*Note: The text below is intended to replace the third and all subsequent paragraphs of section 7.2.1*

The normative, MOF-compliant XMI can be generated from the model represented in the class diagrams of this specification in the following way:

7.2.1.1 Un-stereotyped class translation

An un-stereotyped class in a diagram becomes a class in MOF. When such a class inherits from “BusinessElement”, a generalization association is added to specialize the “BACMPlainEntity” abstract class. These specializations are presented in diagram 7.3.1.3.

7.2.1.2 Un-stereotyped association translation

An un-stereotyped association between classes that specialize “BusinessElement” in a diagram becomes an association in MOF, except when such an association is a leg association of an <<association>> stereotyped class – see below. All other associations are translated into MOF associations.

7.2.1.3 <<class>> stereotyped binary association translation

A binary, directed association in a diagram with a <<class>> stereotype is translated into a MOF class and two binary MOF associations. Navigability is ignored and the implementation must provide bidirectional navigation for both the generated, binary, directed MOF associations. The MOF class represents a relationship and the two associations specify the types of elements that can participate in the relationship. By convention, the cardinality of the association end opposite the MOF class is 0..1, representing the notion that instances of the MOF class contain single valued properties (the owned ends) that reference a single instance of the defined type. The cardinality of the association end at the MOF class is the cardinality of the origin binary association. Since there are two associations, each one represents an end of the origin association. To preserve directionality, a naming convention is used; the name of the MOF association representing the starting association end is prefixed with “from\_” and the name of the MOF association representing the ending association end is prefixed with “to\_”. The MOF class is given the name of the origin association. The generated ownedEnds resulting from this translation are given names that are the names of the MOF association prefixed with “src\_” and “dst\_” respectively to preserve the directionality of these associations (from “src” to “dst”). Thus the origin association “produces” (see Diagram 7.3.2.1) is translated into a MOF class named “produces” and two associations: “from\_produces” with ownedEnds “src\_from\_produces” and “dst\_from\_produces”, and “to\_produces” with ownedEnds “src\_to\_produces” and “dst\_to\_produces”.

The generated MOF class specializes the “BACMBinDirRelation” abstract class and redefines the ownedEnds: “from\_bacm\_entity”, “from\_bacm\_relation”, “to\_bacm\_entity” and “”to\_bacm\_relation”. This specialization and the redefinitions are created in the translation and are not shown or described in this document. The specialization permits MOF reflection to distinguish binary directed relationship instances from other types of instances.

The lifecycle semantics of the configuration of MOF class and MOF associations is equivalent to the lifecycle of the origin association. In particular, if an instance coupled to the “dst\_” prefixed ownedEnd is deleted, then the corresponding instance of the class and the other association instance must also be deleted from the model.

7.2.1.4 <<association>> stereotyped origin class translation

A class in a diagram with an <<association>> stereotype is translated into a MOF class and each un-stereotyped association whose starting ownedEnd is at this class is effectively a component of an n-ary relationship that is represented by the class. The term “leg” is used in this document section to refer to such associations. Note that this allows an origin <<association>> class to participate in other associations where it is the “dst” of such an association that is often stereotyped with <<class>>. Such a configuration (i.e. an <<association>> stereotyped class and some number of leg associations) is translated directly into MOF as a MOF class and MOF associations, but without the <<association>> stereotype on the class. In this case, the class and leg association names remain unchanged in the MOF metamodel as do the ownedEnd names. The ownedEnd cardinalities are also directly translated into MOF.

In the MOF translation, the MOF class specializes “BACMRelation”, allowing MOF reflection to distinguish that instances of the MOF class represent n-ary relations and to identify the associations that represent legs of the relationship.

The lifecycle semantics of the configuration of instances of such a MOF class and instances of its leg associations obey the same rule as for the translated binary directed associations with the <<class>> stereotype. If an participating instance that is referenced by the “dst” slot of the link instance of a leg association is deleted from the model, then the class instance and all other leg association link instances must be deleted from the model.

7.2.1.5 <<shortcut>> stereotyped class and association translation

The basic translation is as if the class was stereotyped <<association>> or the association was stereotyped <<class>>. In addition, the MOF class representing the <<class>> stereotyped association or the <<association>> stereotyped class carries an ownedRule represented by an OpaqueExpression that is specified both in OCL and in a path language whose interpretation is equivalent to the OCL representation. This OCL expression may be invoked by the business architecture modeler to determine if additional model information is available that implies the existence of an instance of the <<shortcut>> element. The shortcut mechanism is described in section 8.1 of this specification. The language type of the alternative path expression is “BACMPathLanguage” and is intended for implementors who are based on RDF (the expression can be directly translated into a SPARQL query) or a graph language.

7.2.1.6 <<individual>> stereotype translation

The specification has a single metaclass, *theBusness* with this stereotype. The concept represents the particular business being modeled and its purpose is to designate *Performers* that *belong\_to* this business (i.e. are employees or contract workers). There should be at most one instance of this metaclass in a model. In translation to MOF, the stereotype is removed and an OCL constraint is added that at most one instance has *theBusiness* as its metaclass,