An framework for model-driven product design and development using Modelica

Adrian Pop, Olof Johansson, Peter Fritzson
Programming Environments Laboratory (PELAB)
Department of Computer and Information Science (IDA)
in collaboration with Department of Mechanical Engineering (IKP)
Linköping University (LiU)

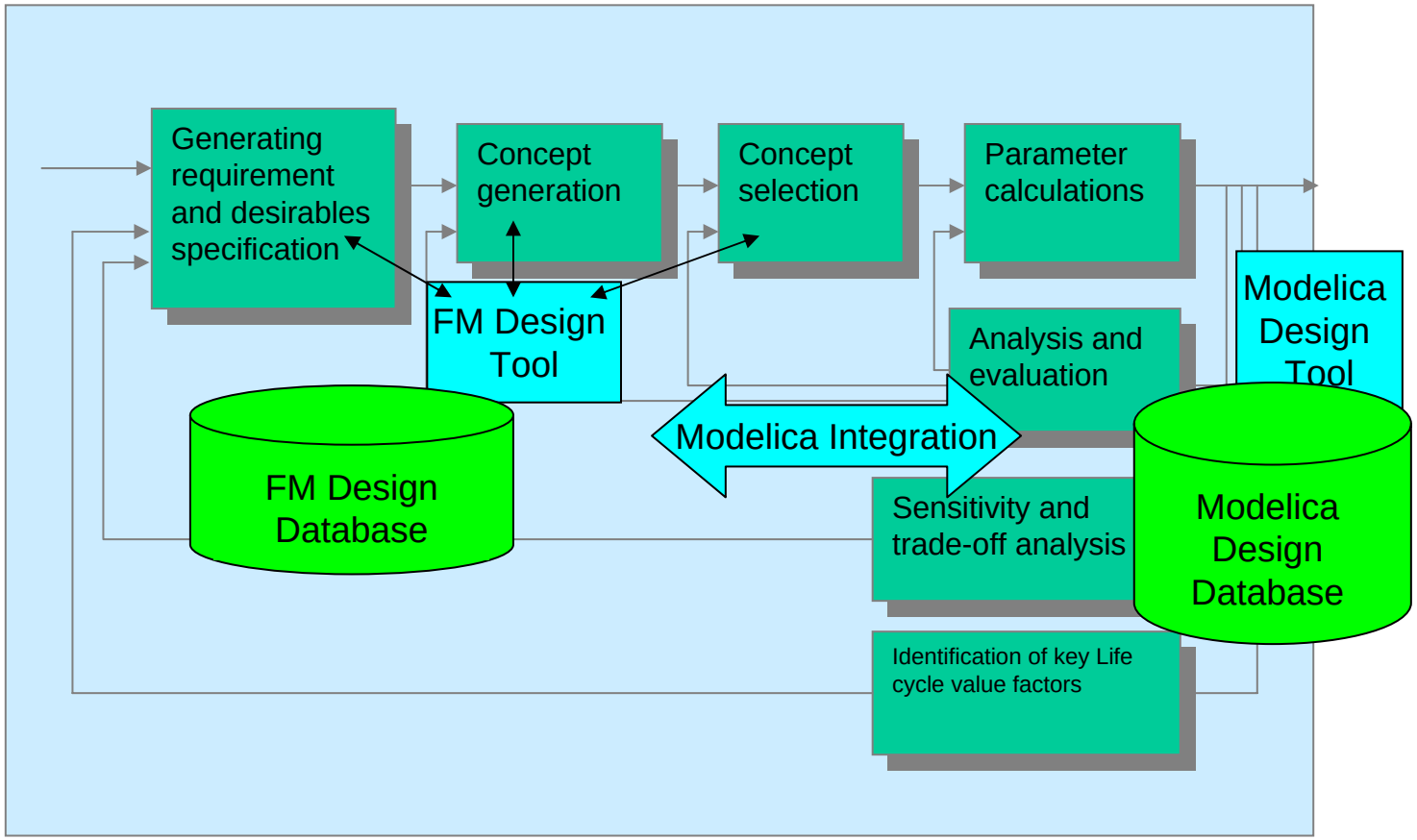
Outline

- Introduction
- Architecture Overview
- Framework Details
  - Modelica & ModelicaXML
  - Modelica Model Database ModelicaDB
  - FMDesign
  - Selection & Configuration tool
  - Automatic Model Generation tool
- Conclusions & Future Work

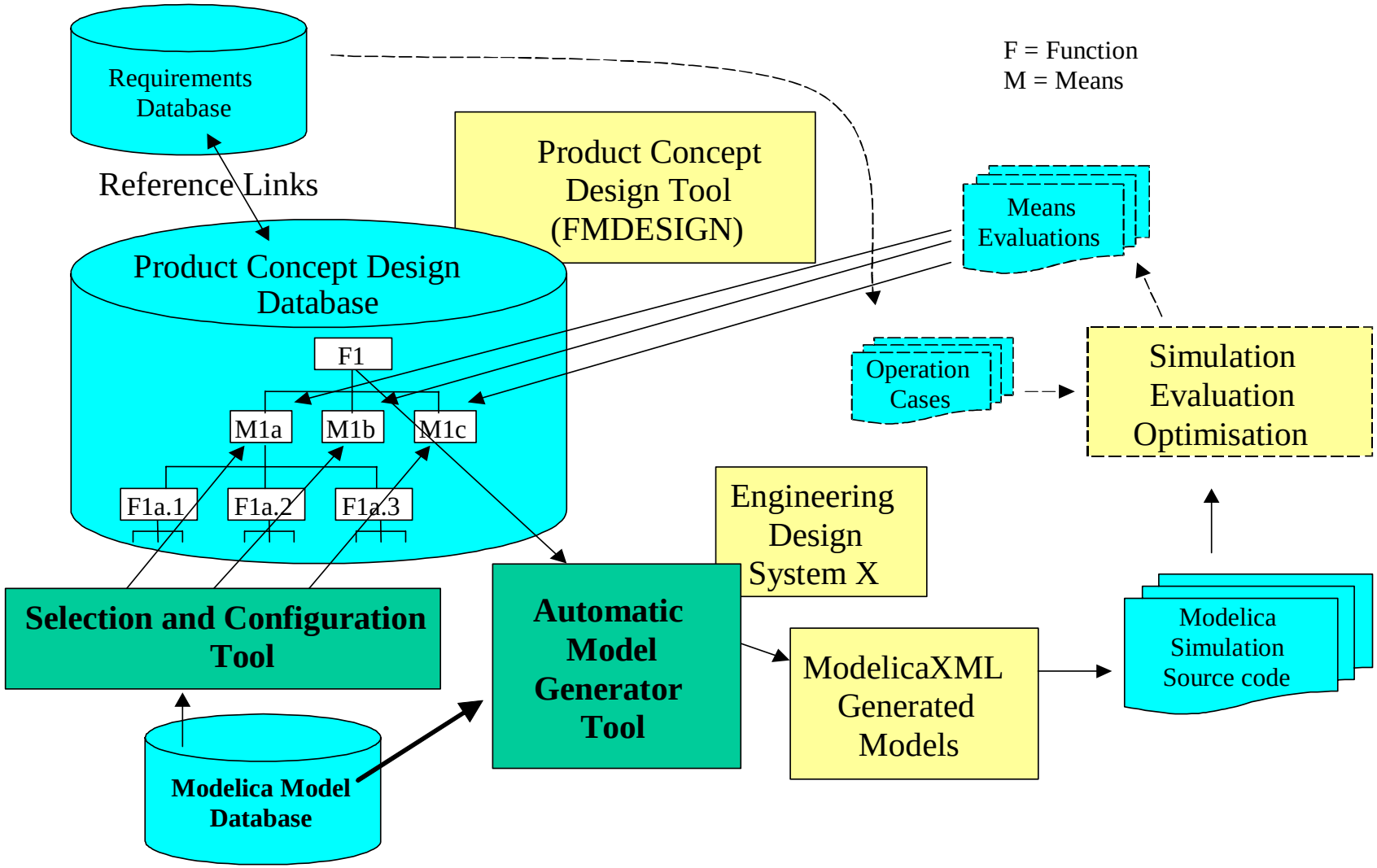
Introduction

- Product design
  - product concept modeling and evaluation
  - physical modeling and simulation
- Integration of
  - conceptual modeling tools and
  - modeling and simulation tools

Product Design phases



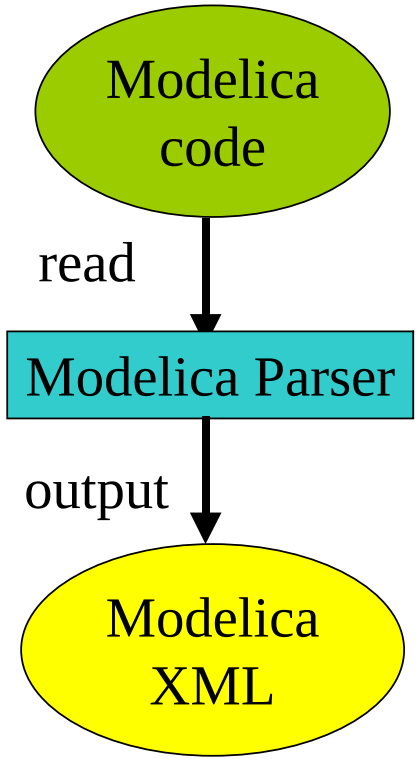
Architecture Overview



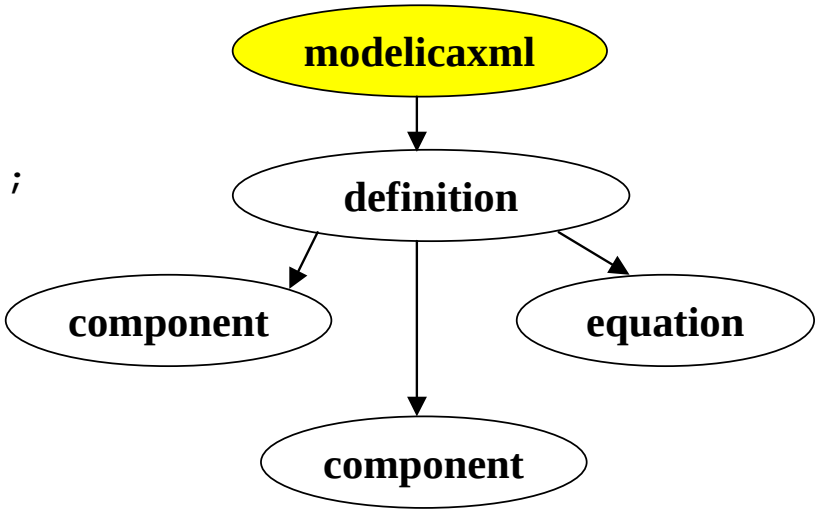
Modelica

- *Declarative language*
  - Equations and mathematical functions allow acausal modeling, high level specification, increased correctness
- *Multi-domain modeling*
  - Combine electrical, mechanical, thermodynamic, hydraulic, biological, control, event, real-time, etc…
- *Everything is a class*
  - Strongly typed object-oriented language with a general class concept, Java & Matlab like syntax
- *Visual component programming*
  - Hierarchical system architecture capabilities

ModelicaXML




```
class Test "comment"
   Real x;
   Real xdot;
 equation
   xdot = der(x);
end Test;
```




```
<modelicaxml>
  <definition ident= "Test"
              comment="comment">
     <component ident="x" type="Real"
                visibility="public" />
     <component ident="xdot" type="Real"
                visibility="public" />
     <equation>...</equation>
  </definition>
</modelicaxml>
```

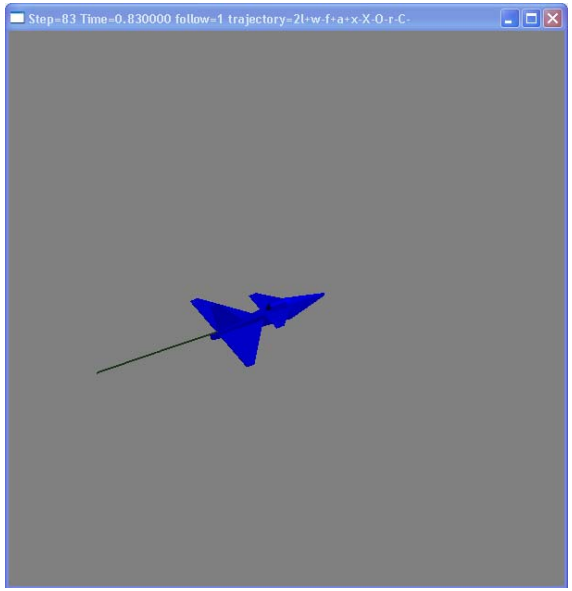
ModelicaDB – Modelica Model Database

- is populated with simulation models by importing their ModelicaXML representation

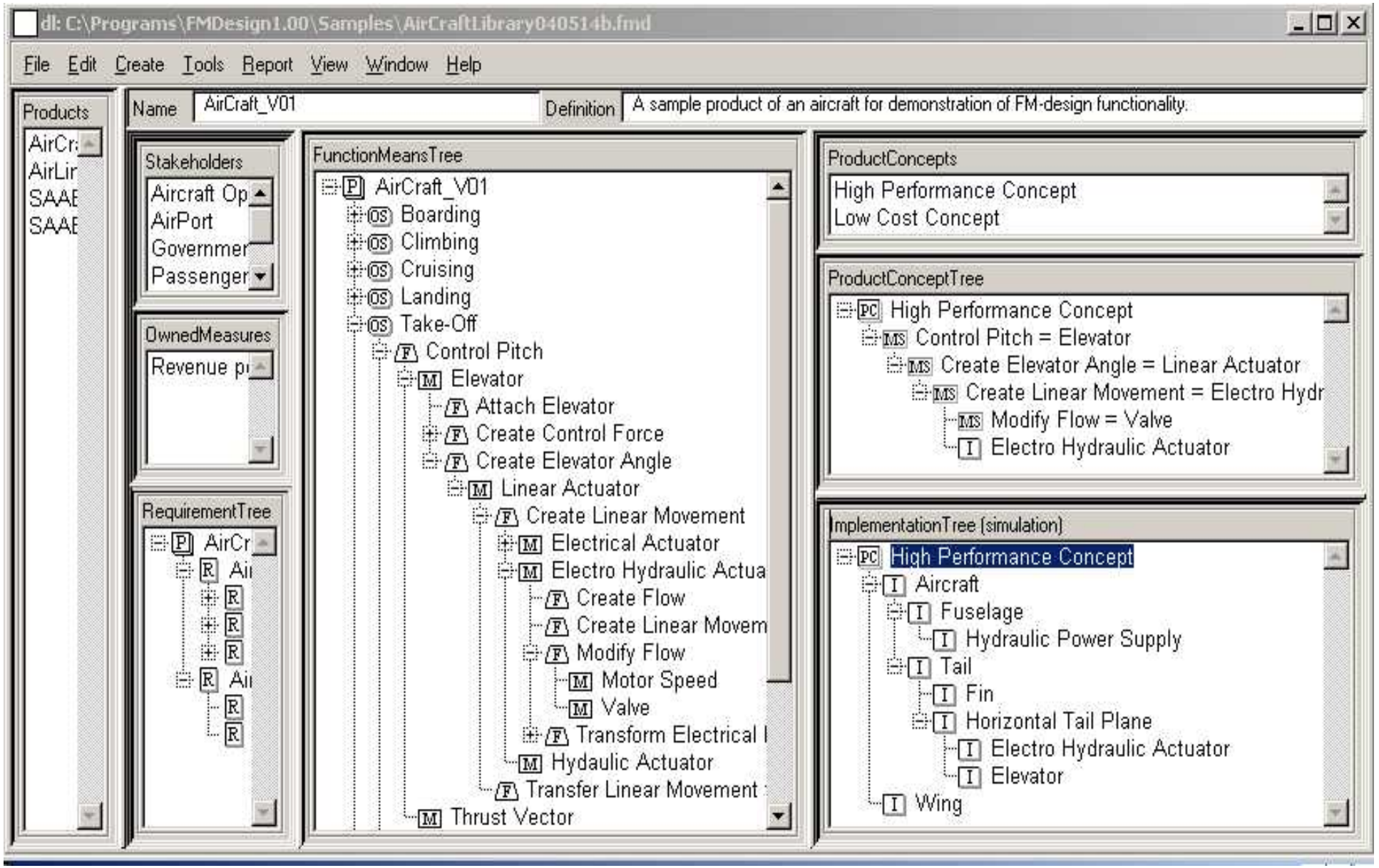
- provides
  - simulation models repository
  - search and organizational features
  - flexibility and scalability
  - collaborative development

Example: design phases of an Aircraft Product

- aircraft conceptual model in FMDesign
  - decomposition of the aircraft into functions and means
  - mapping between means and Modelica simulation components (Implementation Tree)
  - simulation of various design choices
  - choosing the best design choice using the simulation results



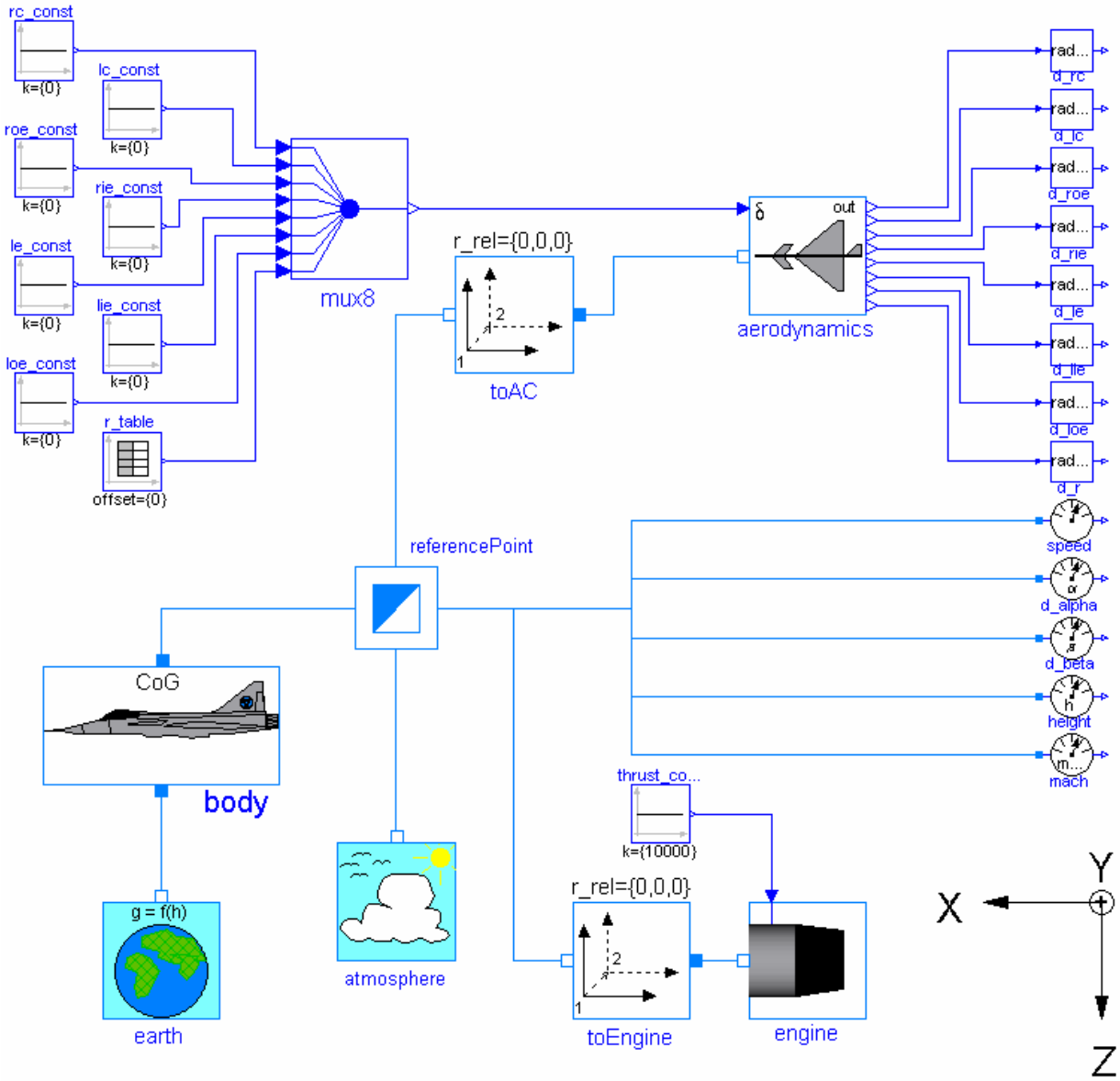
FMDesign


dl: C:\Programs\FMDesign1.00\Samples\AirCraftLibrary040514b.fmd
File Edit Create Tools Report View Window Help
Products
AirCr
AirLir
SAAE
SAAE
Name AirCraft_V01
Definition A sample product of an aircraft for demonstration of FM-design functionality.
Stakeholders
Aircraft Op
AirPort
Governmer
Passenger
OwnedMeasures
Revenue pr
RequirementTree
AirCr
Air
Air
FunctionMeansTree
AirCraft_V01
Boarding
Climbing
Cruising
Landing
Take-Off
Control Pitch
Elevator
Attach Elevator
Create Control Force
Create Elevator Angle
Linear Actuator
Create Linear Movement
Electrical Actuator
Electro Hydraulic Actua
Create Flow
Create Linear Movem
Modify Flow
Motor Speed
Valve
Transform Electrical I
Hydaulic Actuator
Transfer Linear Movement
Thrust Vector
ProductConcepts
High Performance Concept
Low Cost Concept
ProductConceptTree
High Performance Concept
Control Pitch = Elevator
Create Elevator Angle = Linear Actuator
Create Linear Movement = Electro Hydr
Modify Flow = Valve
Electro Hydraulic Actuator
mplementationTree (simulation)
High Performance Concept
Aircraft
Fuselage
Hydraulic Power Supply
Tail
Fin
Horizontal Tail Plane
Electro Hydraulic Actuator
Elevator
Wing

The Selection and Configuration Tool

- part of FMDesign and coupled with
  - ModelicaDB for searching capabilities
  - modeling tools (MathModelica, Dymola, OpenModelica) for creating/editing simulation models (components)
- provides
  - search for simulation models
  - creating/editing simulation models in external Modelica tools
  - configuration dialogs for selected simulation models for specific means implementation

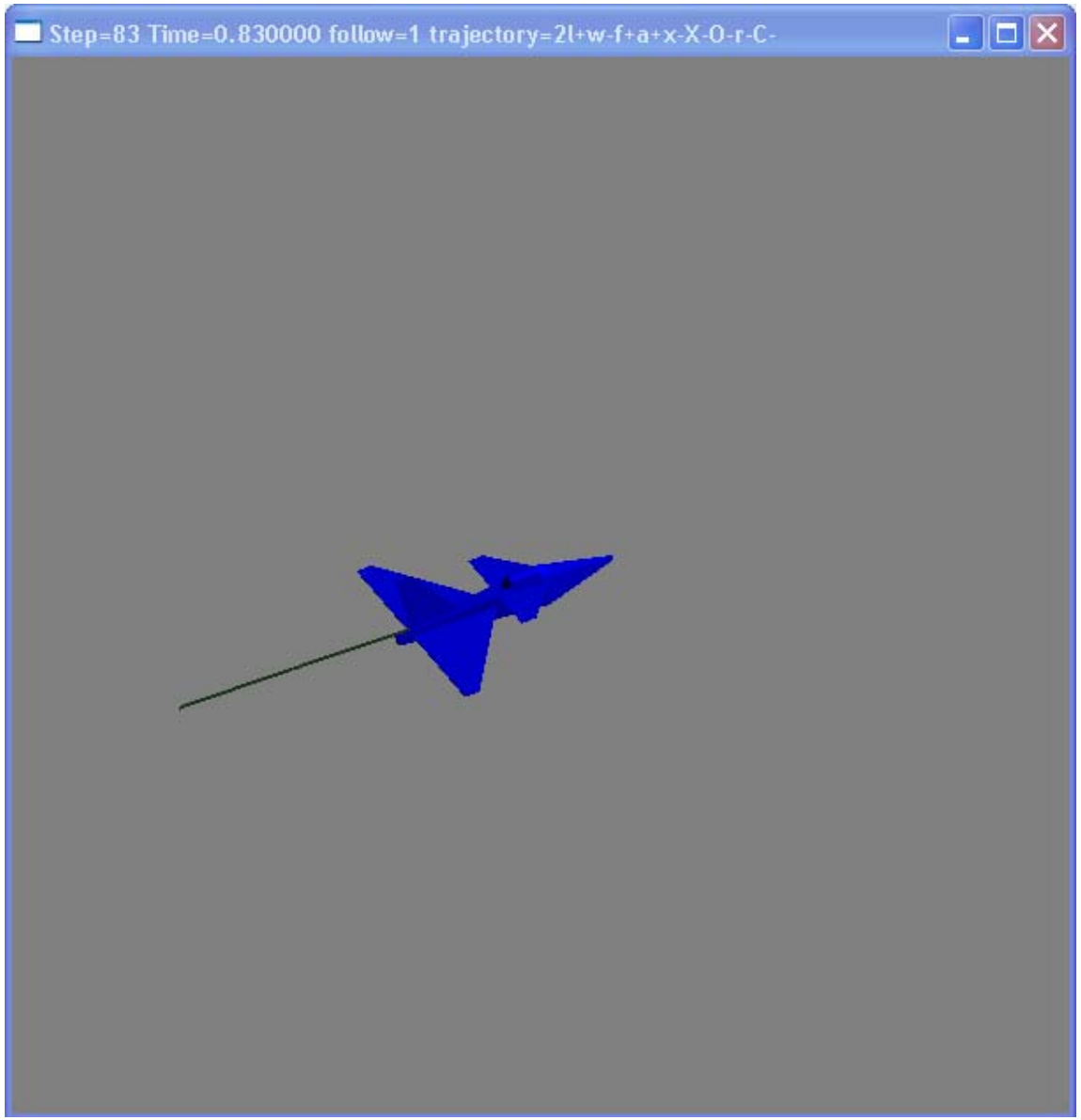
Simulation Components for an Aircraft Product



The Automatic Model Generator Tool

- part of FMDesign
- traverses an Implementation Tree (of type simulation)
- generates ModelicaXML models that are translated back into Modelica
- calls external simulation tools for simulation
- feeds the simulation results back to the designer to help him/her choose the best design choice

Aircraft simulation & visualization



Conclusions

- Conceptual Modeling (FMDesign) combined with Modeling and Simulation tools (Modelica)
- A prototype is currently in development at Linköping University

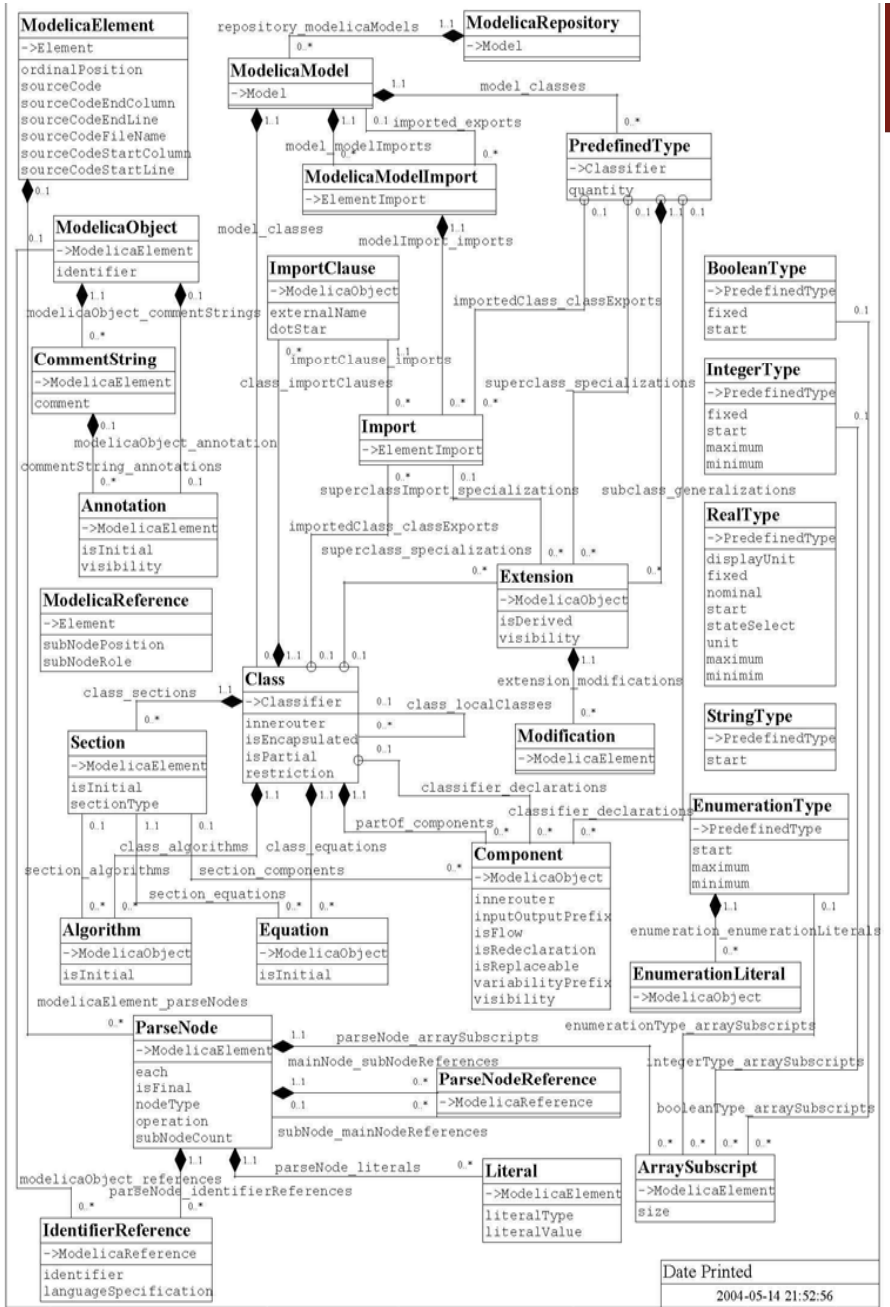
- Future Work
  - Using our ModelicaXML composition framework in the automatic model generator tool
  - Using Ontologies based on Semantic Web languages (Description Logic) for product concept design and classification of component libraries
  - Automatic selection of best design choice based on simulation results and requirements

End

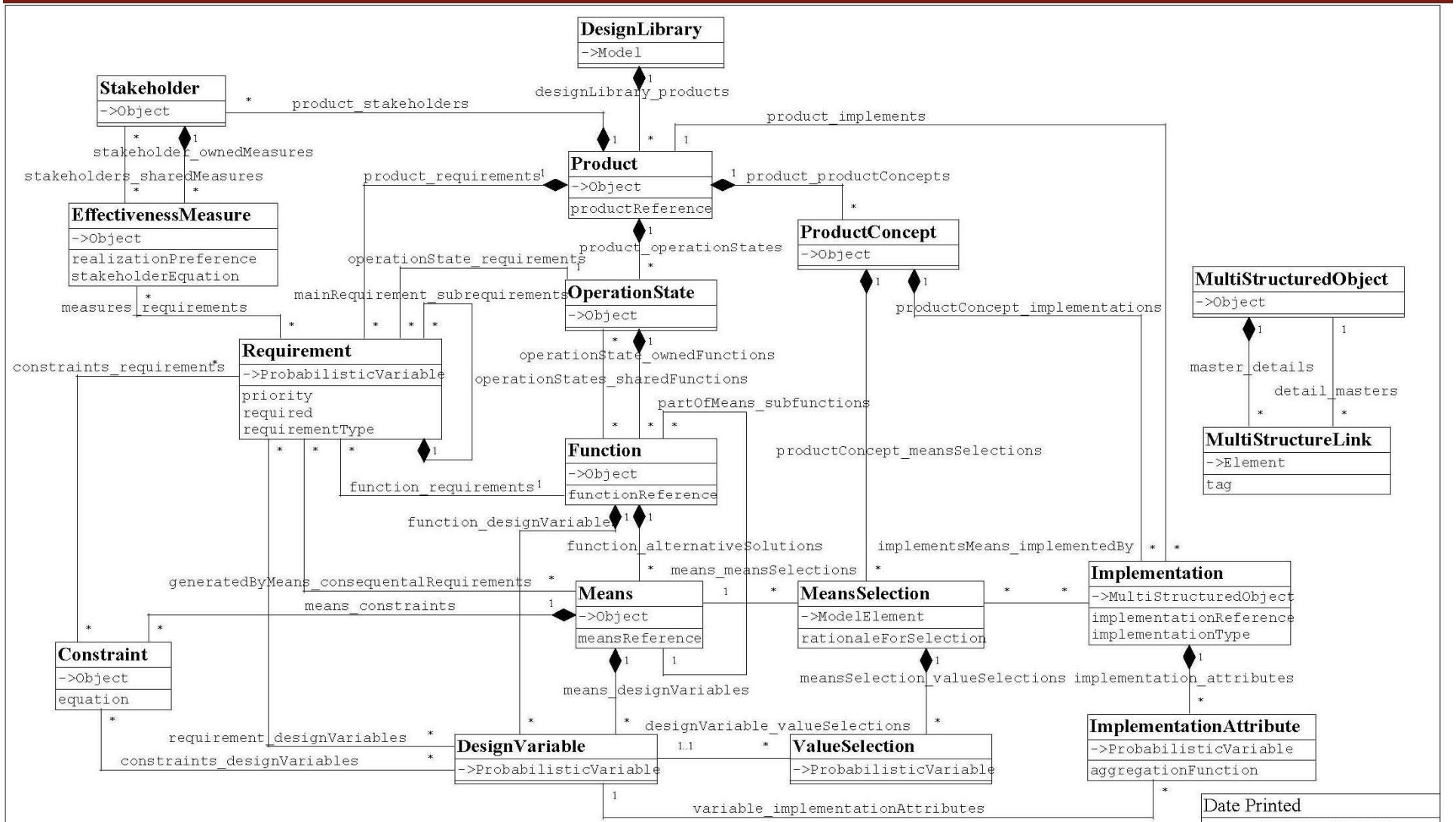
Thank you!

Questions?

ModelicaDB


ModelicaElement
->Element
ordinalPosition
sourceCode
sourceCodeEndColumn
sourceCodeEndLine
sourceCodeFileName
sourceCodeStartColumn
sourceCodeStartLine
repository_modelicaModels
ModelicaRepository
->Model
ModelicaModel
->Model
model_classes
imported_exports
model_modelImports
ModelicaModelImport
->ElementImport
PredefinedType
->Classifier
quantity
ModelicaObject
->ModelicaElement
identifier
model_classes
modelImport_imports
ImportClause
->ModelicaObject
externalName
dotStar
importedClass_classExports
BooleanType
->PredefinedType
fixed
start
modelicaObject_commentStrings
CommentString
->ModelicaElement
comment
importClause_imports
class_importClauses
superclass_specializations
IntegerType
->PredefinedType
fixed
start
maximum
minimum
Import
->ElementImport
modelicaObject_annotation
commentString_annotations
superclassImport_specializations
subclass_generalizations
Annotation
->ModelicaElement
isInitial
visibility
importedClass_classExports
superclass_specializations
RealType
->PredefinedType
displayUnit
fixed
nominal
start
stateSelect
unit
maximum
minimim
Extension
->ModelicaObject
isDerived
visibility
ModelicaReference
->Element
subNodePosition
subNodeRole
extension_modifications
Class
->Classifier
innerouter
isEncapsulated
isPartial
restriction
class_sections
class_localClasses
StringType
->PredefinedType
start
Section
->ModelicaElement
isInitial
sectionType
Modification
->ModelicaElement
classifier_declarations
classifier_declarations
EnumerationType
->PredefinedType
start
maximum
minimum
partOf_components
class_algorithms
class_equations
Component
->ModelicaObject
innerouter
inputOutputPrefix
isFlow
isRedeclaration
isReplaceable
variabilityPrefix
visibility
section_algorithms
section_components
section_equations
enumeration_enumerationLiterals
Algorithm
->ModelicaObject
isInitial
Equation
->ModelicaObject
isInitial
EnumerationLiteral
->ModelicaObject
modelicaElement_parseNodes
ParseNode
->ModelicaElement
each
isFinal
nodeType
operation
subNodeCount
parseNode_arraySubscripts
enumerationType_arraySubscripts
mainNode_subNodeReferences
integerType_arraySubscripts
ParseNodeReference
->ModelicaReference
booleanType_arraySubscripts
subNode_mainNodeReferences
parseNode_literals
modelicaObject_references
parseNode_identifierReferences
Literal
->ModelicaElement
literalType
literalValue
ArraySubscript
->ModelicaElement
size
IdentifierReference
->ModelicaReference
identifier
languageSpecification
Date Printed
2004-05-14 21:52:56
Domain Model
Filename
Date Added
Date Modified
#Cls
#Rel
#Att
DM.
ModelicaDB-21-02n
ModelicaDB-21-02o.odm
2003-03-04 08:03:07
2004-05-14 21:51:54
53
50
137
Modelica 2.1 database design for ModelicaXML information exchange. The goal is a repository that can be round-tripped with Modelica source code.
Dgm
SIMS'2004: Modelica Package
2003-03-04 09:33:10
2004-05-14 21:51:54
29
39
-
Diagram 1: UML Model of Modelica 2.1. (Note that "->ModelicaObject", means inherits from ModelicaObject)

FMDesign UML



Date Printed
2004-05-14 21:46:56

	Domain Model	Filename	Date Added	Date Modified	#Cls	#Rel	#Att
DM.	fmdesign03b	fmdesign03b.odm	2003-04-24 20:01:23	2004-05-14 20:00:16	40	37	92
	Analysis model of a function means design system prototype. Its a summary of the design FMDesign V1.2a.						
Dgm	3) Function Means Application		2003-05-08 01:50:24	2004-05-14 19:58:01	17	33	-
	Architecture of the FM design system, showing inheritance and attributes of classes.						