

An overly optimistic (or pessimistic, depending on which way you look at it) account of the future of artificial intelligence can be found in Kurzweil (1999, 2005). A more conservative account of the expected future of computing over the next 40 years is given in Denning and Metcalfe (1997).

ORM Glossary

This glossary lists key terms and symbols used in Object-Role Modeling (ORM), and briefly explains their meaning. A concise explanation of other technical terms may be found in the chapter summaries. Further details on technical terms may be accessed by using the Index.

Alethic constraint: Constraint that holds necessarily for all states of the model.

Arity: Number of roles in a relationship (unary = 1, binary = 2, ternary = 3, etc.).

Asserted fact: Fact that is simply asserted, rather than being derived from others; also called a primitive fact or base fact.

Asserted subtype: Subtype that is simply asserted (not defined by a subtype definition).

Association: Relationship type, usually involving at least two roles.

Atomic fact: Either an elementary fact or an existential fact.

Base fact: Fact that is primitive (not derived from others). Also called an asserted fact.

Compound fact type: Fact type that is equivalent to a conjunction of smaller fact types

Conceptual schema: Conceptual model of the UoD structure; design that specifies what states and transitions are possible; declaration of fact types, constraints, and derivation rules

Conceptual schema design procedure (CSDP):

- 0 Divide the UoD into manageable sub-sections
- 1 Transform familiar examples into elementary facts, and apply quality checks
- 2 Draw the fact types, and apply a population check
- 3 Check for entity types that should be combined, and note arithmetic derivations
- 4 Add uniqueness constraints, and check arity of fact types
- 5 Add mandatory role constraints, and check for logical derivations
- 6 Add value, set comparison (subset, equality, exclusion) and subtype constraints
- 7 Add other constraints and perform final checks
- 8 Integrate the subschemas into a global conceptual schema

Constraint: Restriction on possible or permissible states (static constraint) or transitions (dynamic constraint).

Compositely identified object type: Either a coreferenced or a nested object type.

Coreferenced object: Object that is identified by means of two or more reference types in combination; hence its identification scheme involves an external uniqueness constraint.

Database: Variable set of related fact instances.

Deontic rule: An obligation, i.e. a rule that ought to be obeyed (but possibly may be violated).

Derivation rule: Rule that declares how one fact type may be derived from others.

Derived fact: Fact that is derived from other fact types using a derivation rule

Derived subtype: Subtype that is derived from other object types using a subtype definition.

Elementary fact: Assertion that an object has a property, or that one or more objects participate in a relationship, where the fact cannot be split into simpler facts with the same object types without information loss. Application of an atomic predicate to a sequence of objects.

Entity: Object that is referenced by relating it to other objects (e.g., the Country that has CountryCode 'AU'); not a value; typically, an entity may undergo changes over time; an entity is either atomic or nested (i.e. an objectified relationship); at the top level, entities are partitioned into primitive entity types, from which subtypes may be defined.

Existential fact: Assertion that an object exists (e.g., there is a Country that has CountryCode 'AU'); also called a reference.

Fact: Proposition that is taken to be true by the relevant business community, where the proposition is elementary or existential (rather than being a constraint or derivation rule).

Fact role: Role in an elementary fact type.

Fact type: Kind of fact, including object terms and either a predicate or existential quantifier.

Flatten: Restate without nesting.

Functional fact type: Fact type with a functional role.

Functional role: Role with a simple uniqueness constraint.

Generalization: Forming a more general case from one or more specific cases; the inverse of specialization.

Independent object: Object that may exist without participating in any elementary fact; the disjunction of fact roles played by an independent object type is optional.

Instance: An individual occurrence (one specific member of a type).

Mandatory role: Role that must be played by all instances in the population of the object type playing the role; also called a total role.

Modality: Mode in which a proposition is expressed. In ORM 2, modalities are either alethic (expressing necessities or possibilities) or deontic (expressing obligations or permissions).

Nested object: Relationship that plays some role (also called an objectified relationship).

Object: Thing of interest; an object may be an entity or a value.

Objectification: Treating a relationship as an object; also called nesting. Strictly speaking, objectification in ORM 2 distinguishes the object formed by the objectification from the original relationship, and hence involves situational rather than propositional nominalization.

Object-Role Modeling (ORM): Conceptual modeling method that pictures a business domain in terms of objects playing roles; it provides graphical and textual languages for verbalizing and querying information as well as various design and transformation procedures.

Population: Set of instances present in a particular state of the database.

Predicate: Proposition with object-holes in it, e.g. "... works for ...".

Reference: Relationship used as the preferred way to reference or identify an object (or to provide part of the identification).

Reference mode: Mode or manner in which a single value references an entity; used to abbreviate simple reference schemes, e.g. (.code), (kg:).

Reference role: Role in a reference (existential fact).

Relationship: Property or association involving one or more objects.

Rigid subtype: Subtype whose instances must remain in that type for their whole lifetime (e.g. Person).

Rmap: Relational mapping procedure.

Role: Part played by an object in a relationship (unary, binary, ternary, etc.).

Role subtype: Subtype whose instances may leave that type during their lifetime (e.g., Child).

Semiderived fact type: Fact type. some of whose instances may be derived from others, while some other instances may be simply asserted.

Semiderived subtype: Subtype, some of whose instances may be derived using a derivation rule while some other instances may be simply asserted.

Subtype: Object type that is properly contained in another object type (e.g., Woman is a subtype of Person).

Type: Set of possible instances.

Uniqueness constraint (UC): Repetition is not allowed in the role or role sequence spanned by the constraint; a uniqueness constraint on a single predicate is an internal UC, and a uniqueness constraint over roles from different predicates is an external UC.

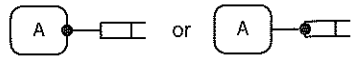
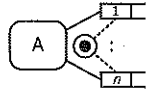
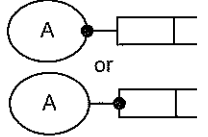
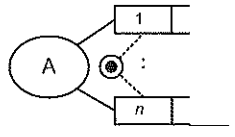
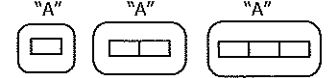
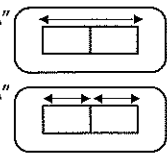

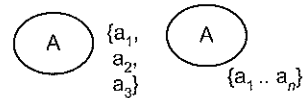
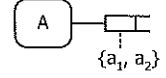
Universe of Discourse (UoD): Business domain (the aspects of the world that we want to talk about).


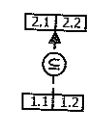
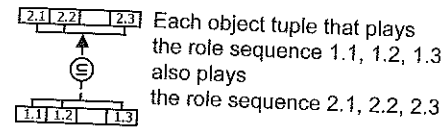
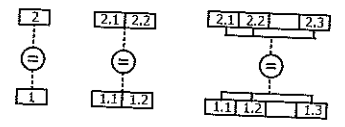
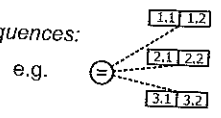
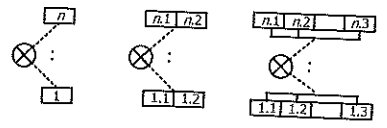
Value: Unchangeable object that is identified by a constant; in this book a value is either a character string or a number; sometimes called a label.

The following symbol glossary covers the main graphical symbols in ORM 2 (as supported by the NORMA tool) and the corresponding symbols in ORM 1 (as supported in Microsoft Visio for Enterprise Architects).

ORM 2	ORM 1
<p>Object Types</p> <p>Entity type A (first shape is the default) From now on, we show only the default shape.</p> <p>A identified by reference mode Ref</p> <p>Value type A</p> <p>Independent entity type A</p> <p>Independent value type A</p> <p>Duplicated object types</p> <p>External object type A</p>	<p>A</p> <p>A (Ref)</p> <p>A</p> <p>A!</p> <p>Not supported.</p> <p>Not supported.</p> <p>A</p>
<p>Predicates</p> <p>Unary: R</p> <p>Binary: R/S, R/S, R/S</p> <p>Ternary: R</p> <p>Quaternary: R</p> <p>etc. (n role-boxes for n-ary predicate)</p>	<p>Predicates are basically the same, except that role boxes are larger</p>
<p>Role names</p> <p>[role1] [role2]</p>	<p>Role names may be entered but are not displayed</p>



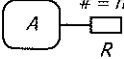
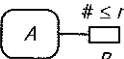
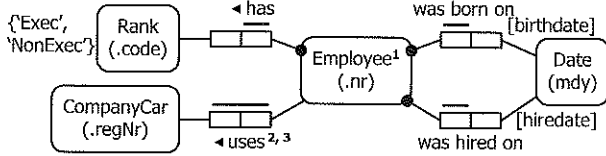
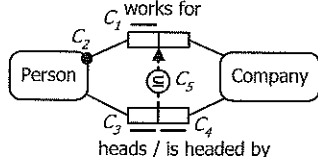
ORM 2	ORM 1
<p>Internal Uniqueness Constraints</p> <p>Unary: </p> <p>Binary: n:1, 1:n, 1:1, m:n</p> <p>UC on role pair 1-2</p> <p>UC on role pair 1, 3</p> <p>Many UC combinations are possible</p> <p>Preferred uniqueness:</p> <p>ORM 2 enables display of preferred uniqueness constraints on n-aries to be toggled on/off.</p>	<p></p> <p></p> <p></p> <p></p> <p></p>
<p>Role Connection</p> <p>A Role is played only by A</p>	<p></p>
<p>External Uniqueness Constraints</p> <p></p> <p>Each B₁, ..., B_n combination (n > 1) relates to only one instance of A</p> <p>Preferred uniqueness:</p> <p></p>	<p></p> <p></p>

ORM 2	ORM 1
<p>Mandatory Role Constraints</p> <p><i>Simple:</i></p>  <p>Role is mandatory for population of A</p> <p><i>Disjunctive (inclusive-or constraint):</i></p>  <p>Each instance in the population of A plays at least one of the n attached roles (n > 1). Role numbers are not displayed.</p>	 
<p>Objectification</p>  <p>etc.</p> <p>Fact type is objectified as object type A. ORM 2 allows any fact type to be objectified.</p>	<p>ORM 1 does not support objectified unaries. It allows objectification only if for a spanning UC or a 1:1 pattern.</p> 
<p>Object Value Constraints</p> <p><i>Enumeration</i> <i>Range</i></p>  <p><i>Semibounded discrete range</i> { a.. } { ..a }</p> <p><i>Bounded continuous range</i></p> <ul style="list-style-type: none"> {[a1 .. a2]} includes both end values {(a1 .. a2)} excludes both end values {[a1 .. a2)} includes first value {(a1 .. a2]} includes last value <p>Combinations are allowed.</p>	 <p>{ a.. } { ..a }</p> <p>{ a1..a2 }</p> <p>ORM1 does not support exclusion of any end values</p> <p>Combinations are allowed.</p>
<p>Role Value Constraints</p>  <p>Same patterns as above</p>	<p>Not supported</p>

ORM 2	ORM 1
<p>Subset Constraints</p> <p><i>Simple:</i></p>  <p>Each object that plays role 1 also plays role 2</p> <p><i>Contiguous Role-pair:</i></p>  <p>Each object pair that plays the role sequence 1.1, 1.2 also plays the role sequence 2.1, 2.2</p> <p><i>Other cases:</i></p>  <p>Each object tuple that plays the role sequence 1.1, 1.2, 1.3 also plays the role sequence 2.1, 2.2, 2.3</p> <p>ORM 2 also displays subset constraints over join paths</p>	<p>Same</p> <p>Same</p> <p>Same</p> <p>ORM 1 does not display subset constraints over join paths</p>
<p>Equality Constraints</p> <p><i>2 role-sequences (of 1 or more roles):</i></p>  <p>Populations of role-sequences must be equal</p> <p><i>3 or more role-sequences:</i></p>  <p>e.g.</p>	<p>Same</p> <p>Not supported (instead use multiple 2-sequence constraints)</p>
<p>Exclusion Constraints</p>  <p>Populations of 2 or more role-sequences must be mutually exclusive</p>	<p>Same</p>

ORM 2	ORM 1												
<p>Exclusive-Or Constraints</p> <p>Each instance in A's population plays exactly one of the n attached roles ($n > 1$)</p>	Same												
<p>Subtyping</p> <p>B is a proper subtype of A and C. A (but not C) provides a path to the preferred identifier for A.</p> <p style="text-align: center;">Exclusive Total Partition</p>	<p>B is a proper subtype of A (its primary supertype) and C (a secondary supertype)</p> <p>ORM 1 does not display constraints over subtyping connections</p>												
<p>Frequency Constraints</p> <p>Each instance that plays role 1 does so f times</p> <p>Each instance pair that plays roles 1, 2 does so f times</p> <p>Each instance pair that plays roles 1, 2 does so f times</p> <p>The frequency specification f may be any of the following</p> <table border="0"> <tr> <td>n</td> <td>exactly n (a positive integer)</td> <td>n</td> </tr> <tr> <td>$\geq n$</td> <td>at least n</td> <td>$\geq n$</td> </tr> <tr> <td>$\leq n$</td> <td>at most n</td> <td>$\leq n$</td> </tr> <tr> <td>$n..m$</td> <td>at least n and at most m</td> <td>$n..m$</td> </tr> </table>	n	exactly n (a positive integer)	n	$\geq n$	at least n	$\geq n$	$\leq n$	at most n	$\leq n$	$n..m$	at least n and at most m	$n..m$	Same
n	exactly n (a positive integer)	n											
$\geq n$	at least n	$\geq n$											
$\leq n$	at most n	$\leq n$											
$n..m$	at least n and at most m	$n..m$											

ORM 2	ORM 1
<p>Ring Constraints</p> <p>Irreflexive</p> <p>Asymmetric</p> <p>Intransitive</p> <p>Antisymmetric</p> <p>Acyclic</p> <p>Asymmetric + Intransitive</p> <p>Acyclic + Intransitive</p> <p>Symmetric</p> <p>Symmetric + Irreflexive</p> <p>Symmetric + Intransitive</p> <p>Purely Reflexive</p>	<p>o_{as}</p> <p>o_{it}</p> <p>o_{ans}</p> <p>o_{ac}</p> <p>$o_{(as,it)}$</p> <p>$o_{(ac,it)}$</p> <p>o_{sym}</p> <p>$o_{(ir,sym)}$</p> <p>$o_{(it,sym)}$</p> <p>Not supported</p>
<p>Value-comparison Constraints</p>	Not supported
<p>Derived Fact Types</p> <p>* = derived, ** = derived and stored + = semi-derived</p>	Same for first two options. 3 rd option not supported.
<p>Deontic Constraints</p> <p>Colored blue rather than violet. Most include "o" for "obligatory". Deontic ring constraints instead use dashed lines.</p> <p>Uniqueness \ominus</p> <p>Mandatory \odot</p> <p>Subset, Equality, Exclusion \subseteq \equiv \otimes</p> <p>Frequency o_f</p> <p>Irreflexive \ominus Acyclic recycle</p> <p>Asymmetric \ominus Asym-Intrans asym-intrans</p> <p>Intransitive intrans Acyclic-Intrans acyclic-intrans</p> <p>Antisymmetric antisym Symmetric sym</p> <p>Purely Reflexive purely reflexive etc.</p>	No deontic constraints are supported

ORM 2	ORM 1
<p>Object Cardinality Constraints</p> <p>$\# = n$  Each population of <i>A</i> includes exactly <i>n</i> instances</p> <p>$\# \leq n$  Each population of <i>A</i> includes at most <i>n</i> instances</p>	Not supported
<p>Role Cardinality Constraints</p> <p> $\# = n$ Each population of <i>R</i> includes exactly <i>n</i> instances</p> <p> $\# \leq n$ Each population of <i>R</i> includes at most <i>n</i> instances</p>	Not supported
<p>Textual Constraints (ORM 2 example)</p>  <p>¹ For each Employee, birthdate < hiredate. ² Each Employee who has Rank 'NonExec' uses at most one CompanyCar. ³ Each Employee who has Rank 'Exec' uses some CompanyCar.</p>	
<p>Constraint Verbalization (ORM 2 example)</p>  <p><i>C</i>₁: Each Person works for at most one Company. <i>C</i>₂: Each Person works for some Company. <i>C</i>₃: Each Person heads at most one Company. <i>C</i>₄: Each Company is headed by at most one Person. <i>C</i>₅: Each Person who heads some Company also works for that Company.</p> <p>The absence of a UC on the top righthand role verbalizes as It is possible that more than one Person works for the same Company.</p>	