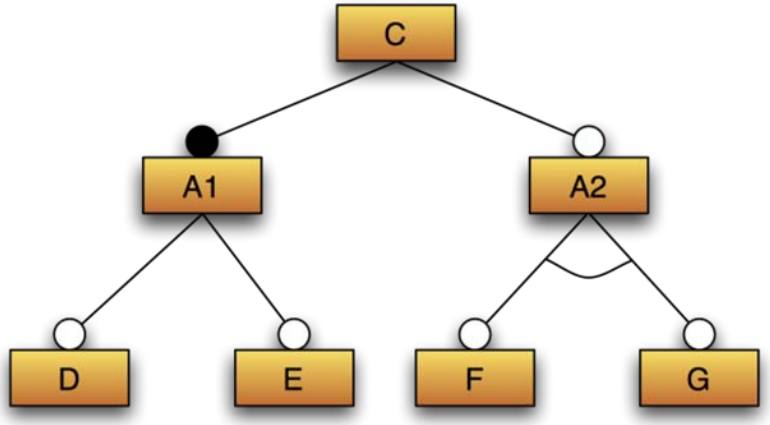
KumbangTools

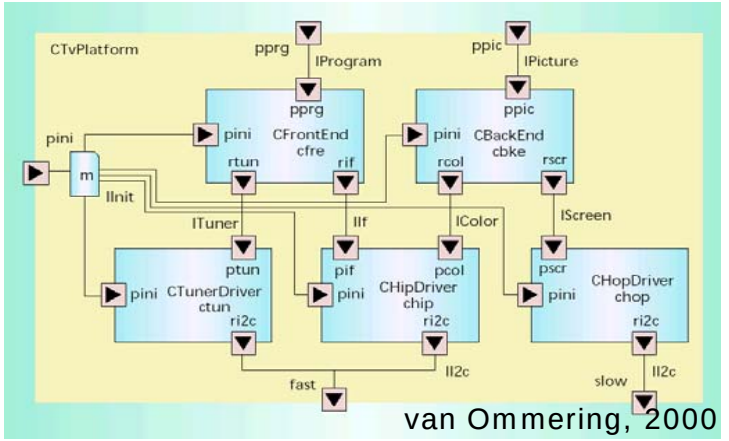
for modelling and configuring large variability

Varvana Myllärniemi, Mikko Raatikainen, Tomi Männistö

Helsinki University of Technology
Finland

Background





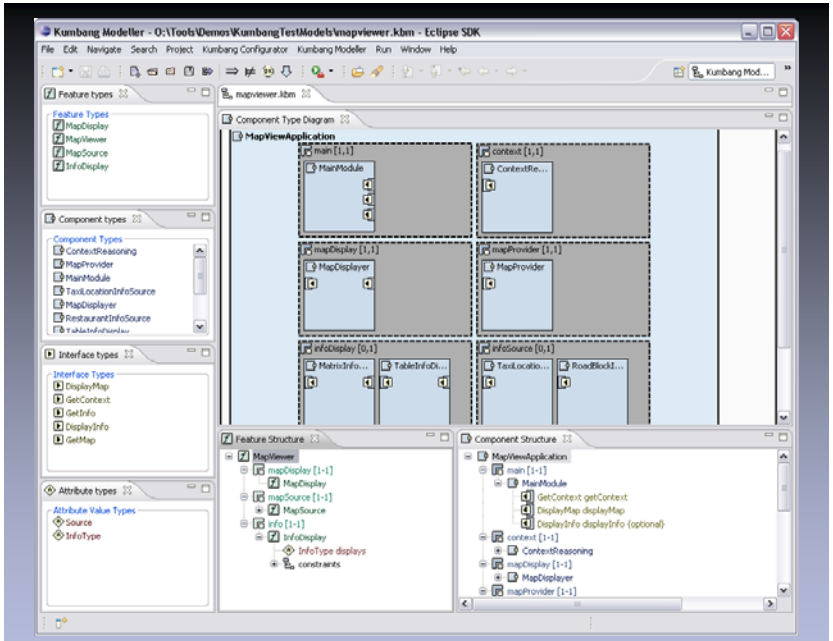
van Ommering, 2000

Combination of feature modeling and structural modeling as a meta-model (ontology) called Kumbang

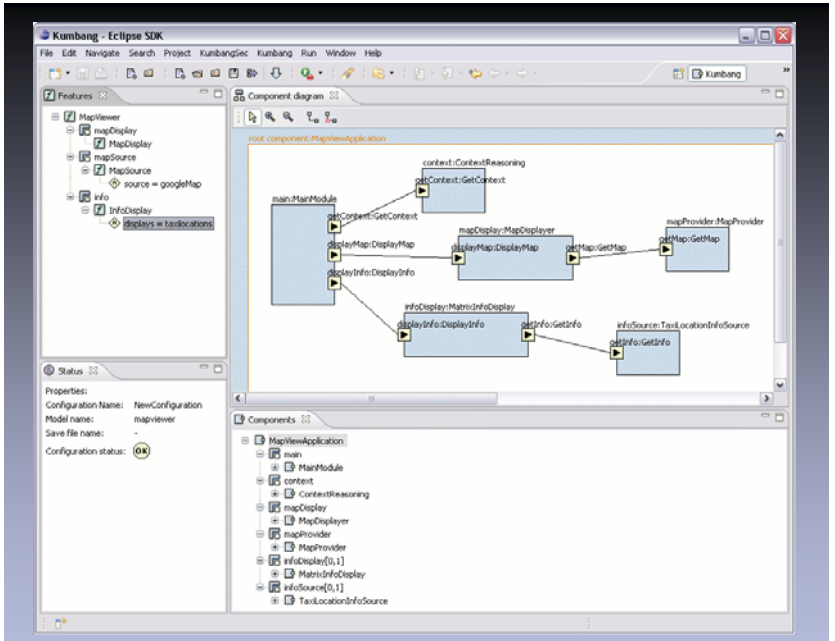
Kumbang Meta-Model

- Kumbang is a *meta-model* (domain ontology, conceptualization), and a *language* for configurable applications
- Kumbang provides concepts for modelling variability from two viewpoints adhering to IEEE 1471-2000 standard
  - The user-visible characteristics, i.e., *features*
    - Synthesizes existing methods
  - The *structure* of the products in terms of components, interfaces etc.
    - Builds on Koala concepts and representation, but does not require Koala component model
- Interrelations between the views can be specified
- Kumbang is provided a formal semantics by defining a mapping from the meta-model to weight constraint rule language
- Supports domain engineering (variability modelling, types) and application engineering (product derivation, instances)

KumbangTools



Domain engineering by Kumbang Modeller



Application engineering by Kumbang Configurator

Kumbang Core
uses AI inference engine smodels

KumbangTools

- KumbangTools
  - Feature and structural viewpoints adhering to Kumbang meta-model
  - Several representations to the model such as component type listing, structural diagram, and structure tree
  - Eclipse plug-ins
  - Uses AI inference engine smodels
  - Freely available under GPL
- *Kumbang Modeller* for creating a model of the variability (domain engineering) in a software product family
  - Checks model validity
- *Kumbang Configurator* for resolving the variability (product derivation) in a software product family to meet the specific set of requirements
  - Checks completeness, consistency, and consequences

Demo!

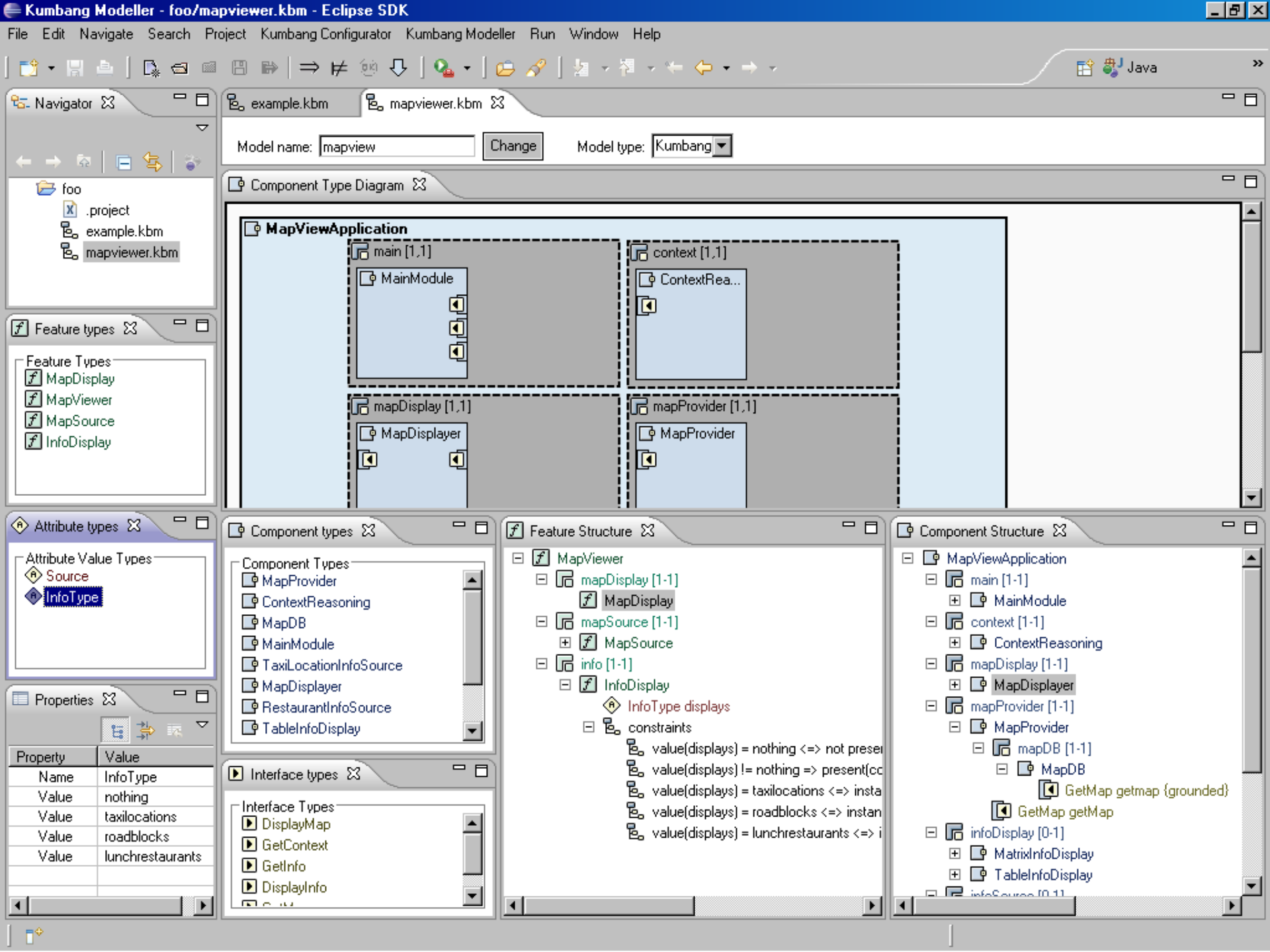
Final Remarks

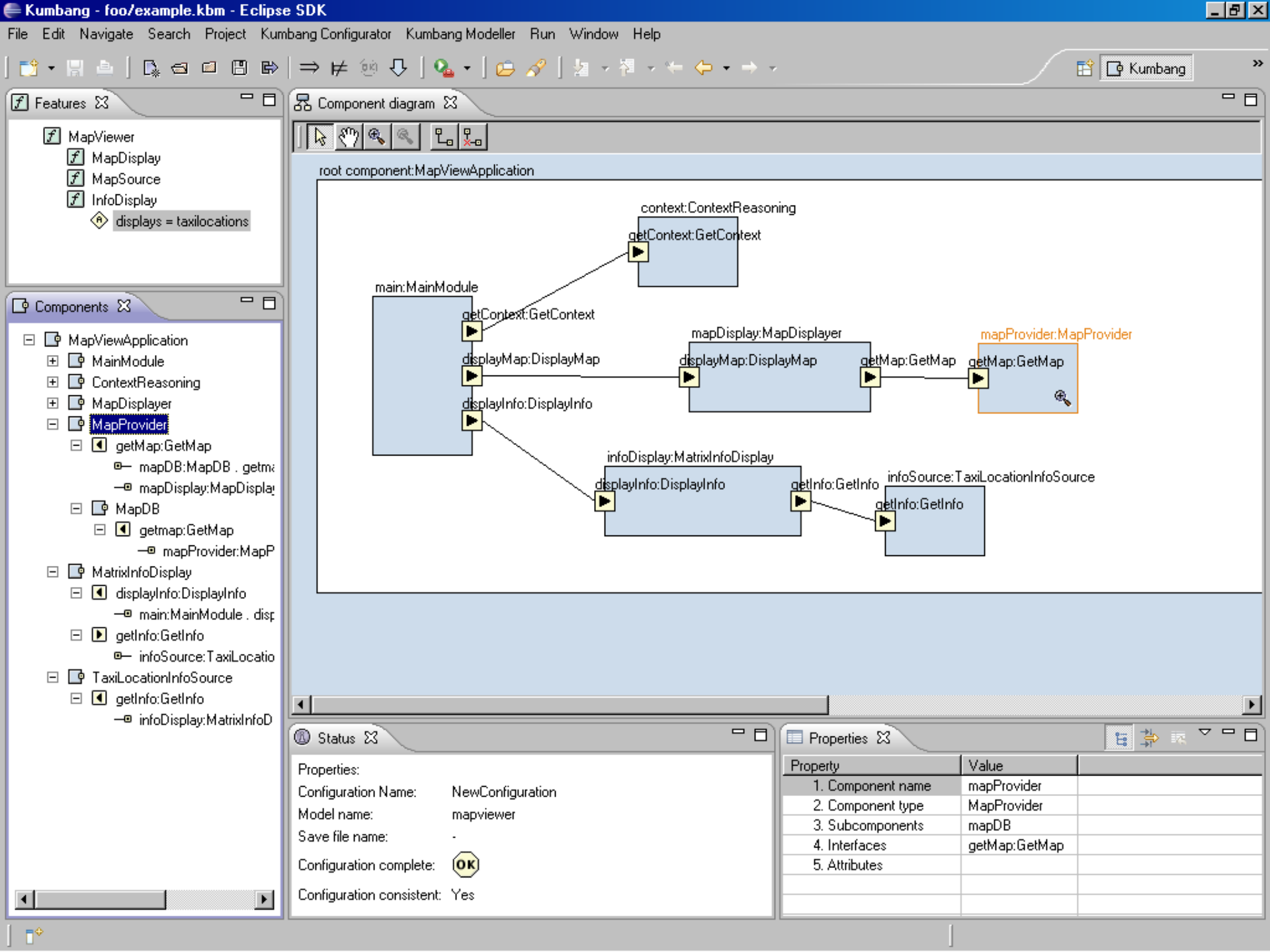
- Demo available
- Tools and material available:
  - A USB memory stick here at SPLC for copying
  - www.soberit.tkk.fi/preago

Thank you!

Questions?

mikko.raatikainen@tkk.fi


Kumbang Modeller - foo/mapviewer.kbm - Eclipse SDK
File Edit Navigate Search Project Kumbang Configurator Kumbang Modeller Run Window Help
Java
Navigator
foo
.project
example.kbm
mapviewer.kbm
example.kbm
mapviewer.kbm
Model name: mapview
Change
Model type: Kumbang
Component Type Diagram
MapViewApplication
main [1,1]
MainModule
context [1,1]
ContextRea...
mapDisplay [1,1]
MapDisplayer
mapProvider [1,1]
MapProvider
Feature types
Feature Types
MapDisplay
MapViewer
MapSource
InfoDisplay
Attribute types
Attribute Value Types
Source
InfoType
Component types
Component Types
MapProvider
ContextReasoning
MapDB
MainModule
TaxiLocationInfoSource
MapDisplayer
RestaurantInfoSource
TableInfoDisplay
Interface types
Interface Types
DisplayMap
GetContext
GetInfo
DisplayInfo
Feature Structure
MapViewer
mapDisplay [1-1]
MapDisplay
mapSource [1-1]
MapSource
info [1-1]
InfoDisplay
InfoType displays
constraints
value(displays) = nothing <=> not prese
value(displays) != nothing => present(cc
value(displays) = taxilocations <=> insta
value(displays) = roadblocks <=> instan
value(displays) = lunchrestaurants <=> i
Component Structure
MapViewApplication
main [1-1]
MainModule
context [1-1]
ContextReasoning
mapDisplay [1-1]
MapDisplayer
mapProvider [1-1]
MapProvider
mapDB [1-1]
MapDB
GetMap getmap {grounded}
GetMap getMap
infoDisplay [0-1]
MatrixInfoDisplay
TableInfoDisplay
Properties
Property
Value
Name
InfoType
Value
nothing
Value
taxilocations
Value
roadblocks
Value
lunchrestaurants


Kumbang - foo/example.kbm - Eclipse SDK
File Edit Navigate Search Project Kumbang Configurator Kumbang Modeller Run Window Help
Kumbang
Features
MapViewer
MapDisplay
MapSource
InfoDisplay
displays = taxilocations
Components
MapViewApplication
MainModule
ContextReasoning
MapDisplayer
MapProvider
getMap:GetMap
mapDB:MapDB . getma
mapDisplay:MapDisplay
MapDB
getmap:GetMap
mapProvider:MapP
MatrixInfoDisplay
displayInfo:DisplayInfo
main:MainModule . disp
getInfo:GetInfo
infoSource:TaxiLocatio
TaxiLocationInfoSource
getInfo:GetInfo
infoDisplay:MatrixInfoD
Component diagram
root component:MapViewApplication
context:ContextReasoning
getContext:GetContext
main:MainModule
getContext:GetContext
mapDisplay:MapDisplayer
mapProvider:MapProvider
displayMap:DisplayMap
displayMap:DisplayMap
getMap:GetMap
getMap:GetMap
displayInfo:DisplayInfo
infoDisplay:MatrixInfoDisplay
displayInfo:DisplayInfo
getInfo:GetInfo
infoSource:TaxiLocationInfoSource
getInfo:GetInfo
Status
Properties:
Configuration Name: NewConfiguration
Model name: mapviewer
Save file name: -
Configuration complete: OK
Configuration consistent: Yes
Properties
Property
Value
1. Component name
mapProvider
2. Component type
MapProvider
3. Subcomponents
mapDB
4. Interfaces
getMap:GetMap
5. Attributes