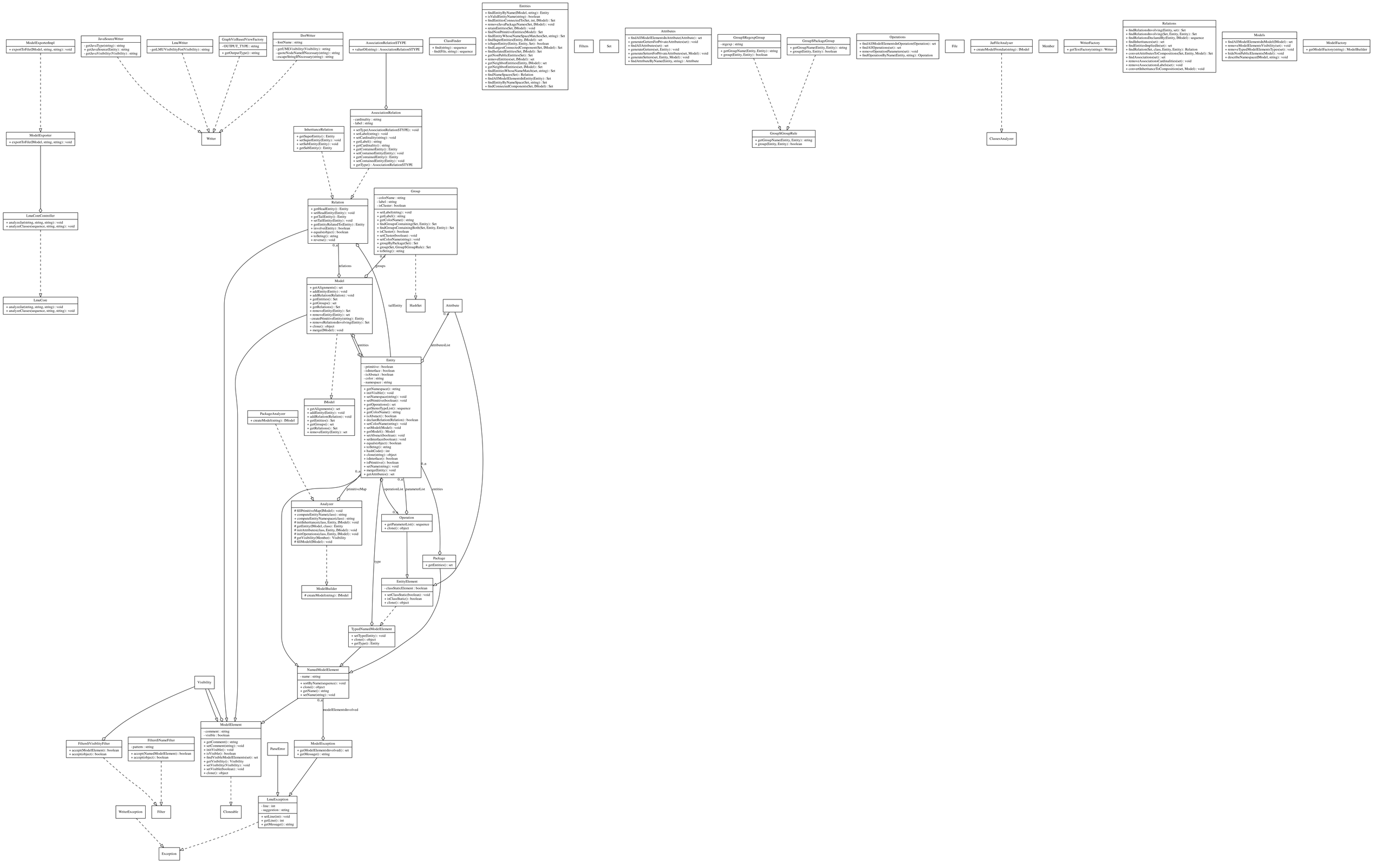
ModelExporterImpl
+ exportToFile(IModel, string, string) : void
JavaSourceWriter
- getJavaType(string) : string
+ getJavaSource(Entity) : string
- getJavaVisibilityVisibility() : string
LmuWriter
- getLMUVisibilityForVisibility() : string
GraphVizBasedViewFactory
- OUTPUT_TYPE : string
+ getOutputType() : string
DotWriter
- fontName : string
- getLMUVisibilityVisibility() : string
- quoteNodeNameIfNecessary(string) : string
- escapeString(INecessary(string) : string
AssociationRelation$TYPE
+ valueOf(string) : AssociationRelation$TYPE
ClassFinder
+ find(string) : sequence
- findFile, string) : sequence
Entities
+ findEntityByNamed(IModel, string) : Entity
+ isValidEntityName(string) : boolean
+ findEntitiesConnectedToSet, int, IModel) : Set
+ removeJavaPackageNameSet, IModel) : void
+ retainEntitiesSet, IModel) : void
+ findNonPrimitiveEntitiesIModel) : Set
+ findEntityWhoseNameSpaceMatches(Set, string) : Set
+ findSuperEntitiesEntity, IModel) : set
+ isSuperEntity(Entity, Entity, Set) : boolean
+ findLargestConnectedComponent(Set, IModel) : Set
+ findIsolatedEntities(Set, IModel) : Set
+ getNonPublicEntities(Set) : Set
+ removeEntities(set, IModel) : set
+ getNeighborEntities(Entity, IModel) : set
+ getNeighborEntities(set, IModel) : Set
+ findEntitiesWhoseNameMatch(set, string) : Set
+ findNameSpaces(Set) : Relation
+ findAllModelElementsInEntity(Entity) : Set
+ findEntityByNameSpace(Set, string) : Set
+ findConnectedComponents(Set, IModel) : Set
Filters
Set
Attributes
+ findAllModelElementsInAttribute(Attribute) : set
+ generateGettersForPrivateAttributes(set) : void
+ findAllAttributes(set) : set
+ generateGetters(set, Entity) : void
+ generateSettersForPrivateAttributes(set, Model) : void
+ generateSetters(set, Entity, Model) : void
+ findAttributeByNameEntity, string) : Attribute
Group$RegexpGroup
- regexp : string
+ getGroupNameEntity, Entity) : string
+ groupsEntity, Entity) : boolean
Group$PackageGroup
+ getGroupNameEntity, Entity) : string
+ groupsEntity, Entity) : boolean
Operations
+ findAllModelElementsInOperation(Operation) : set
+ findAllOperations(set) : set
+ removeOperationParameters(set) : void
+ findOperationByNameEntity, string) : Operation
File
JarFileAnalyser
+ createModelFromJars(string) : IModel
Member
WriterFactory
+ getTextFactory(string) : Writer
Relations
+ findRelationsInvolving(Entity, set) : Set
+ findRelationsInvolving(Set, Entity, Entity) : Set
+ findRelationsDeclaredBy(Entity, IModel) : sequence
+ findInheritances(set) : set
+ findEntitiesImplicitIn(set) : set
+ findRelation(Set, class, Entity, Entity) : Relation
+ convertAttributesToCompositions(Set, Entity, Model) : Set
+ findAssociations(set) : set
+ removeAssociationsCardinalities(set) : void
+ removeAssociationsLabels(set) : void
+ convertInheritanceToComposition(set, Model) : void
Models
+ findAllModelElementsIsModel(IModel) : set
+ removeModelElementsVisibility(set) : void
+ removeTypedModelElementsTypes(set) : void
+ hideNonPublicElementsModel) : void
+ describeNamespaces(IModel, string) : void
ModelFactory
+ getModelFactory(string) : ModelBuilder
ModelExporter
+ exportToFile(IModel, string, string) : void
Writer
LmuCoreController
+ analyzeJar(string, string, string) : void
+ analyzeClasses(sequence, string, string) : void
LmuCore
+ analyzeJar(string, string) : void
+ analyzeClasses(sequence, string, string) : void
InheritanceRelation
+ getSuperEntity() : Entity
+ setSuperEntity(Entity) : void
+ setSubEntity(Entity) : void
+ getSubEntity() : Entity
AssociationRelation
- cardinality : string
- label : string
+ setType(AssociationRelation$TYPE) : void
+ setLabel(string) : void
+ setCardinality(string) : void
+ getLabel() : string
+ getCardinality() : string
+ getContainerEntity() : Entity
+ setContainerEntity(Entity) : void
+ getContainedEntity() : Entity
+ setContainedEntity(Entity) : void
+ getType() : AssociationRelation$TYPE
Relation
+ getHeadEntity() : Entity
+ setHeadEntity(Entity) : void
+ getTailEntity() : Entity
+ setTailEntity(Entity) : void
+ getEntityRelatedToEntity() : Entity
+ involveEntity() : boolean
+ equals(object) : boolean
+ toString() : string
+ reverse() : void
Group
- coloreName : string
- label : string
- isCluster : boolean
+ setLabel(string) : void
+ getLabel() : string
+ getColoreName() : string
+ findGroupsContaining(Set, Entity) : Set
+ findGroupsContainingBoth(Set, Entity, Entity) : Set
+ isCluster() : boolean
+ setCluster(boolean) : void
+ setColoreName(string) : void
+ groupByPackage(Set) : Set
+ groups(Set, Group$GroupRule) : Set
+ toString() : string
HashSet
Model
+ getAlignments() : set
+ addEntity(Entity) : void
+ addRelation(Relation) : void
+ getEntities() : Set
+ getGroups() : set
+ getRelations() : Set
+ removeEntity(Entity) : Set
+ removeEntity(Entity) : set
- createPrimitiveEntity(string) : Entity
+ removeRelationsInvolving(Entity) : Set
+ clone() : object
+ merge(IModel) : void
Attribute
IModel
+ getAlignments() : set
+ addEntity(Entity) : void
+ addRelation(Relation) : void
+ getEntities() : Set
+ getGroups() : set
+ getRelations() : Set
+ removeEntity(Entity) : set
PackageAnalyzer
+ createModel(string) : IModel
Entity
- primitive : boolean
- isInterface : boolean
- isAbstract : boolean
- color : string
- namespace : string
+ getNamespace() : string
+ initVisible() : void
+ setNamespace(string) : void
+ setPrimitive(boolean) : void
+ getOperations() : set
+ getStereoTypeList() : sequence
+ getColorName() : string
+ isAbstract() : boolean
+ declareRelationRelation() : boolean
+ setColorName(string) : void
+ setModel(IModel) : void
+ getModel() : Model
+ setAbstract(boolean) : void
+ setInterface(boolean) : void
+ equals(object) : boolean
+ toString() : string
+ hashCode() : int
+ clone(string) : object
+ isInterface() : boolean
+ isPrimitive() : boolean
+ setName(string) : void
+ mergeEntity() : void
+ getAttributes() : set
Analyzer
fillPrimitiveMap(IModel) : void
+ computeEntityName(class) : string
+ computeEntityNamespace(class) : string
# initInheritance(class, Entity, IModel) : void
# getEntity(IModel, class) : Entity
# initAttributes(class, Entity, IModel) : void
# initOperations(class, Entity, IModel) : void
# getVisibility(Member) : Visibility
# fillModel(IModel) : void
Operation
+ getParameterList() : sequence
+ clone() : object
Package
+ getEntities() : set
ModelBuilder
createModel(string) : IModel
EntityElement
- classStaticElement : boolean
+ setClassStatic(boolean) : void
+ isClassStatic() : boolean
+ clone() : object
TypedNamedModelElement
+ setTyperEntity() : void
+ clone() : object
+ getTyper() : Entity
NamedModelElement
- name : string
+ sortByName(sequence) : void
+ clone() : object
+ getName() : string
+ setName(string) : void
Visibility
ModelElement
- comment : string
- visible : boolean
+ getComment() : string
+ setComment(string) : void
+ initVisible() : void
+ isVisible() : boolean
+ findVisibleModelElement(set) : set
+ getVisibility() : Visibility
+ setVisibility(Visibility) : void
+ setVisible(boolean) : void
+ clone() : object
FiltersVisibilityFilter
+ accept(IModelElement) : boolean
+ accept(object) : boolean
FiltersNameFilter
- pattern : string
+ accept(NamedModelElement) : boolean
+ accept(object) : boolean
ParseError
ModelException
+ getModelElementsInvolved() : set
+ getMessage() : string
WriterException
Filter
Cloneable
LmuException
- line : int
- suggestion : string
+ setLine(int) : void
+ getLine() : int
+ getMessage() : string
Exception
0..n
relations
groups
tailEntity
entities
attributesList
primitiveMap
operationList
parameterList
entities
type
modelElementsInvolved