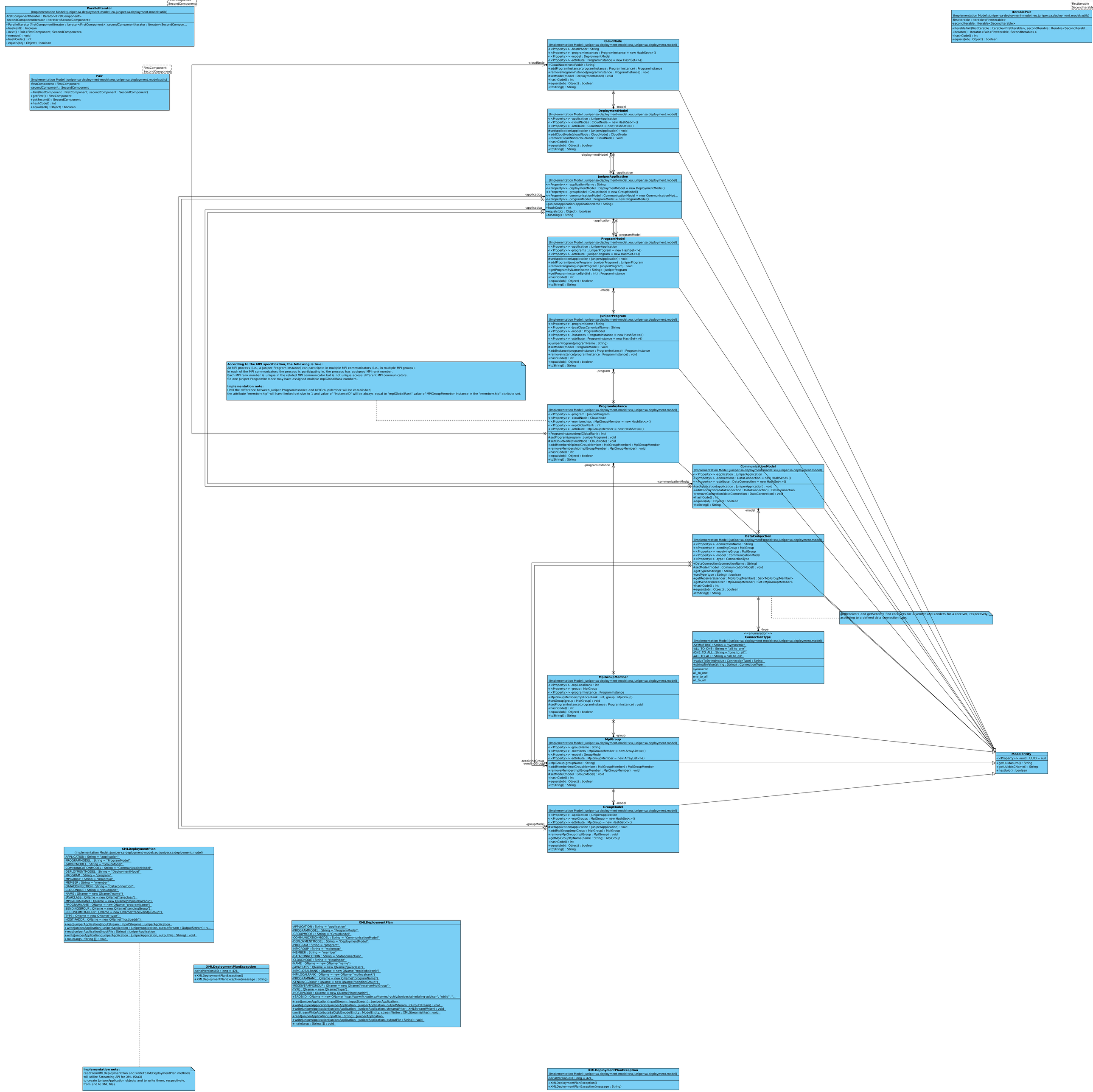
ParallelIterator
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model::utils)
-firstComponentIterator : Iterator<FirstComponent>
-secondComponentIterator : Iterator<SecondComponent>
+ParallelIterator(firstComponentIterator : Iterator<FirstComponent>, secondComponentIterator : Iterator<SecondCompon...
+hasNext() : boolean
+next() : Pair<FirstComponent, SecondComponent>
+remove() : void
+hashCode() : int
+equals(obj : Object) : boolean
FirstComponent
SecondComponent
Pair
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model::utils)
-firstComponent : FirstComponent
-secondComponent : SecondComponent
~Pair(firstComponent : FirstComponent, secondComponent : SecondComponent)
+getFirst() : FirstComponent
+getSecond() : SecondComponent
+hashCode() : int
+equals(obj : Object) : boolean
FirstIterable
SecondIterable
IterablePair
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model::utils)
-firstIterable : Iterable<FirstIterable>
-secondIterable : Iterable<SecondIterable>
+IterablePair(firstIterable : Iterable<FirstIterable>, secondIterable : Iterable<SecondIterabl...
+iterator() : Iterator<Pair<FirstIterable, SecondIterable>>
+hashCode() : int
+equals(obj : Object) : boolean
CloudNode
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -hostIPAddr : String
<<Property>> -programInstances : ProgramInstance = new HashSet<>()
<<Property>> -model : DeploymentModel
<<Property>> -attribute : ProgramInstance = new HashSet<>()
+CloudNode(hostIPAddr : String)
+addProgramInstance(programInstance : ProgramInstance) : ProgramInstance
+removeProgramInstance(programInstance : ProgramInstance) : void
#setModel(model : DeploymentModel) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-cloudNode
-model
DeploymentModel
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -application : JuniperApplication
<<Property>> -cloudNodes : CloudNode = new HashSet<>()
<<Property>> -attribute : CloudNode = new HashSet<>()
#setApplication(application : JuniperApplication) : void
+addCloudNode(cloudNode : CloudNode) : CloudNode
+removeCloudNode(cloudNode : CloudNode) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-deploymentModel
-application
JuniperApplication
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -applicationName : String
<<Property>> -deploymentModel : DeploymentModel = new DeploymentModel()
<<Property>> -groupModel : GroupModel = new GroupModel()
<<Property>> -communicationModel : CommunicationModel = new CommunicationMod...
<<Property>> -programModel : ProgramModel = new ProgramModel()
+JuniperApplication(applicationName : String)
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-application
-application
-application
-programModel
ProgramModel
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -application : JuniperApplication
<<Property>> -programs : JuniperProgram = new HashSet<>()
<<Property>> -attribute : JuniperProgram = new HashSet<>()
#setApplication(application : JuniperApplication) : void
+addProgram(juniperProgram : JuniperProgram) : JuniperProgram
+removeProgram(juniperProgram : JuniperProgram) : void
+getProgramByName(name : String) : JuniperProgram
+getProgramInstanceById(id : int) : ProgramInstance
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-model
JuniperProgram
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -programName : String
<<Property>> -javaClassCanonicalName : String
<<Property>> -model : ProgramModel
<<Property>> -instances : ProgramInstance = new HashSet<>()
<<Property>> -attribute : ProgramInstance = new HashSet<>()
+JuniperProgram(programName : String)
#setModel(model : ProgramModel) : void
+addInstance(programInstance : ProgramInstance) : ProgramInstance
+removeInstance(programInstance : ProgramInstance) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
According to the MPI specification, the following is true:
An MPI process (i.e., a Juniper Program instance) can participate in multiple MPI communicators (i.e., in multiple MPI groups).
In each of the MPI communicators the process is participating in, the process has assigned MPI rank number.
Each MPI rank number is unique in the related MPI communicator but is not unique across different MPI communicators.
So one Juniper ProgramInstance may have assigned multiple mpiGlobalRank numbers.
Implementation note:
Until the difference between Juniper ProgramInstance and MPIGroupMember will be established,
the attribute "membership" will have limited set size to 1 and value of "instanceID" will be always equal to "mpiGlobalRank" value of MPIGroupMemeber instance in the "membership" attribute set.
-program
ProgramInstance
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -program : JuniperProgram
<<Property>> -cloudNode : CloudNode
<<Property>> -memberships : MpiGroupMember = new HashSet<>()
<<Property>> -mpiGlobalRank : int
<<Property>> -attribute : MpiGroupMember = new HashSet<>()
+ProgramInstance(mpiGlobalRank : int)
#setProgram(program : JuniperProgram) : void
#setCloudNode(cloudNode : CloudNode) : void
+addMembership(mpiGroupMember : MpiGroupMember) : MpiGroupMember
+removeMembership(mpiGroupMember : MpiGroupMember) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-programInstance
-communicationModel
CommunicationModel
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -application : JuniperApplication
<<Property>> -connections : DataConnection = new HashSet<>()
<<Property>> -attribute : DataConnection = new HashSet<>()
#setApplication(application : JuniperApplication) : void
+addConnection(dataConnection : DataConnection) : DataConnection
+removeConnection(dataConnection : DataConnection) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-model
DataConnection
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -connectionName : String
<<Property>> -sendingGroup : MpiGroup
<<Property>> -receivingGroup : MpiGroup
<<Property>> -model : CommunicationModel
<<Property>> -type : ConnectionType
+DataConnection(connectionName : String)
#setModel(model : CommunicationModel) : void
+getTypeAsString() : String
+setType(type : String) : boolean
+getReceivers(sender : MpiGroupMember) : Set<MpiGroupMember>
+getSenders(receiver : MpiGroupMember) : Set<MpiGroupMember>
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
getReceivers and getSenders find receivers for a sender and senders for a receiver, respectively, according to a defined data connection type.
-type
<<enumeration>>
ConnectionType
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
-SYMMETRIC : String = "symmetric"
-ALL_TO_ONE : String = "all_to_one"
-ONE_TO_ALL : String = "one_to_all"
-ALL_TO_ALL : String = "all_to_all"
+valueToString(value : ConnectionType) : String
+stringToValue(string : String) : ConnectionType
symmetric
all_to_one
one_to_all
all_to_all
MpiGroupMember
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -mpiLocalRank : int
<<Property>> -group : MpiGroup
<<Property>> -programInstance : ProgramInstance
+MpiGroupMember(mpiLocalRank : int, group : MpiGroup)
#setGroup(group : MpiGroup) : void
#setProgramInstance(programInstance : ProgramInstance) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-group
MpiGroup
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -groupName : String
<<Property>> -members : MpiGroupMember = new ArrayList<>()
<<Property>> -model : GroupModel
<<Property>> -attribute : MpiGroupMember = new ArrayList<>()
+MpiGroup(groupName : String)
+addMember(mpiGroupMember : MpiGroupMember) : MpiGroupMember
+removeMember(mpiGroupMember : MpiGroupMember) : void
#setModel(model : GroupModel) : void
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-receivingGroup
-sendingGroup
-model
GroupModel
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
<<Property>> -application : JuniperApplication
<<Property>> -mpiGroups : MpiGroup = new HashSet<>()
<<Property>> -attribute : MpiGroup = new HashSet<>()
#setApplication(application : JuniperApplication) : void
+addMpiGroup(mpiGroup : MpiGroup) : MpiGroup
+removeMpiGroup(mpiGroup : MpiGroup) : void
+getMpiGroupByName(name : String) : MpiGroup
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String
-groupModel
ModelEntity
<<Property>> -uuid : UUID = null
+getUuidAsUrn() : String
+getUuidAscName() : String
+hasUuid() : boolean
XMLDeploymentPlan
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
-APPLICATION : String = "application"
-PROGRAMMODEL : String = "ProgramModel"
-GROUPMODEL : String = "GroupModel"
-COMMUNICATIONMODEL : String = "CommunicationModel"
-DEPLOYMENTMODEL : String = "DeploymentModel"
-PROGRAM : String = "program"
-MPIGROUP : String = "mpigroup"
-MEMBER : String = "member"
-DATACONNECTION : String = "dataconnection"
-CLOUDNODE : String = "cloudnode"
-NAME : QName = new QName("name")
-JAVACLASS : QName = new QName("javaclass")
-MPIGLOBALRANK : QName = new QName("mpiglobalrank")
-PROGRAMNAME : QName = new QName("programName")
-SENDINGGROUP : QName = new QName("sendingGroup")
-RECEIVERMPIGROUP : QName = new QName("receiverMpiGroup")
-TYPE : QName = new QName("type")
-HOSTIPADDR : QName = new QName("hostipaddr")
+readJuniperApplication(inputStream : InputStream) : JuniperApplication
+writeJuniperApplication(juniperApplication : JuniperApplication, outputStream : OutputStream) : v...
+readJuniperApplication(inputFile : String) : JuniperApplication
+writeJuniperApplication(juniperApplication : JuniperApplication, outputFile : String) : void
+main(args : String []) : void
XMLDeploymentPlanException
-serialVersionUID : long = 42L
+XMLDeploymentPlanException()
+XMLDeploymentPlanException(message : String)
XMLDeploymentPlan
-APPLICATION : String = "application"
-PROGRAMMODEL : String = "ProgramModel"
-GROUPMODEL : String = "GroupModel"
-COMMUNICATIONMODEL : String = "CommunicationModel"
-DEPLOYMENTMODEL : String = "DeploymentModel"
-PROGRAM : String = "program"
-MPIGROUP : String = "mpigroup"
-MEMBER : String = "member"
-DATACONNECTION : String = "dataconnection"
-CLOUDNODE : String = "cloudnode"
-NAME : QName = new QName("name")
-JAVACLASS : QName = new QName("javaclass")
-MPIGLOBALRANK : QName = new QName("mpiglobalrank")
-MPILOCALRANK : QName = new QName("mpilocalrank")
-PROGRAMNAME : QName = new QName("programName")
-SENDINGGROUP : QName = new QName("sendingGroup")
-RECEIVERMPIGROUP : QName = new QName("receiverMpiGroup")
-TYPE : QName = new QName("type")
-HOSTIPADDR : QName = new QName("hostipaddr")
+SAOBJID : QName = new QName("http://www.fit.vutbr.cz/homes/rychly/juniper/scheduling-advisor", "objid", "...
+readJuniperApplication(inputStream : InputStream) : JuniperApplication
+writeJuniperApplication(juniperApplication : JuniperApplication, outputStream : OutputStream) : void
+writeJuniperApplication(juniperApplication : JuniperApplication, streamWriter : XMLStreamWriter) : void
-xmlStreamWriteAttributeSaObjId(modelEntity : ModelEntity, streamWriter : XMLStreamWriter) : void
+readJuniperApplication(inputFile : String) : JuniperApplication
+writeJuniperApplication(juniperApplication : JuniperApplication, outputFile : String) : void
+main(args : String []) : void
Implementation note:
readFromXMLDeploymentPlan and writeToXMLDeploymentPlan methods
will utilize Streaming API for XML (StaX)
to create JuniperApplication objects and to write them, respectively,
from and to XML files.
XMLDeploymentPlanException
(Implementation Model::juniper-sa-deployment-model::eu.juniper.sa.deployment.model)
-serialVersionUID : long = 42L
+XMLDeploymentPlanException()
+XMLDeploymentPlanException(message : String)