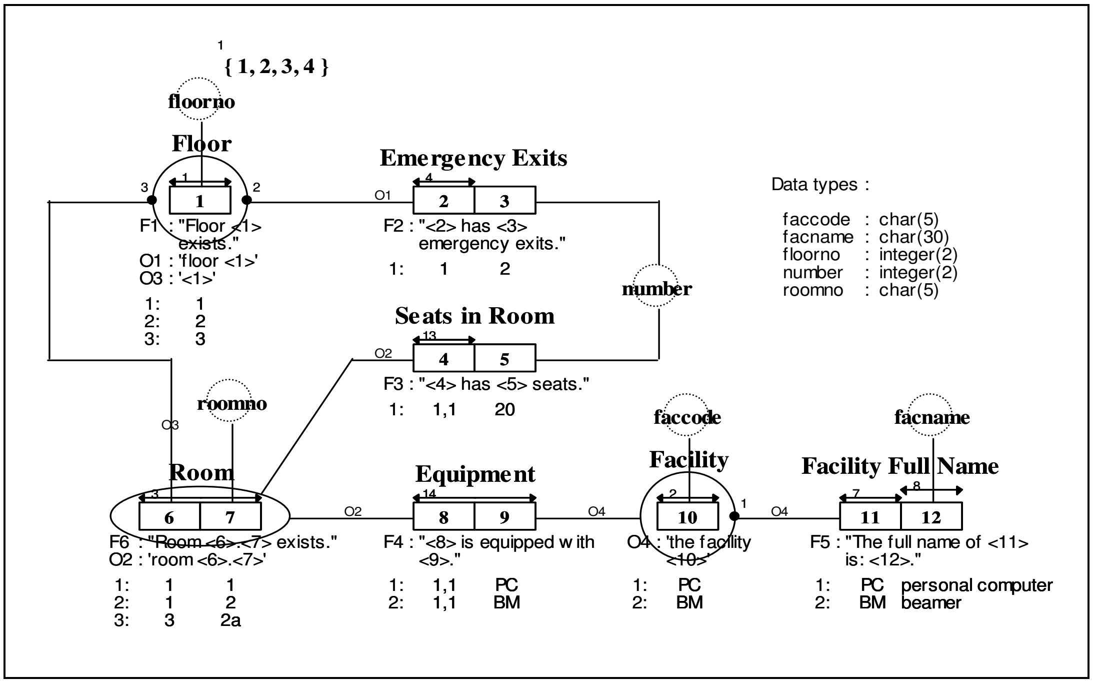


*Figure 6: FCO-IM information diagram with basic constraints, data types and extended population*

To add basic constraints to an FCO-IM information model, several interfaces are conceivable. As in other FOM modeling tools, in the FCO-IM modeling tool CaseTalk a preliminary information diagram (figure 5) is used as a graphical interface for adding basic constraints: value constraints, uniqueness constraints, totality constraints, etc. The diagram is also used as an interface to add textual constraints, for correcting purposes (correct names of fact types, object types and label types, correct text parts of fact type expressions and object type expressions), to add data types, to add more example facts (extend the population) and so on. See figure 6.

All added metadata and data (population extensions) is automatically stored in the FCO-IM repository as well and is freely accessible at any time. Things can be regenerated in textual form for users and domain experts at any time. See figure 7, in which not only the instance level fact stating sentences have been regenerated, but in which also the basic constraints have been verbalized at the type level in an understandable way ("... must be …", "… is uniquely identified by …", "… must have a …").

```
Floor:
"Floor 1 exists."
"Floor 2 exists."
"Floor 3 exists."
"Floor is uniquely identified by floorno."
"Floor must have a Emergency Exits."
"Floor must have a Room."

Emergency Exits:
"Floor 1 has 2 emergency exits."
"Floor 2 has 0 emergency exits."
"Floor 3 has 2 emergency exits."
"Emergency Exits is uniquely identified by Floor."

Room:
"Room 1.1 exists."
"Room 1.2 exists."
"Room 3.2a exists."
"Room is uniquely identified by Floor and roomno."

Seats in Room:
"Room 1.1 has 20 seats."
"Seats in Room is uniquely identified by Room."

Equipment:
"Room 1.1 is equipped with the facility PC."
"Room 1.1 is equipped with the facility BM."
"Equipment is uniquely identified by Room and Facility."

Facility:
(No expressions available)
"Facility is uniquely identified by faccode."
"Facility must have a Facility Full Name."

Facility Full Name:
"The full name of the facility PC is: personal computer."
"The full name of the facility BM is: beamer."
"Facility Full Name is unique on Facility."
```

*Figure 7: The stored data and metadata (re)generated by CaseTalk in verbal form*

## 4. The analysis process considered further

Consider the extended data use case shown in figure 8:

| Floor | Emergency Exits | Room | Seats in Room | Equipment |
|---|---|---|---|---|
| 1 | 2 | 1.1 | 20 | PC (10)<br>BM (1) |
| | | 1.2 | 30 | -- |
| 2 | 0 | 2.1 | 20 | PC (20) |
| 3 | 2 | 3.2a | -- | --- |

**Full names of facilities and their availability:**
PC: personal computer (250), BM: beamer (10), WB: whiteboard (100)

*Figure 8: Extended data use case*

We now extend the UoD: the total number of available items of each facility is to be recorded, and for each room it must be registered with how many items of a facility it is equipped. In figure 9 the ClaQua process is applied to two example fact expressions, one for each new fact type.

Available Facilities:
F7: “There are 250 PCs available.”
number Facility:06
‘PCs’
faccode

Installed Facilities:
F8: "The number of installed PCs in room 1.1 is 10."
Equipment:O7 number
‘PCs in room 1.1’
Facility:06 Room:O4
*match* *match*

*Figure 9: ClaQua of two new fact expressions*

The new metadata (names of fact types, object types, label types, fact type expressions, object type expressions, …) and data (example population) is automatically added to the FCO-IM repository and presented in the diagram. Standard constraints have been added. See figure 10, in which the two new fact types Available Facilities and Installed Facilities are shown. This last fact type caused the automatic objectification of fact type Equipment to an object type of the same name, and the nominalization of fact type expression F4 to object type expression O7, which is to be substituted into fact type expression F8.

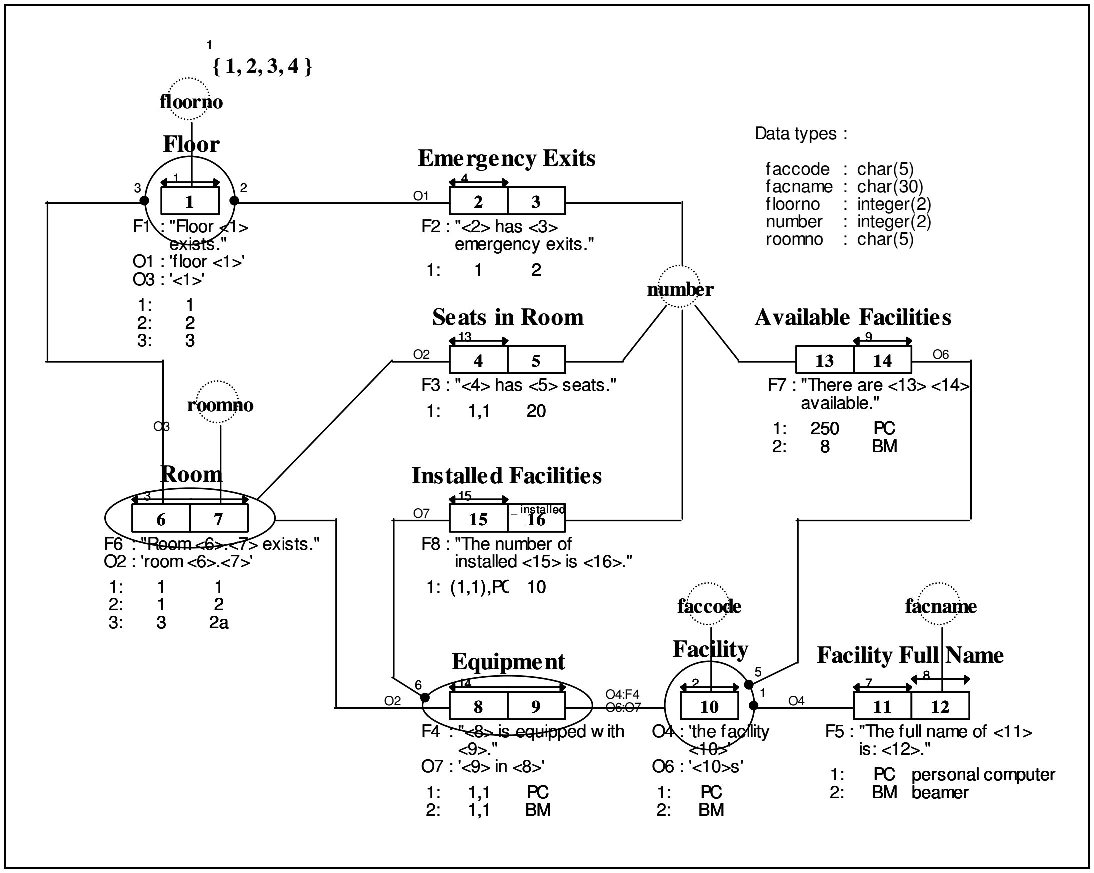


*Figure 10: FCO-IM information diagram with two fact types added, the second automatically nominalized*

## 5. Semantically equivalent transformations

Step 5 concerns carrying out semantically equivalent transformations, if desired. It is interesting that all the familiar FOM semantically equivalent transformations (see [1], [3], [4]) can be carried out in FCO-IM just by doing ClaQua on the same set of fact expressions in a different way. As an illustration, consider fact type Installed Facilities in figure 10, with role 15 played by object type Equipment. This could be modeled instead as a ternary fact type Installed Facilities. Figure 11 illustrates that the FOM nominalization-denominalization semantic equivalence is fully based on doing ClaQua for the same fact expression F8 in different ways. At the left side this yields the binary nominalized fact type / object type Equipment (as in figure 10) playing role 15 in the binary fact type Installed Facilities, whereas at the right side we get a ternary fact type Installed Facilities instead. See figure 12.

Installed Facilities:
F8: "The number of installed PCs in room 1.1 is 10."
Equipment:O7 number
'PCs in room 1.1'
Facility:06 Room:O2
*match* *match*

Installed Facilities:
F8: "The number of installed PCs in room 1.1 is 10."
Facility:06 Room:O2 number
*match* *match*

*Figure 11: The semantic equivalence nominalization-denominalization is based on ClaQua done differently*

In the FCO-IM modeling tool CaseTalk the denominalization of object type Equipment can be done via the diagram instead of redoing the ClaQua of F8 as shown in figure 11 at the right. There is an option 'reduce' in a pop-up menu for fact type / object type Equipment in the diagram shown in figure 10, causing it to disappear and transforming fact type Installed Facilities into a ternary fact type as shown in figure 12. This means that the new ClaQua of F8 is done automatically, i.e. instead of using the ClaQua interface, the diagram is used as a graphical interface on the FCO-IM repository to speed up the denominalization.

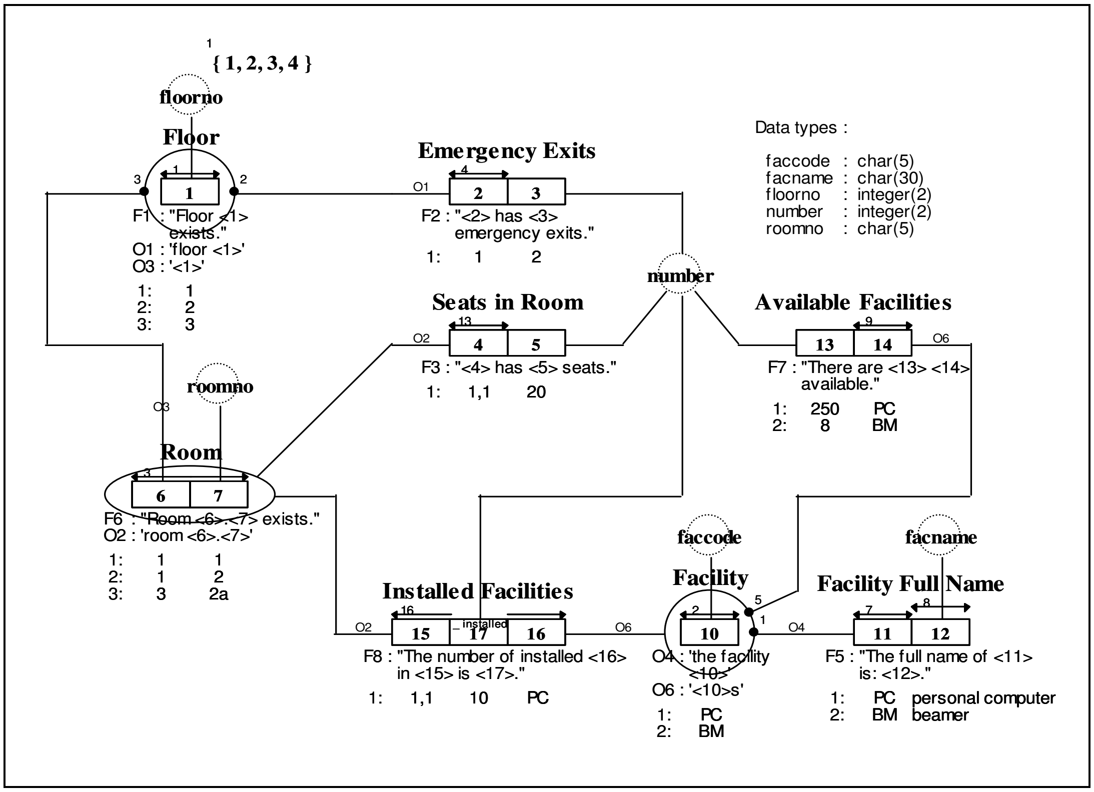


*Figure 12: FCO-IM information diagram with ternary fact type Installed Facilities*

Such nominalization-denominalization transformations can be done the other way around as well, which is more often the case. Therefore we will continue with the nominalized version shown in figure 10.

Not only the nominalization-denominalization transformation, but also the FOM fact-type-to-object-type transformation (also called role-to-object transformation), with as a special case the unary-to-binary transformation, proves to be nothing else than doing ClaQua for the same fact expressions differently[1].

## 6. Subtyping

In a comparable way (i.e. just ClaQua-ing fact expressions) the ClaQua process is capable of dealing with subtyping (or specialization). Suppose that registering the number of installed facilities is only relevant for all the lesson rooms, but not relevant for lecturer rooms. This cannot be expressed in the form of constraints added to the IG yet, because it is not known whether rooms are lesson rooms or lecturer rooms. A fact expression is introduced for this and the ClaQua result is shown in figure 13, in which an extra constraint C1 has been added as well (step 6). Fact types Seats in Room and Equipment are the same as in figure 10.

Room Typing:
F9: "Room 1.1 is a lesson room."
    Room: O2    type of room
    *match*

C1: x in Equipment (8),
    if and only if (x, 'lesson room') in Room Typing (17, 18)

*Figure 13: New fact type Room Types introduced plus extra constraint*

Figure 14 shows the extensions added to the IGD, with Installed Facilities a binary fact type, as in figure 10.

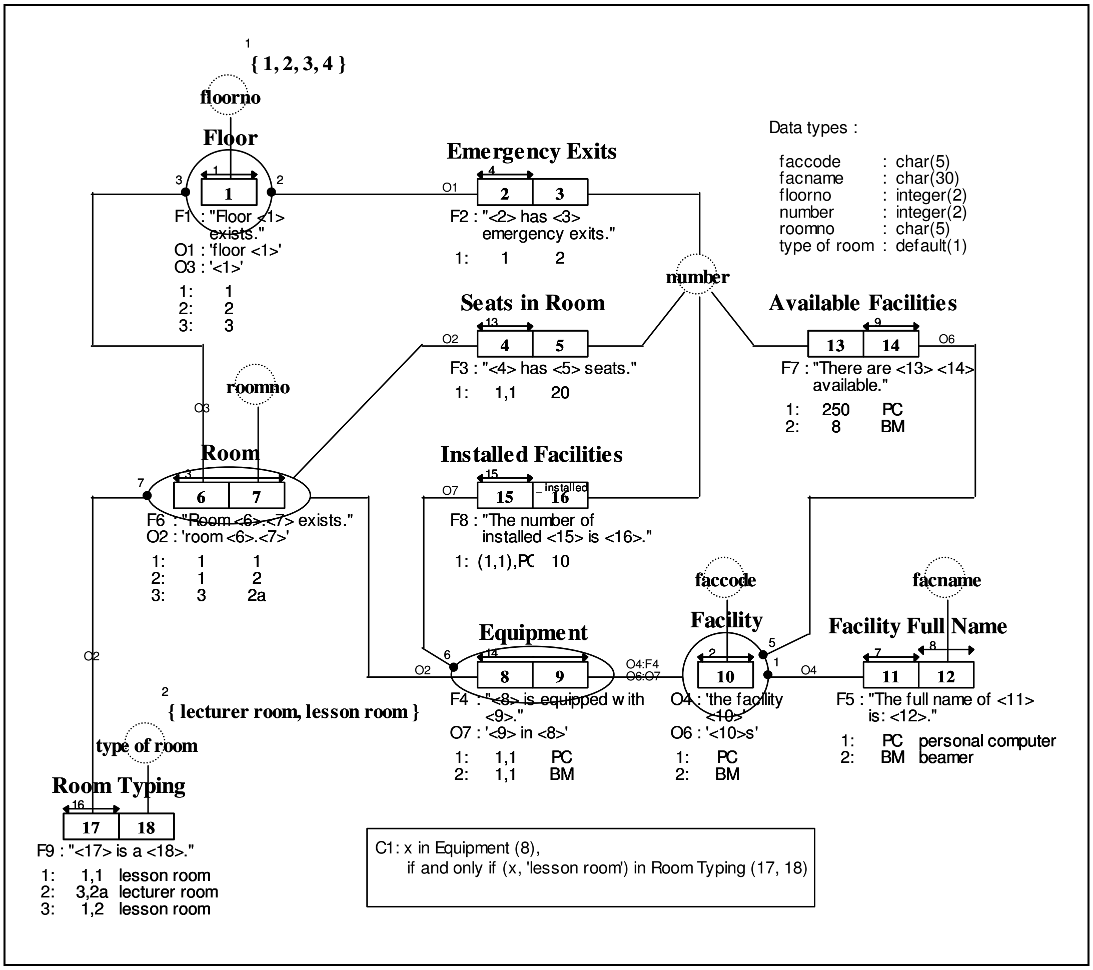


*Figure 14: FCO-IM information diagram extended with fact type Room Types*

The well known constraint minimalization principle[4] now requires us to introduce a subtype Lesson Room of supertype Room by changing the ClaQua of the fact expression belonging to fact type expression F4 of Equipment. See Figure 15.

[4] René Veldwijk, a Dutch database design expert, calls this one of the 10 commandments of good database design

Equipment:
F4: "Room 1.1 is equipped with the facility PC."
Lesson Room:O9 Facility:O4
'Room 1.1' *match*
Room:O1
*Match*

C2: x in Lesson Room (19),
if (x, 'lesson room') in Room Types (17, 18)

*Figure 15: New ClaQua of* Equipment

Figure 16 shows the new diagram with subtype Lesson Room of supertype Room. Constraint 1 is now modeled by letting role 8 be played by Lesson Room, together with totality constraint 8. The new constraint C2 is a subtype defining derivation rule for the derivable (indicated by '*' behind the name) subtype Lesson Room *. A second subtype Lecturer Room could be introduced as well, but in this case it is not needed (it would not play any role).

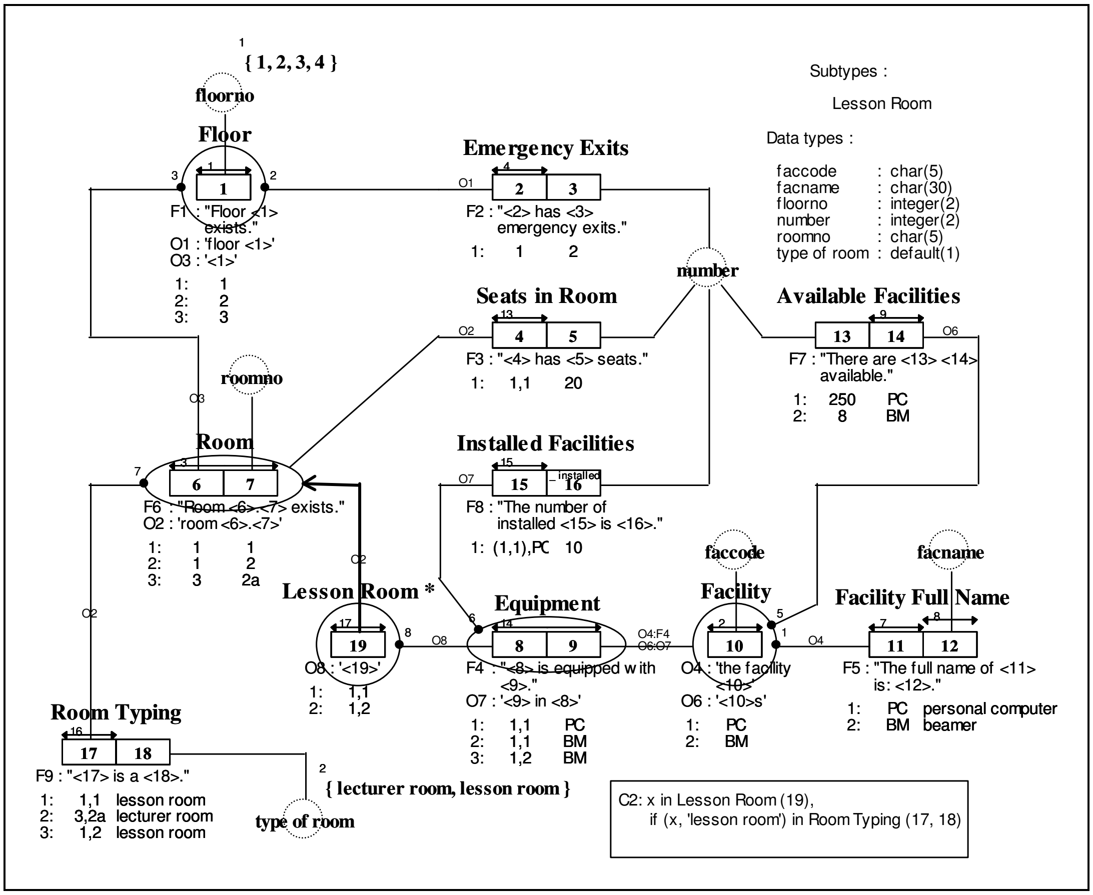


*Figure 16: FCO-IM information diagram extended with derivable subtype* Lesson Room

In the FCO-IM modeling tool CaseTalk, the introduction of a subtype can be done graphically as well: instead of re-doing ClaQua as shown in figure 15. There is an option 'new subtype' in a pop-up menu for fact type / object type Room, which will cause a new subtype to be introduced that can be given the name Lesson Room, and role 8 can be selected to be played by the new subtype Lesson Room. The ClaQua for fact expression F4 will be redone automatically as well, i.e. the diagram is again used as a graphical interface on the FCO-IM repository to facilitate introducing subtypes.

It is also possible to delete fact type Room Typing and to consider F9 as the existence postulating fact type expression for subtype Lesson Room (see figure 17 for the different ClaQua of F9), which so becomes a *declarative subtype* (i.e. rooms belonging to Lesson Room are explicitly declared by the population).

Lesson Room:
F9: "Room 1.1 is a lesson room."
Room: O2
*Match*

*Figure 17: ClaQua for* Lesson Room *as declarative subtype*

As a consequence the subtype derivation rule C2 and the '*' in the name of the subtype can be deleted as well. See figure 18.

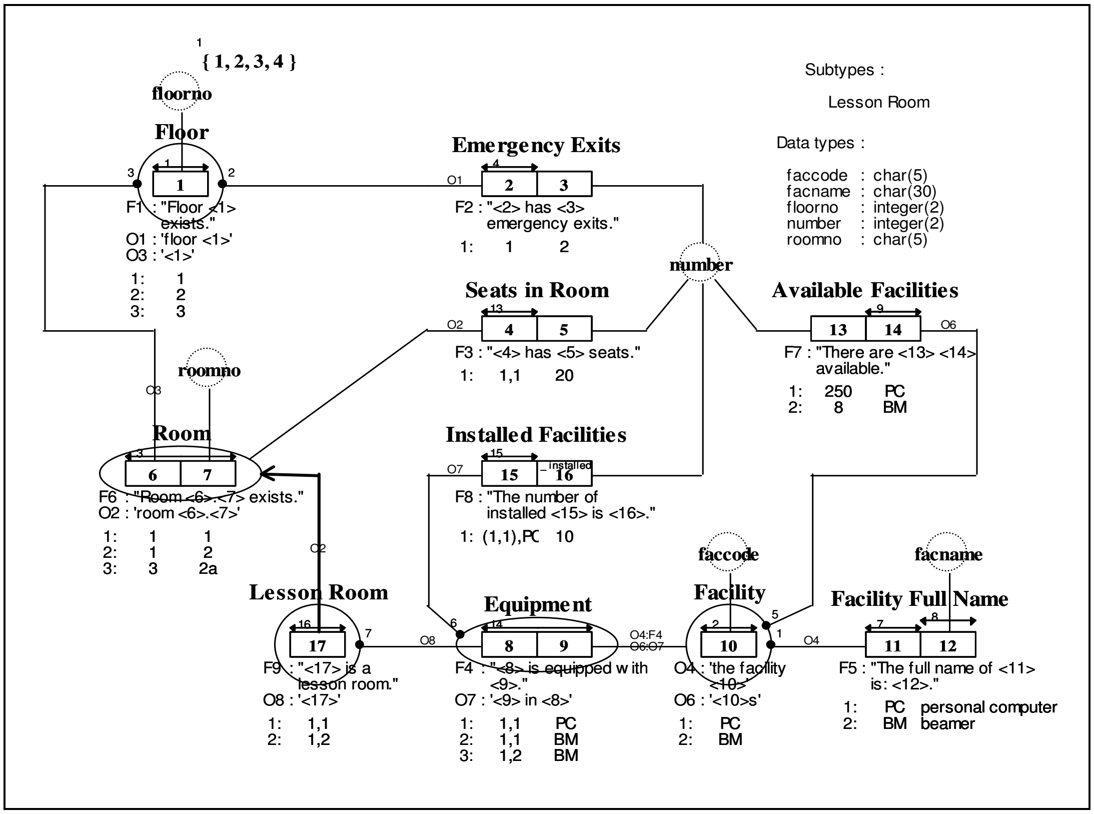


*Figure 18: FCO-IM information model with declarative subtype* Lesson Room

## 7. Generalization

Suppose that floors can have special rooms, such as studies, auditoria and storage rooms. These special rooms are not indicated by a floorno and a roomno (1.1, 1.2, 3.2a and so on), but by a floorno and a room name: study, auditorium, etc. The fact expression in figure 19 is an existence postulating fact expression for a new declarative subtype Special Room of supertype Room.

Special room:
F10: "There is a study on floor 1."
Room: O9
'study on floor 1'
room name Floor: O1
*Match*

*Figure 19: ClaQua of existence postulating fact expression for* Special Room

See figure 20 for the resulting extensions to the information diagram. The above ClaQua process leads automatically to a so-called generalization of object type Room, because there are now two different identification structures for it: Room can be identified by object expression O2: 'room <6>.<7>', but now also by object expression O9: '<19> on <6>'. Automatically roles 7 and 19 have been indicated as OP for optional. If every room is either a special room identified by a floorno and a room name or a room identified by a floorno and a roomno (i.e. by O2 or by O9), then an extra constraint C3 must be specified so that for each tuple of Room, either role 7 or role 19 must have a value (role 6 always has a value: it is not optional).

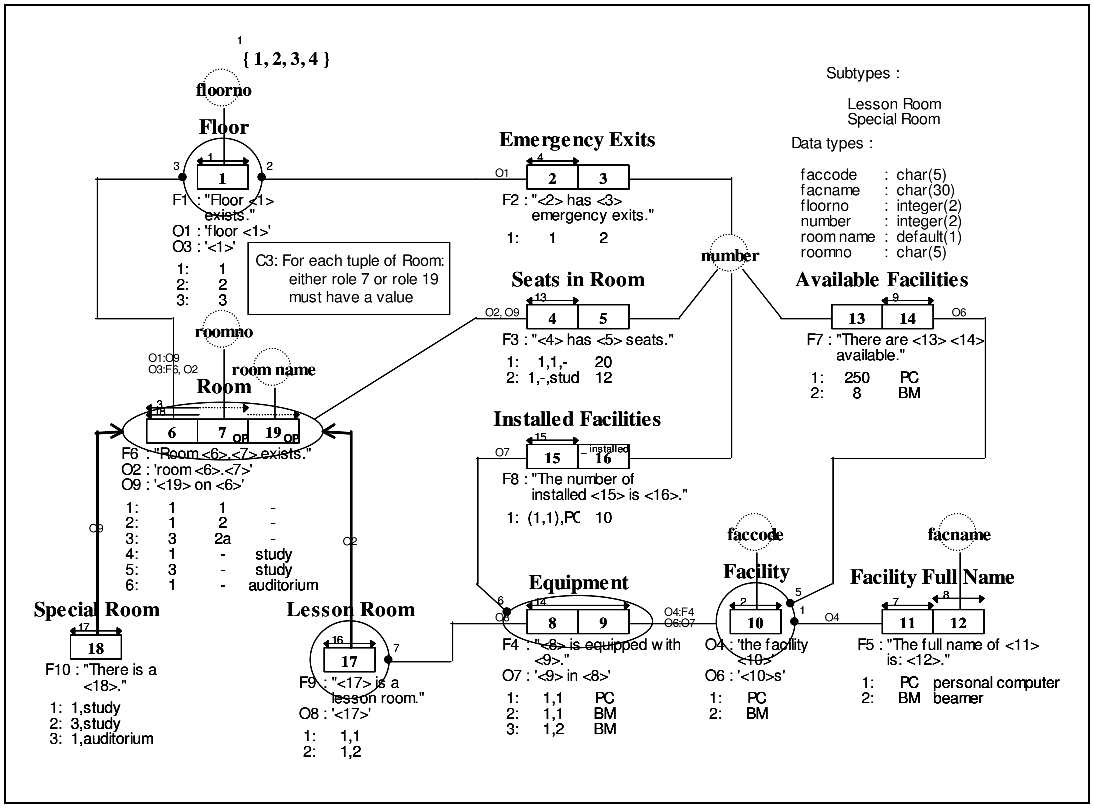


*Figure 20: FCO-IM information model with subtype Special Room and generalized object type Room*

## 8. Summary and conclusion

The FCO-IM ClaQua process not only deals with nested predicative analysis of fact stating sentences verbalized by users and domain experts, but can be used as well to produce semantically equivalent models and to introduce specialization and generalization. Even truly recursive identification is covered [1][10]. As a consequence FCO-IM can claim to be the only fully communication orientated information modeling method [11], and also to be the only method (even the only FOM method) that complies with the 100% conceptualization principle [12]. The FCO-IM modeling tool CaseTalk [8] supports all this.

## References

[1] Bakema Guido, and Jan Pieter Zwart, Harm van der Lek, 'Fully Communication Oriented Information Modeling (FCO-IM)', English translation of Dutch text book, HAN University, The Netherlands, 1999.

[2] Engelbart Marco, and Rob Arntz, Jorg Janssen, 'Fully Communication Oriented Information Modeling: exercises and cases', English translation of Dutch text book, Arnhem, The Netherlands, 2001.

[3] Nijssen G.M., with T.A. Halpin, 'Conceptual Schema and Relational Database Design', Prentice-Hall, Sydney, Australia, 1989.

[4] Halpin T.A., 'Information Modeling and Relational Databases', Academic Press, USA, 2001.

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

[6] Manoku Elton with Guido Bakema, 'Integrated Tool Support for Datawarehouse Design', Journal of Conceptual Modeling, January 2003.

[7] Bakema Guido, 'FCO-IM automodel and the thereof generated FCO-IM repository', HAN University, 1999.

[8] BCP Software, 'CaseTalk', FCO-IM modeling tool with integrated user guide, Utrecht, The Netherlands, 2002.

[9] Lek H. van der, with G.P. Bakema, 'Objecten zijn in feite feiten' (i.e. 'Objects are in fact facts'), common publication by Atos Origin and HAN University), Den Haag/Nijmegen, The Netherlands, 1991.

[10] Azizah Fazat Nur, and Guido Bakema, Jan Pieter Zwart, 'A Case Study of Recursive Data Modeling', first Libyan international symposium on Information Systems Modeling and Development, Tripoli, Libya, June 2006.

[11] Zwart J.P., with G.P. Bakema, 'Advances in FCO-IM (1); Classification, Qualification and Semantic Equivalence: Even more fully Communication Oriented Information Modeling', to be presented at the international OTM-ORM'07 workshop, Vilamoura, Portugal, November 2007.

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