Guided Tour: Intelligent Conceptual Modelling in EER and UML-like class diagrams with iCOM compared to ORM2

Abstract. In this guided tour we illustrate the advantages of intelligent conceptual modelling, i.e., demonstrate benefits a modeller obtains by using *automated reasoning over a conceptual data model*. This is conducted through using the UML-like graphical interface of the ICom tool[1] together with the Racer automated reasoner versus common model *validation* as implemented in the NORMA CASE tool for ORM2.

Example 1. Let us start with an example EER/UML diagram, showing all the main features of the conceptual modelling language provided by ICOM:

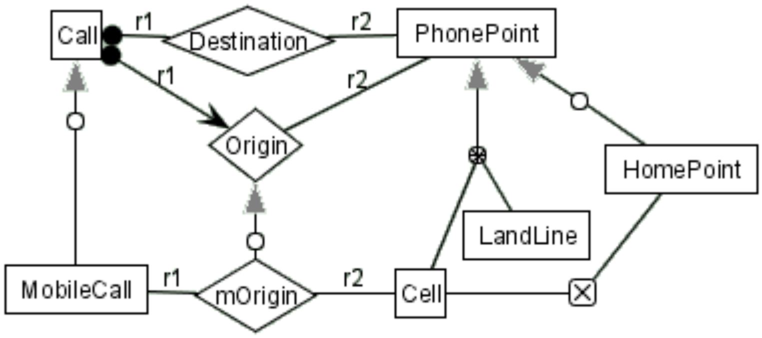


Figure 1. Phone points example 1 in ICom.

Rectangles denote classes, diamonds relations, closed arrows subclassing, starred circle disjointness and covering, black blob mandatory, and open arrow functional ("at most one").

The above diagram has the following ORM2 equivalent (we take ORM's "verbalizer labels" also as name of the relation):

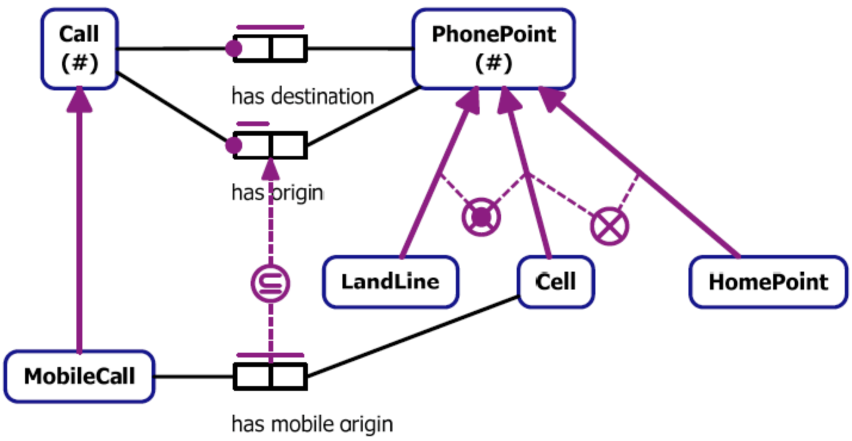


Figure 2. Phone points example 1 in NORMA with ORM2; disjoint classes *in Fig.1 are represented in ORM as disjoint* is-a *relation (encircled cross) and reference scheme (#) is added to avoid validation errors.*

[1] Modified from the iCOM tutorial, available online at http://www.inf.unibz.it/~franconi/icom/tutorial.html. iCOM version used: v0.3.0, built on 4-7-2007, at: http://www.cs.uns.edu.ar/~prf/ontoeditor.zip.

The above diagrams in *Figure 1* and *2* state that mobile calls are a type of calls (ISA link between entities); that phone points are partitioned between cell points and landline points (i.e., any phone point is either a cell or a landline point, but not both: they form a covering and disjoint ISA hierarchy); that home points are among phone points, none of them being a cell point (i.e., the home point entity is disjoint from the cell entity). Each call has at least one destination phone point (mandatory participation of cell to origin), while it has exactly one origin phone point. Mobile calls are related to cells through the mOrigin relationship. Finally, the binary relationship mOrigin is included in the binary relationship origin.

What are the consequences of the EER diagram? Upon running the reasoner, ICOM automatically produces the following completed diagram:

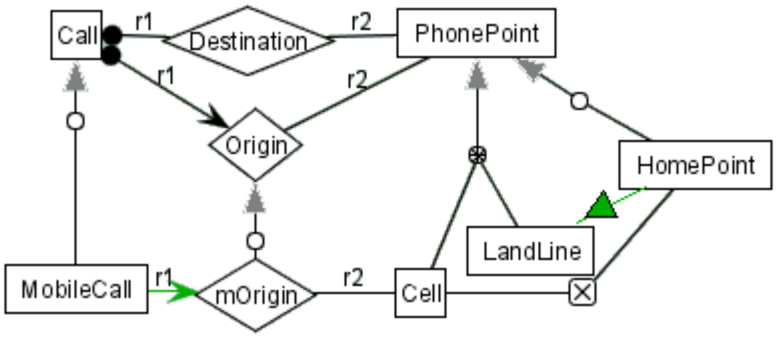


Figure 3. Phone points example 1 in ICom, with deductions.

It is stated that in the above scenario it is necessarily true (**green** deduction) that any home point is also a landline point, and that each mobile call may have an origin from at most one cell point. In order to understand why this is true, consider the following.

- **Given** that (i) the class of home points is a subclass of all the phone points, and (ii) it is disjoint from the class of cell points; since (iii) any phone point is either a cell point or a landline point, **then** any home point should necessarily be a landline point.
- **Given** that (i) the mOrigin binary relationship is included in the origin binary relationship, i.e., any pair in mOrigin is also among the pairs in origin; (ii) since each call participates exactly once as first argument to the origin relationship, **then** if I take a generic sub class of calls, such as the class of mobile calls, and a sub relationship of the origin relationship, such as mOrigin, then we can conclude that necessarily each mobile call participates at most once as first argument to the mOrigin relationship. Nothing can be concluded about the minimum participation, since the mOrigin relationship may not contain all calls in the Origin relationship.

The NORMA validator, being a (syntax) validator and not a reasoner, does not detect these deductions. Manually adding the deduced information gives us the model as depicted in Figure 4. This, however, results in a validation error on HomePoint, stating that "Supertypes for object type 'HomePoint' in model 'Phone Points' must be compatible and intransitive". Thus, deleting the (redundant) is-a between HomePoint and PhonePoint solves the validation error.

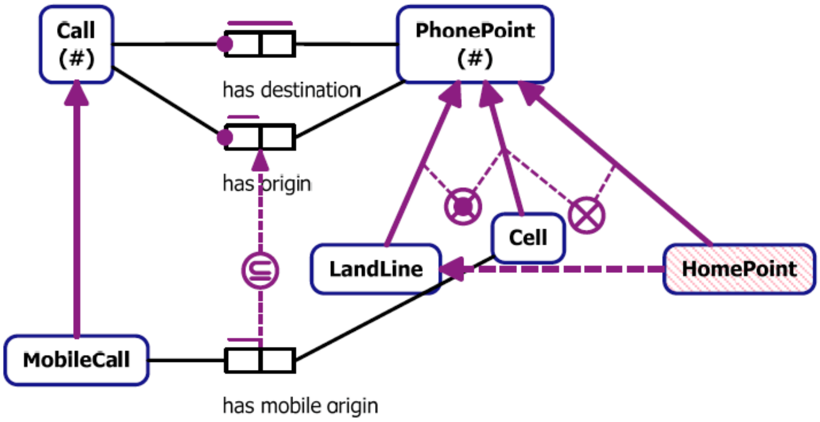


Figure 4. Phone points example 1 with NORMA in ORM2, after having added the deductions manually *and running the validator (result in red).*

Let us now add an ISA link to the original diagram from *Figure 1*, stating that each cell point is among the landline points:

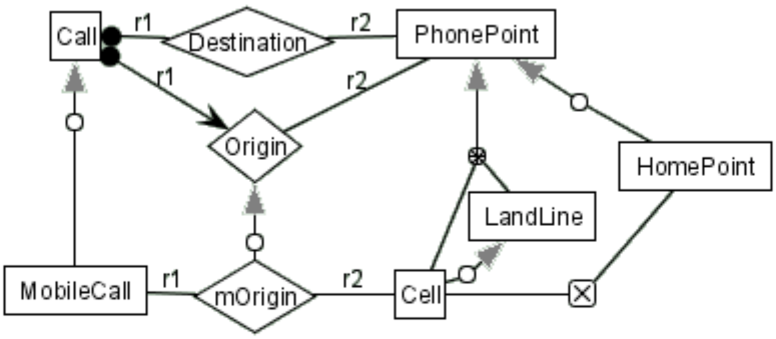


Figure 5. Original diagram with an additional ISA *link between* Cell *and* LandLine*.*

This has the following ORM2 equivalent, as depicted in Figure 6. *Observe that the NORMA validator returns the same type of error as above: "Supertypes for object type 'Cell' in model 'Phone Points' must be compatible and intransitive", and does not conclude the following.*

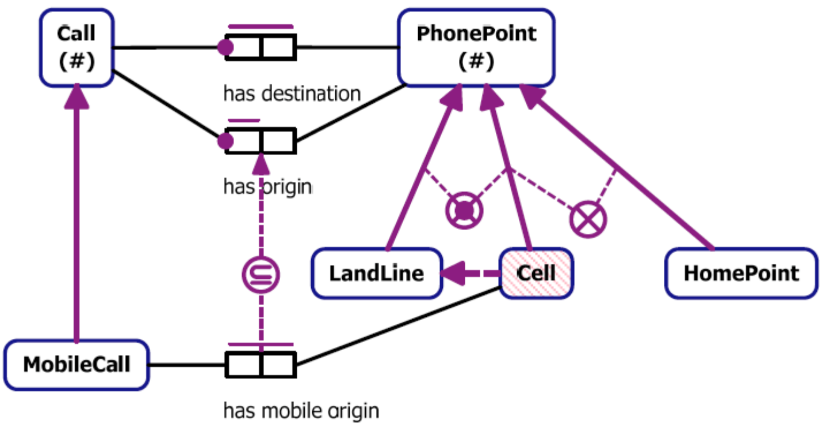


Figure 6. Original diagram with an additional ISA *link between* Cell *and* LandLine*, together with validator result (in red).*

ICOM now concludes the following three deductions, elaborated on after *Figure 7*.

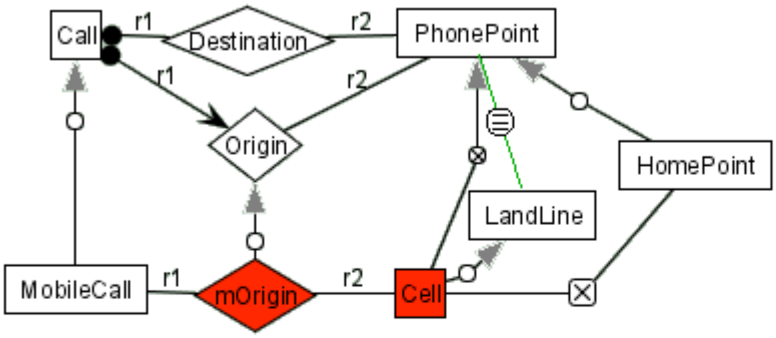


Figure 7. ICom deductions based on the model depicted in Figure 5.

- The cell entity type is inconsistent, i.e., does not—*cannot*—have any instance, since the disjointness constraint in the ISA link states that there is no element in common between cell and landline phone points. The empty set denoted by the cell entity type is the only set that can be at the same time disjoint and a subset of another set.
- Since the phone point entity is formed by the union of the cell and landline entities, and the cell entity is inconsistent, the landline entity becomes equivalent to the phone point entity.
- Since there is no cell point, there is no pair in the mOrigin relationship either (i.e., it is inconsistent): the diagram states that any second argument of the mOrigin relationship should be of the cell type.

The mobile call entity is not inconsistent, since it may be populated by calls that have no mOrigin at all (this is possible, because there is no mandatory participation constraint).

Let us now add a cardinality constraint, stating that now each mobile call should participate at least once to the mOrigin relationship (i.e., a mandatory participation constraint):

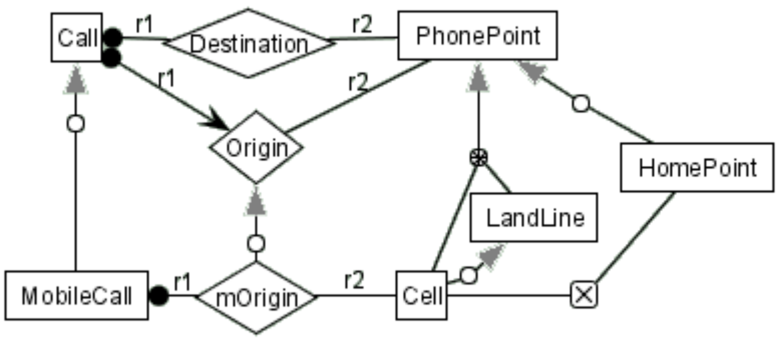


Figure 8. Model as in Figure 5, *with an additional cardinality constraint for MobileCall.*

This has the ORM2 equivalent as shown in Figure 9.

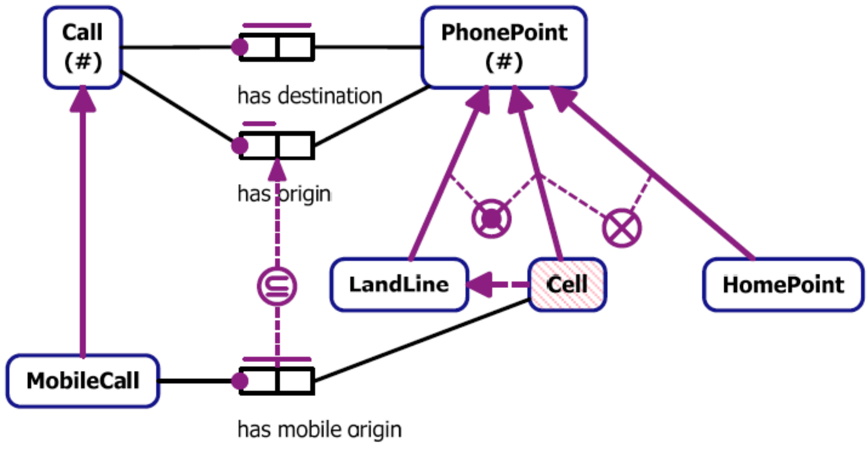


Figure 9. Model as in Figure 6*, with an additional cardinality constraint for MobileCall.*

Now ICOM deduces that the mobile call entity is inconsistent as well (see *Figure 10*).

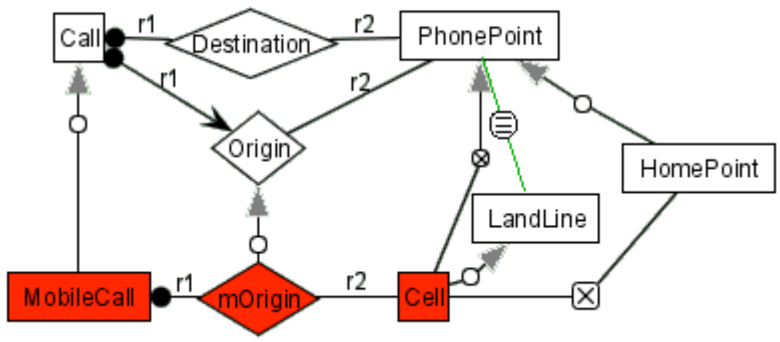


Figure 10. ICom deductions for the model of Figure 8.

When we force the landline point entity to be inconsistent as well—by adding an IsA link between landline points and cell points—then there will be a cascade effect. The phone point entity becomes inconsistent, since it is the union of two inconsistent classes. Therefore, the home point entity will be inconsistent as well, since it is a sub entity of an inconsistent entity. The destination and origin relationships will be inconsistent, since there is no phone point participating to them. Given the mandatory participation constraint of the call entity to the destination (or origin) relationship, we can conclude that the call entity is necessarily inconsistent as well:

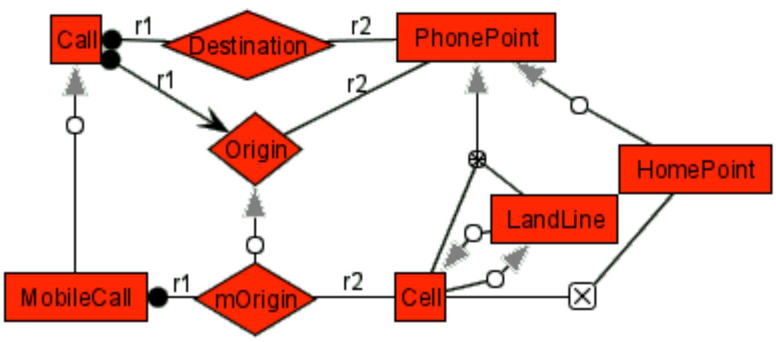
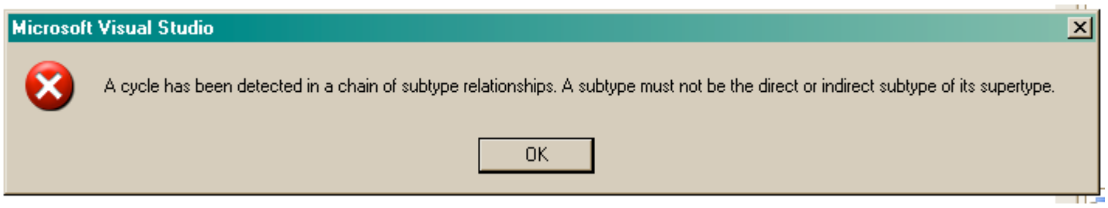


Figure 11. ICom deductions when an additional IsA is added in the model of Figure 8 *between landline points and cell point.*

This change, adding another subtype link, from Landline to Cell, results in a MS Visio error, i.e a NORMA error; hence, a prevention *of the inconsistency.*



<u>*Example 2.*</u> Let us now consider a different example diagram, showing some reasoning involving attributes and domains; see *Figure 12*.

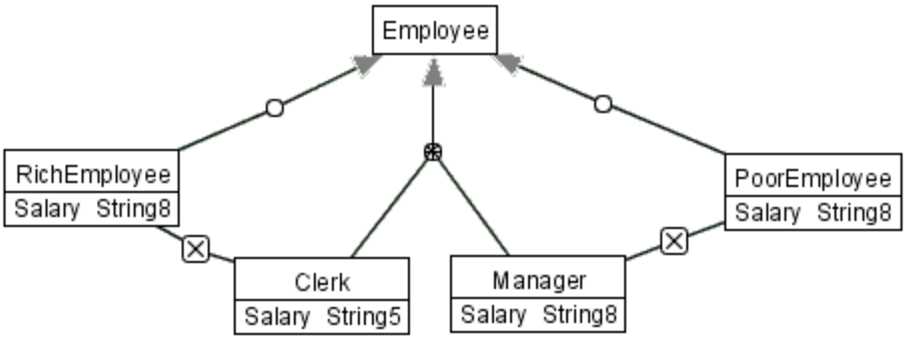


Figure 12. ICom conceptual model for Example 2*, now focusing on attributes of classes.*

Here we have a partition of employees between clerks and managers, plus two more sub-entity types of employee, namely rich employee and poor employee, that are disjoint from the clerk and the manager entities, respectively. All the sub entity types have the salary attribute restricted to a string of length 8, except for the clerk entity that has the salary attribute restricted to be a string of length 5.

The "equivalent diagram" in ORM—depicted in Figure 13*—is actually different, because ORM is an attribute-free language so that one needs to define the same attribute only once instead of 3 times, and one cannot have two value types with the same name that have a different "text:fixed length". Here, we called the "string5" salary* Salary1*. In addition, the EER model of* Figure 12 *does not make explicit the participation conditions for the entity- and value types in the* has salary *ORM-roles. However, the deductions (explained below) still hold, i.e., also with the ORM model, one would derive that rich employees are managers and that* PoorEmployee *is inconsistent.*

Considering the ICom deductions (see *Figure 14*), it is easy to see that a necessary consequence of the above diagram is that rich employees are managers and that poor employees are clerks. However, since that the salary attribute for clerks and poor employees have incompatible domains, we can not have any poor employee in any legal database, i.e., the poor employee entity is inconsistent.

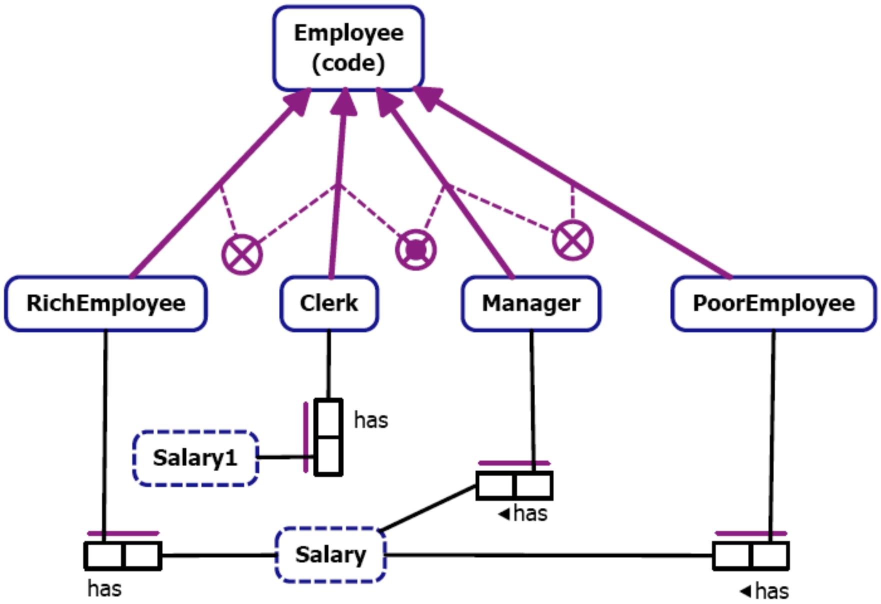


Figure 13. Near-equivalent diagram of Example 2 *in ORM.*

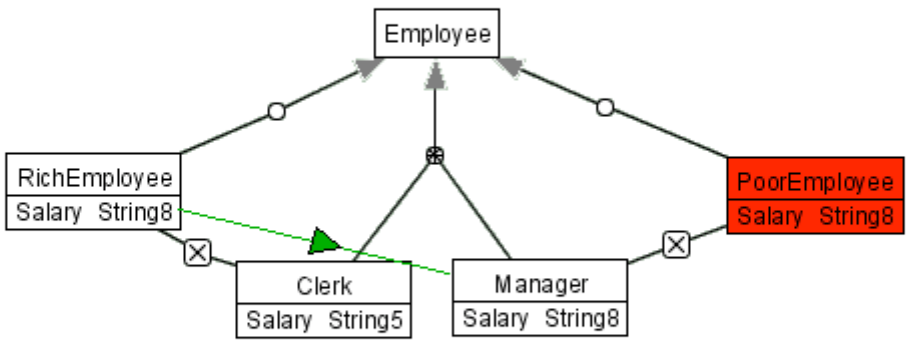


Figure 14. ICom deductions obtained from the model of Figure 12.

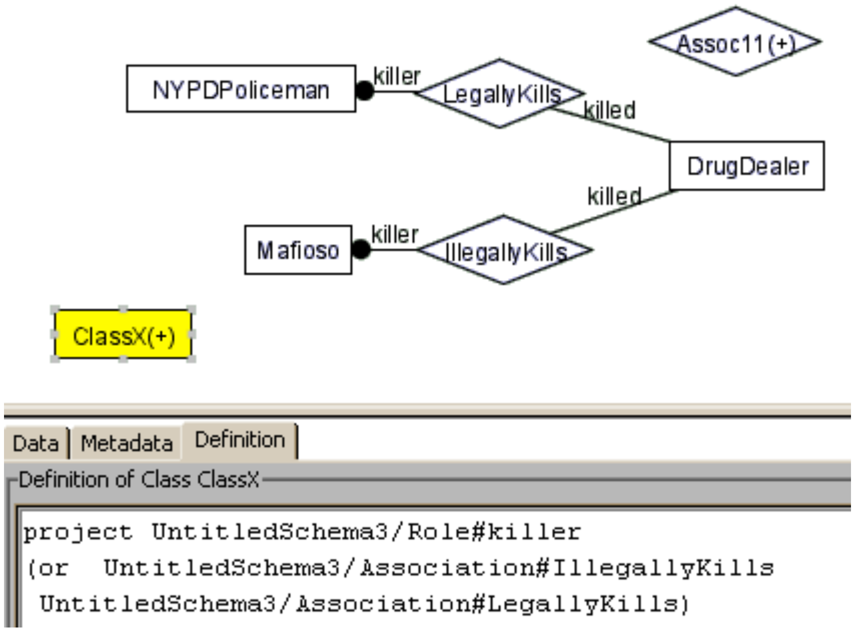


Figure 15. ICom model of Example 3*, with the class definition by means of a view.*

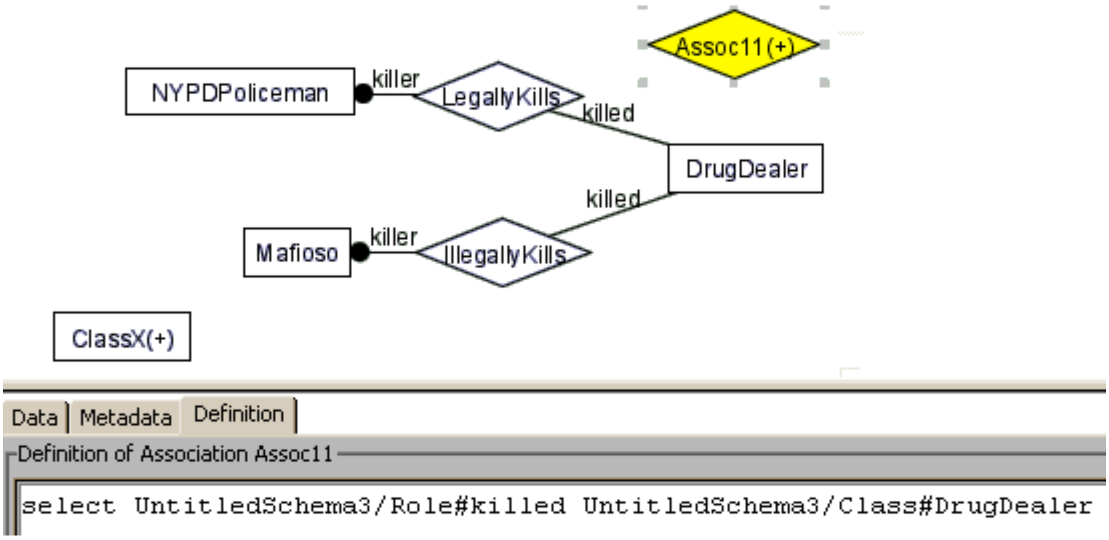


Figure 16. ICom model of Example 3*, with the relation definition by means of a view.*

Example 3. As the last example, let us introduce a conceptual model where some entity and some relationship are defined by means of views over other elements of the model. The view language is an ICom rendering of the $\mathcal{DLR}$ description logic. The view language includes two syntactic sorts: one for entities and one for relationships. Full boolean operators are allowed for both entities and relationship expressions, plus a *selection* operator (selecting tuples in a relationship with a specific entity type in some named role argument) and a unary *projection* operator (projecting a relationship over a named role argument). A generalised projection operator with cardinality restrictions is available as well.

The diagram depicted in figures 15 and 16 introduces NYPD policemen, each one of them legally killed at least one drug dealer, and mafioso people, each one of them illegally killed at least one drug dealer. Moreover, a relation view has been defined (*Figure 16*) as the relation having the "killed" attribute selected to be of the drug dealer type (the view makes use of the selection operator), and an entity view has been defined (*Figure 15*) as the entity obtained by projecting over the killer attribute the union of the legally and illegally kills relationships. As it is shown in the deduced diagram (*Figure 17*), the relation view is a relationship including both the legally and the illegally kills relationships, while the entity view is clearly a super entity of both the NYPD policeman and the mafioso entities.

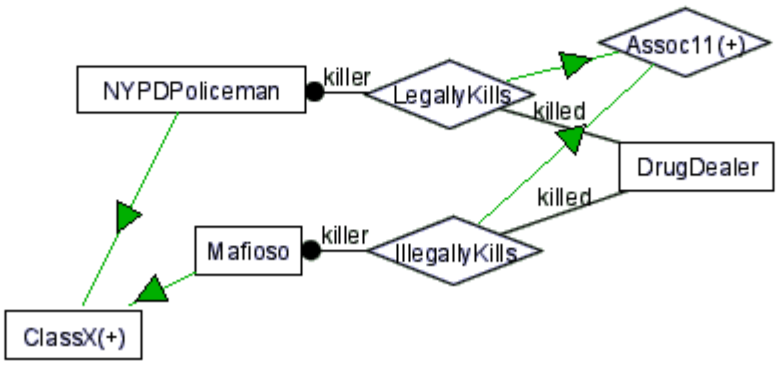
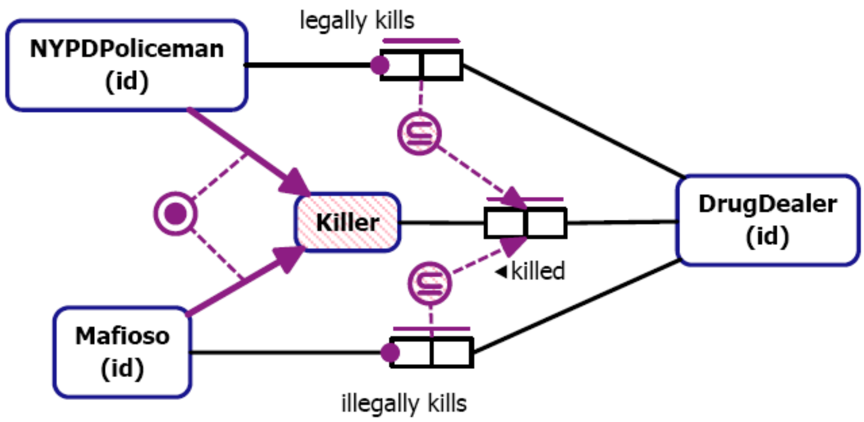


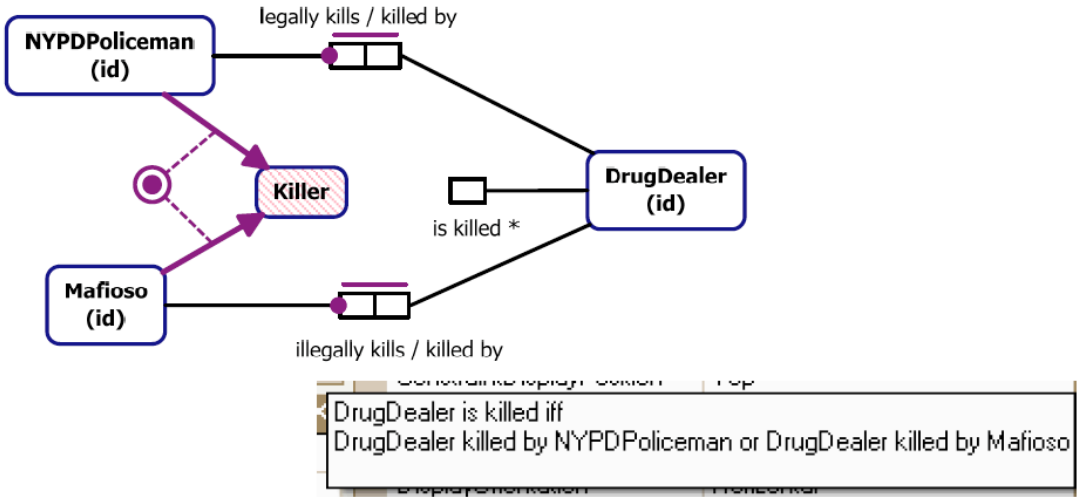
Figure 17. ICom deductions for the model of Example 3.

For the (not-equivalent) scenario in ORM, we have two options, A and B, which are included in two figures below; the projection and selection partially translate to the information derived above, with two options for the selection (see explanation below).

A

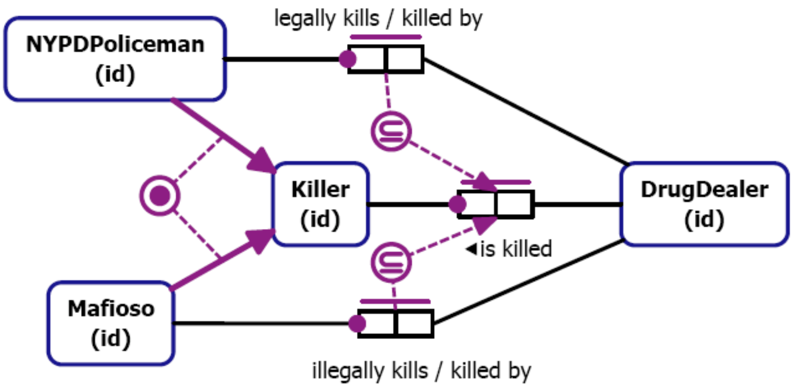


B

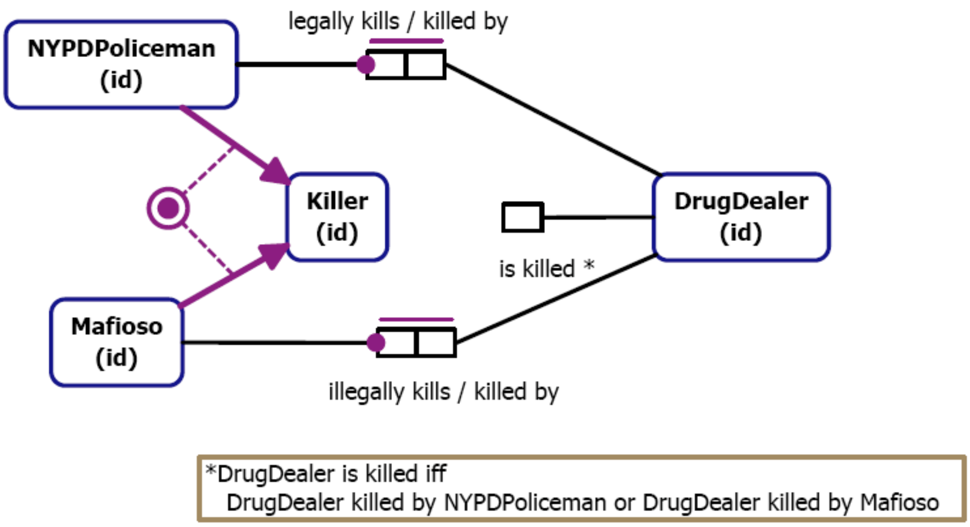


Adding the "entity view" for Killer, *we add it as supertype, This returns a validation error, because* Killer *does not have a reference scheme yet.*
Second, the killed *relation.*

- *<u>option A</u> is close to what is derived with the EER model. However, the two subset constraints return a validation error: "Subset constraint 'SubsetConstraint2' and mandatory constraint 'SimpleMandatoryConstraint32' in model 'Phone Points' are not well-modeled. Suggested ways to fix: (1) Delete the mandatory constraint. (2) Add another mandatory constraint on the superset role. (3) Delete the subset constraint.".* *Manual changes into a valid model leads to the following:*



- Option B *is different in that in the ORM model a unary fact type with a derivation rule is added (*is killed *is true if DrugDealer participates in* killed by NYPDPoliceman *or* killed by Mafioso *or both). Option B is a more straightforward approach when a modeller does not add the projection and selection at the same time.*



Thus, with ORM as currently implemented, the onus is on the modeller to think of the derivations and explicitly represent it, whereas this is derived in iCOM.

Last, but not least, the ICom diagrams do have their mappings to a formal representation by availing of the $\mathcal{DLR}$ description logic and link to a DL reasoner using the DIG interface.
In the meantime while producing this guided tour, developments in expressive DL languages and transformations from industry-grade conceptual modelling languages to DLs have been pushing the limits further at a theoretical level, idem ditto for automated reasoners even already on an implementation level. In due time, these advances will be harmonised, so that one can benefit from satisfiability checking, interesting deductions, and "glass box reasoning" with automatically generated explanations with conceptual models that use more fancy language features.
In addition, one can extend the approach to both *intelligent querying* of a database by using the conceptual model and logic-based *reverse engineering*. Advances in those fields will be elaborated on in other guided tours.