Objectification and Atomicity

Terry Halpin (p. 2 updated 2020 April 28)

In the original NIAM and its initial extension to ORM, a fact type could be objectified only if was binary or longer and it had a spanning uniqueness constraint (e.g. see both editions of *Conceptual Schema and Relational Database Design* as well as the first edition of *Information Modeling and Relational Databases*). Objectification of a fact type with a spanning UC is non-problematic, and its formal semantics are well defined, so that kind of objectification is ignored in the rest of this paper.

Later, ORM was relaxed to allow objectification of 1:1 fact types (while still ORM1). Later again, ORM was extended to ORM2, which allowed any fact type (unary or longer, with or without a spanning UC) to be objectified (see ORM glossary in the second edition of *Information Modeling and Relational Databases*).

Objectification of unary fact types is fine (they have a spanning UC, asserted or implied). Apart from the unary case however, I now believe that both those relaxations are undesirable. In other words, *objectification should be restricted to fact types with a spanning UC*. While it is likely difficult to make this change in NORMA, as many existing schemas would become illegal, I would prefer these relaxations to be at deprecated in NORMA.

My main reasons for this change are:

(1) To ensure that ORM facts entered by users are atomic;
(2) To avoid confusion when populating fact types attached to the objectification of a fact type without a spanning UCs;
(3) To ensure that all object types are populated in a consistent way, using just the preferred identifier;
(4) To avoid confusion over whether an objectification may be declared independent;
(5) To provide a simple formal semantics of objectification without requiring any sub-conceptual choice of a preferred UC;
(6) Remodeling a schema to avoid non-spanning objectifications typically results in a better schema.

Page 510 of the second edition of *Information Modeling and Relational Databases* extended the Relational mapping (Rmap) procedure with the following step:

0.7 Indicate mapping choice where required for any objectified associations that have no spanning uniqueness constraint.

As discussed on pp. 523-524, when the objectified fact type has multiple UCs, one of these should be chosen as preferred for the relational mapping. Such a choice is often not a conceptual issue (e.g. choosing the husband or the wife role when objectifying "Person is husband of/is wife of Person" as CurrentMarriage).

Within the context of the UoD, an *atomic (elementary or existential) fact type cannot be split without information loss into two or more simpler fact types that collectively involve the same object types*. Restricting ORM fact types to atomic fact types simplifies various mapping procedures, and treats the grouping of multiple atomic fact types into implementation structures (e.g. relational table schemes) as an implementation issue not a conceptual issue.

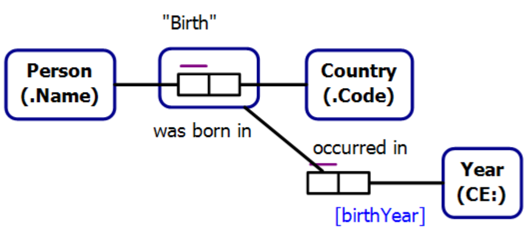
It is possible to formally model objectification of a fact type with a non-spanning UC without violating atomicity. However, this requires that:

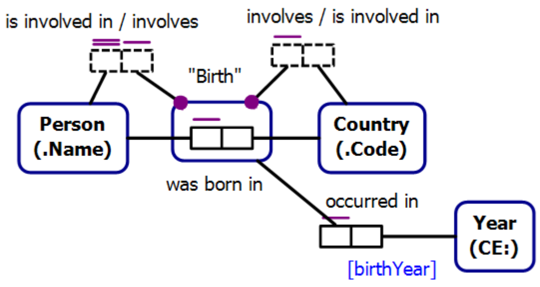
(1) the preferred identifier of the object type resulting from the objectification is based on a UC corresponding to a single UC spanning *n*-1 roles in the original fact type; and
(2) this preferred identifier is used to identify instances of the objectified object type in each non-referential (aka non-existential) role hosted by that object type.

In practice, these two requirements may often prove to be confusing to many users, especially when the underlying link fact type representation of the objectification is not displayed (the default case in NORMA). Moreover, any objectification of a fact type with a non-spanning UC is easily (and better) modelled without using such an objectification. The following pages discuss some typical cases.

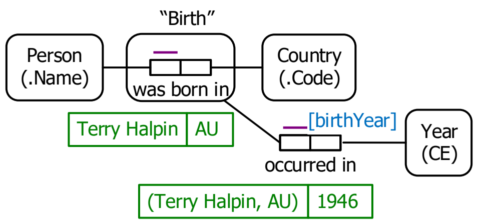
Case (1): Objectifying an *n*:1 fact type.

Since there is only one UC on the fact type, it is automatically chosen as the preferred UC for the objectification. The NORMA diagram on the left is displayed in NORMA with link fact types on the right (the default link fact type readings shown here would be better rephrased, e.g. "Birth is of/had Person").





When populating the "Birth occurred in Year" fact type, NORMA requires entries for Birth to include both the Person and Country, e.g. Birth ('Terry Halpin', 'AU') occurred in Year 1946. But that entry identifies the Birth *fact*, not the Birth *object* (which is simply identified by the person 'Terry Halpin'). Using Visio, this populated model may be displayed as shown.



Looking at the NORMA's linked fact type display, I would expect to use just the person's name when populating the birth year fact type. As when populating all other fact types, I believe each instance in a non-referential role population should use just the preferred identifier of the object playing the role. Without seeing the linked fact type display, some users might find it confusing to use just the person name,. If they use the person and country to identify the birth object, they are *populating the birth year fact type with a non-atomic fact*. I think it is fundamentally wrong to encourage users to use non-atomic facts in fact populations.

For example, "Birth ('Terry Halpin', 'AU') occurred in Year 1946" is basically shorthand for:

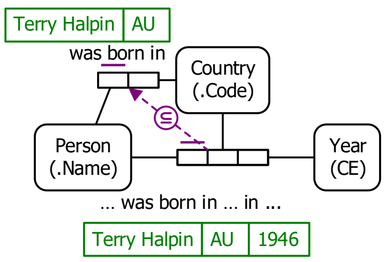
The Birth that involves the Person who has the PersonName 'Terry Halpin' **and** that involves the Country that has the CountryCode 'AU' occurred in the Year 1946 CE.

Clearly this is a conjunction of two elementary facts:

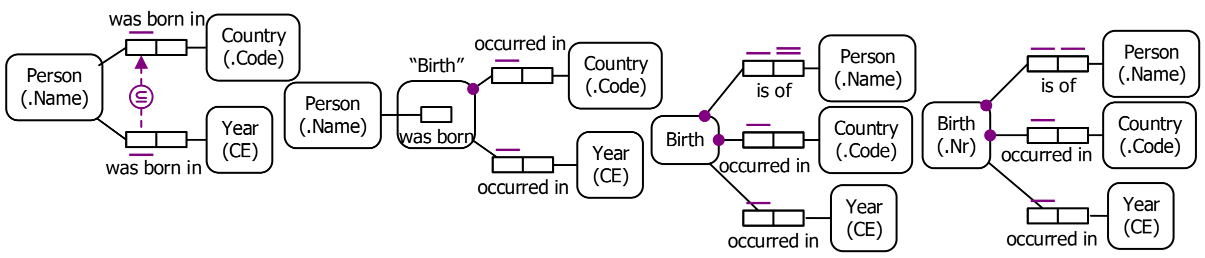
The Birth that involves the Person who has the PersonName 'Terry Halpin' involves the Country that has the CountryCode 'AU'.

The Birth that involves the Person who has the PersonName 'Terry Halpin' occurred in the Year 1946 CE.

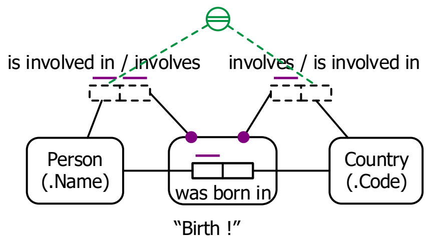
In flattened form, the populated model may be displayed thus, revealing that when populated in this way the birth age fact type is effectively a ternary, with a single role UC. This violates the *n*-1 rule, so is clearly not atomic. Since this model is illegal in ORM, I believe the nested version shown earlier should also be treated as illegal.



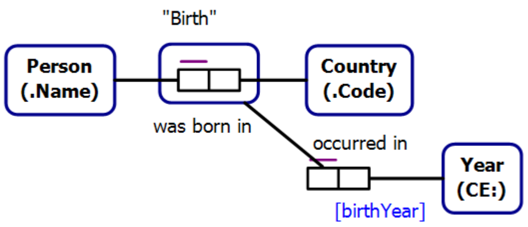
If Birth hosts no other roles, I prefer the left-most schema below. If Birth hosts multiple roles, it's better to use one of the other 3 schemas. Add role names (e.g. birthyear, birthCountry) as appropriate.

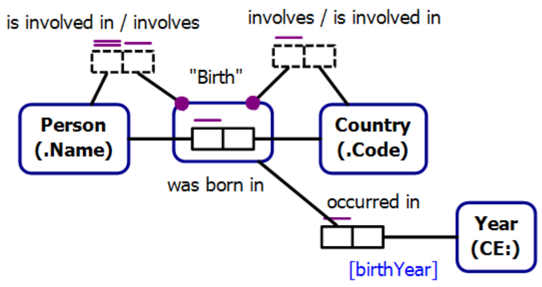


Using the whole fact entry to populate Birth in the birth age fact type suggested to me that the user naturally thought that Birth's identifier is based on the implied external UC shown. This would also allow Birth to be declared independent. But my earlier proposal to actually consider this idea was clearly wrong, as it violates elementarity.

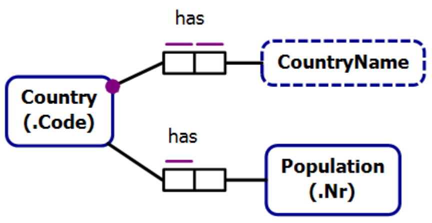


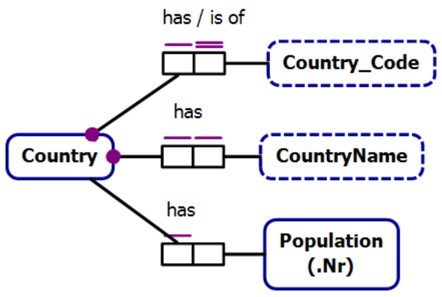
If the link fact types are not displayed, some users might think that the birth year role hosted by Birth is implied to be mandatory, since Birth has not been declared independent. However, once the link fact types are displayed, it is clear that Birth is not independent since its non-referential role with Country is mandatory. So Birth's role in the birth year fact type is truly optional. This potential confusion would be avoided if objectification of an *n*:1 fact type is not allowed.





If the link fact types are displayed, I think that most users would expect to use the preferred identifier for Birth (using just the person's name) when populating Birth in the birth year fact type. For example, consider the Country schemas below, with Country's reference scheme shown in both compact and expanded form. When populating Country in its population fact type, NORMA expects just the country code. I regard this situation as analogous the Birth schema shown on the right above. The only difference is that NORMA displays the link fact types with dashed lines. However, in terms of formal semantics these "link fact types" are just normal fact types. Hence I would expect the birth year fact type to be populated just like the country population fact type (using just the simple preferred identifier for any Birth instance).

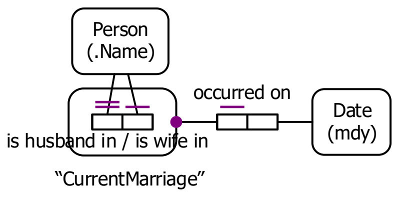


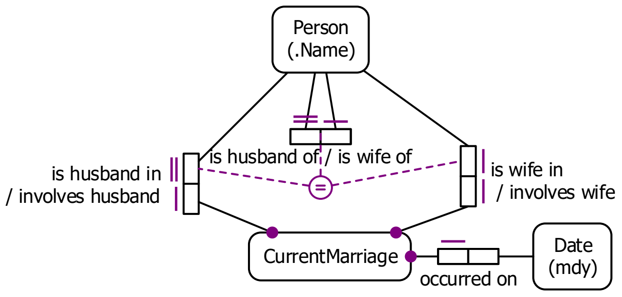


Case (2): Objectifying a 1:1 fact type.

The only reason ORM was relaxed to allow objectification of 1:1 fact types was to avoid forcing the user to choose which of the two UCs on the original fact type is used as the basis for the preferred reference scheme of the objectification object type. Such a choice is typically not a conceptual issue, but is needed when mapping to a relational schema. NORMA actually requires the choice to be made as soon as the 1:1 fact type is objectified. I now agree with NORMA in this regard, mainly to ensure that all objects are consistently identified by their preferred reference scheme, and that an unambiguous formal semantics of objectification can be provided without violating atomicity.

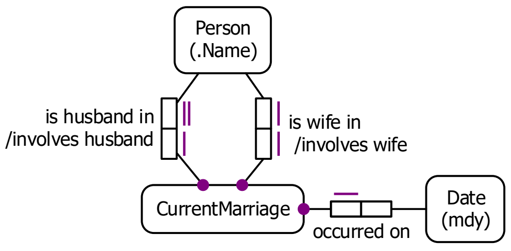
In the following example, the husband role is chosen as the basis for the preferred reference scheme of CurrentMarriage. After renaming the default "involves/is involves in" predicate readings for the link fact types, the formal semantics of this objectification may be displayed as shown on the right. NORMA instead uses dashed lines for the link fact types, and ignores the pair equality constraint that sets up the 1:1 correspondence for the formal equivalence. I'm not asking that NORMA use the display shown on the right, as its mirror role implementation does not require this.

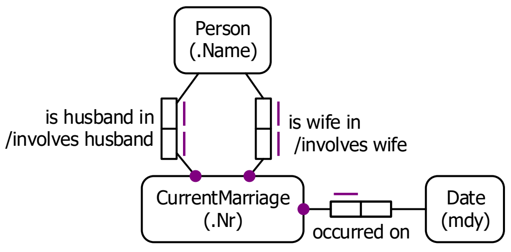




When populating the marriage date fact type, NORMA requires both the husband and wife to identify the marriage, so the fact entry is not atomic. For similar reasons discussed for the *n*:1 case, I believe that instances of CurrentMarriage in the marriage date fact type should be identified simply by the husband.

To avoid confusion and avoid non-atomic fact entries, the schema on the left below provides a better solution. Alternatively, choose a marriage number (e.g. marriage certificate number) to identify a marriage. (see schema on the below right).

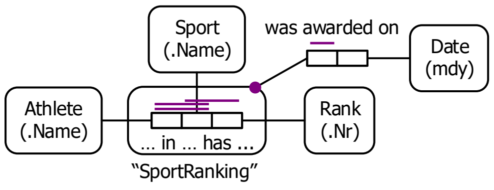


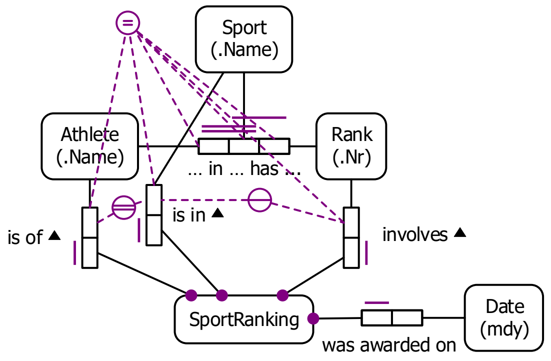


Case (3): Objectifying an *n*-ary (ternary or longer) fact type with multiple UCs spanning *n*-1 roles.

The only reason ORM was relaxed to allow objectification of *n*-ary fact types was to avoid forcing the user to choose which of the multiple UCs on the original fact type is used as the basis for the preferred reference scheme of the objectification object type. Such a choice is often not a conceptual issue, but is needed when mapping to a relational schema. NORMA actually requires the choice to be made as soon as the 1:1 fact type is objectified. I now agree with NORMA in this regard, mainly to ensure that all objects are consistently identified by their preferred reference scheme, and that an unambiguous formal semantics of objectification can be provided without violating atomicity.

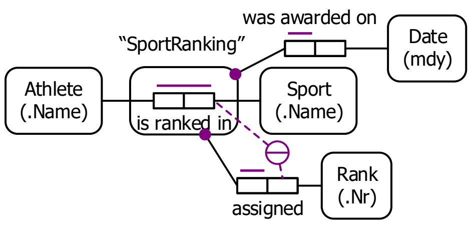
The ternary fact type in the following example has two overlapping UCs, the first of which has been chosen as the basis for the preferred reference scheme for the objectification object type SportRanking. After renaming the default "involves/is involves in" predicate readings for the link fact types, the formal semantics of this objectification may be displayed as shown on the right. NORMA instead uses dashed lines for the link fact types, and ignores the triple equality constraint that sets up the 1:1 correspondence for the formal equivalence. I'm not asking that NORMA use the display shown on the right, as its mirror role implementation does not require this.

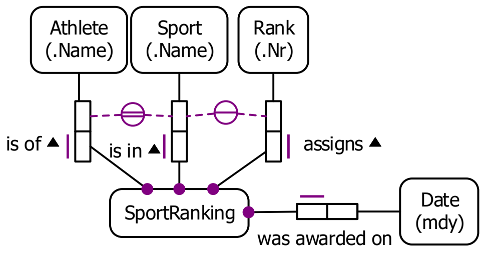




When populating the award date fact type, NORMA requires the athlete, sport and rank to identify the sport ranking, so the fact entry is not atomic. For similar reasons discussed for the *n*:1 and 1:1 case, I believe that instances of SportRanking in the award date fact type should be identified simply by the athlete and sport.

To avoid confusion and avoid non-atomic fact entries, either of the schemas shown provides a better solution.





To sum up, although a formal semantics can be provided for objectification of fact types without a spanning uniqueness constraint, for the six reasons cite on page 1 it's better not to allow this kind of objectification