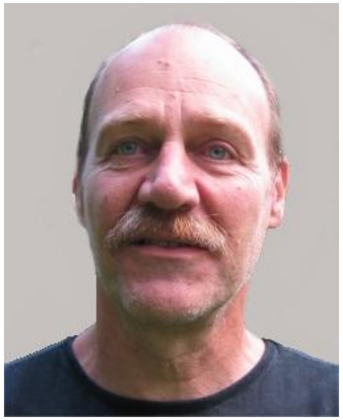
esito

Bringing Life to Models

Model driven development of user interfaces and services with Genova

Who are we?

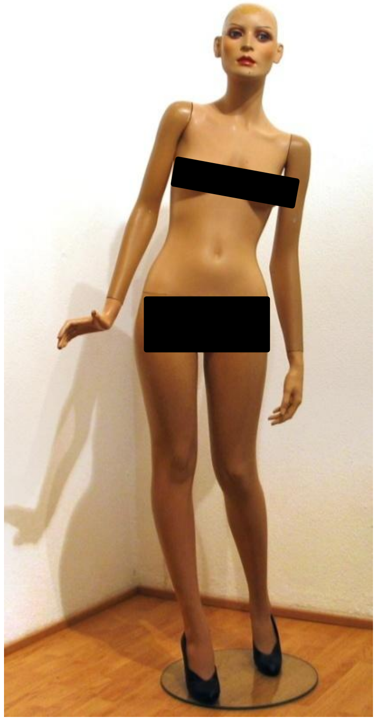
- Dag Bøyesen
  - Development Manager
  - db@esito.no



- Knut Sagli
  - Senior Consultant
  - Contribute as architect to Genova frameworks
  - ksa@esito.no

Contents

- Esito
- Applications built with Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets

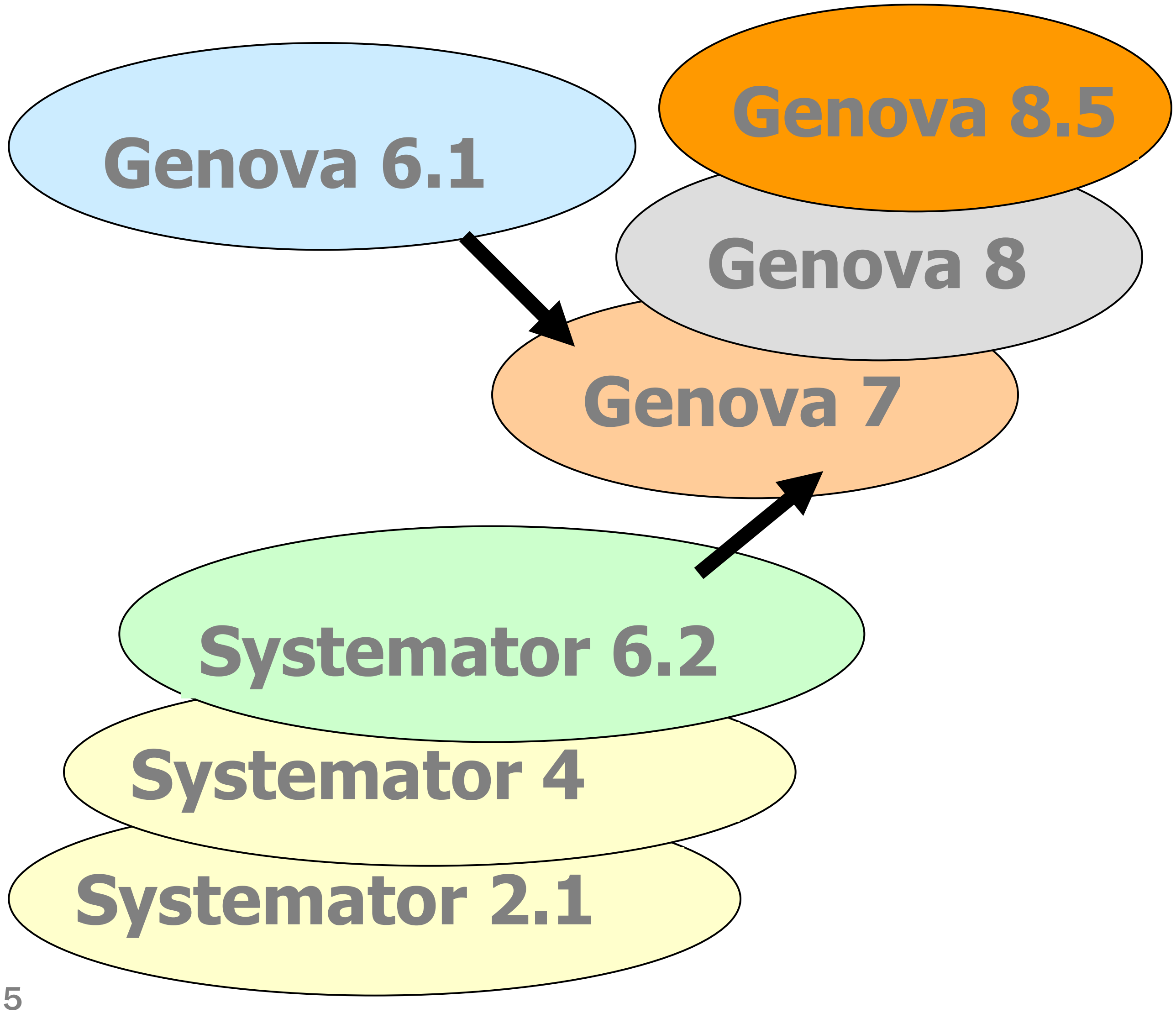


Esito vision

Aim at giving our customers lower cost and improved quality when developing and maintaining their applications

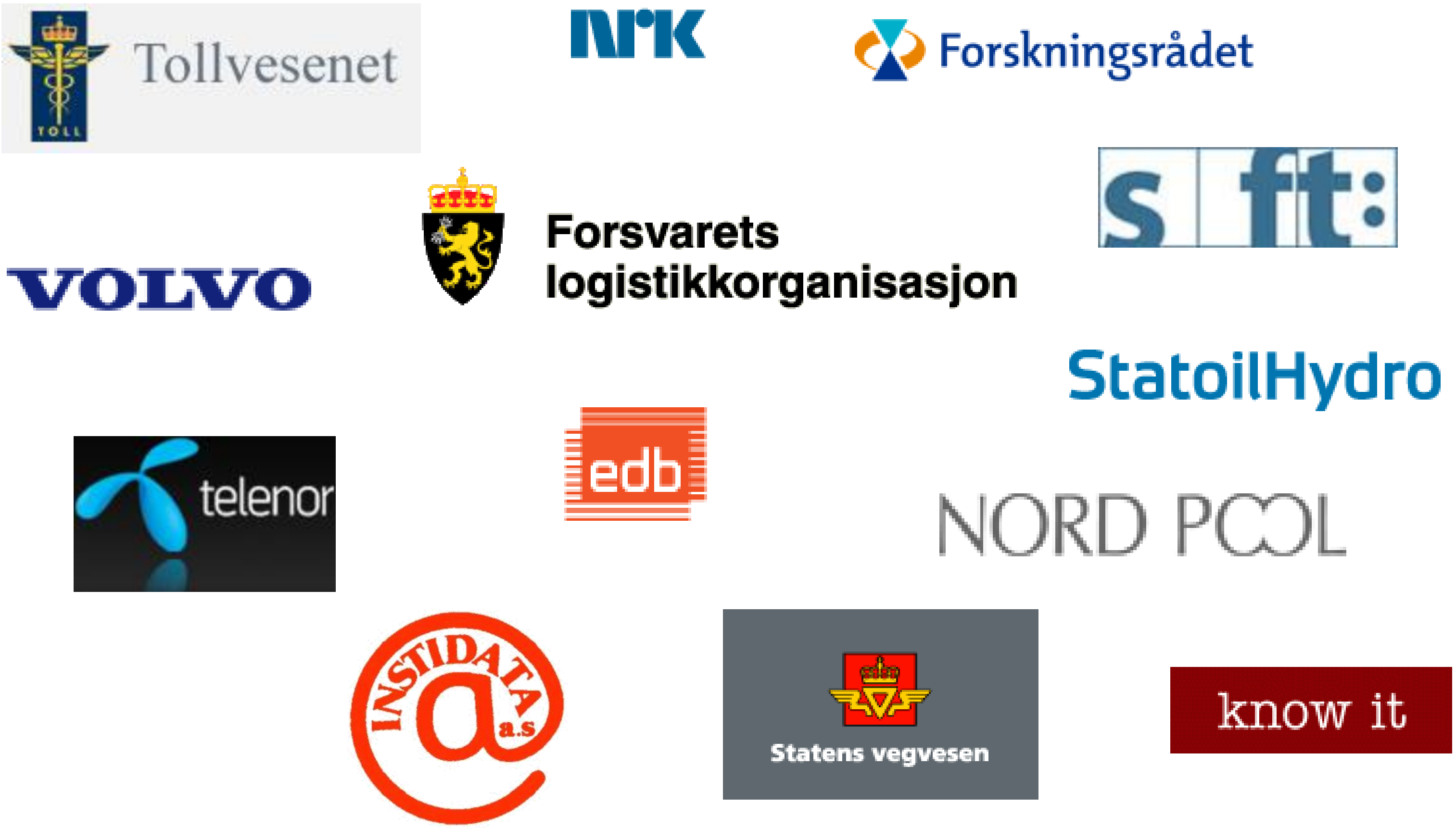
Support model driven development and make the best code generators

Tool history



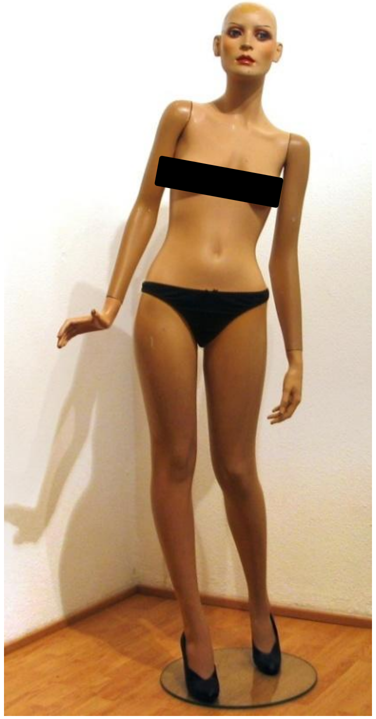
2010: Genova 8.5
jVine/Jouteur
2009: Genova 8.3
XML repository
2008: Genova 8.2
Enterprise Architect
User defined generator
2007: Genova 8.1
Service Designer
2005: Esito
2004: Genova 8, new IDE
2000: Genova 7
with Sysdul
1998: Genova 6
1997: Genera
1996: Windows NT
1995: Systemator 5
1992: GUI
1989: Unix
80s: NSB, TAD
1980: Sysdeco
70s: Research
Frode Aschim

Customers


Tollvesenet
TOLL
NRK
Forskningsrådet
VOLVO
Forsvarets logistikkorganisasjon
sft:
StatoilHydro
telenor
edb
NORD POOL
INSTIDATA a.s
Statens vegvesen
know it

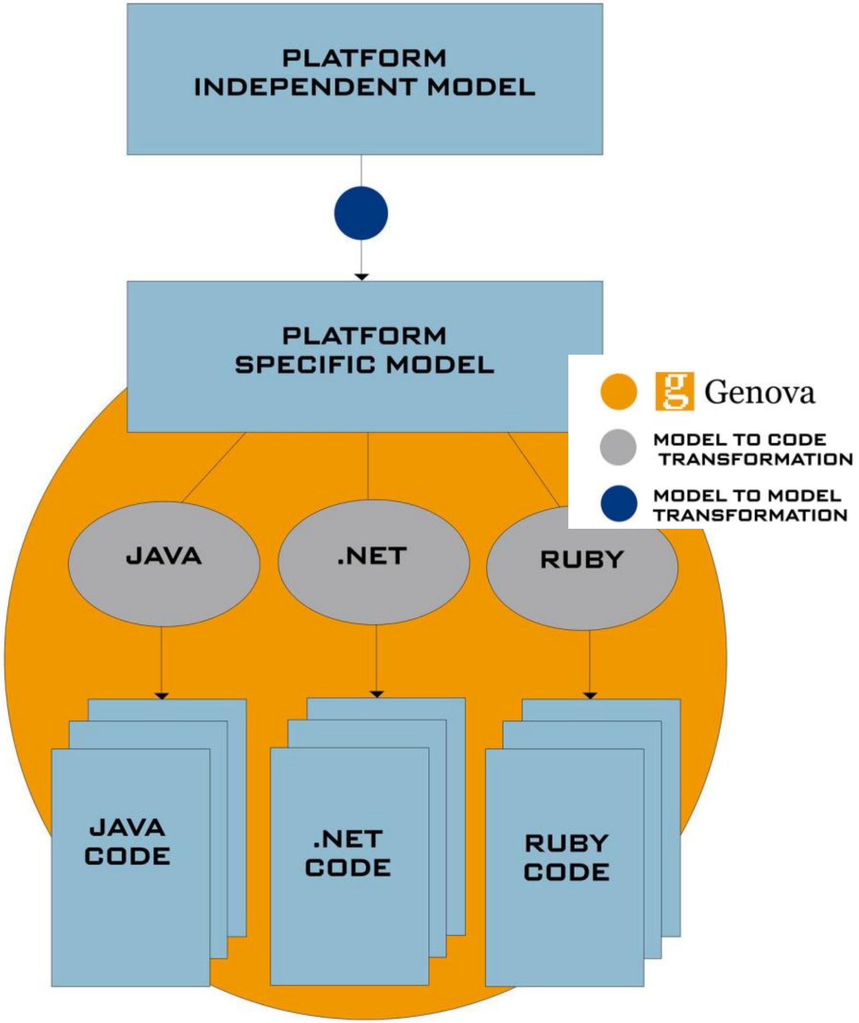
Contents

- Esito
- Applications built using Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets



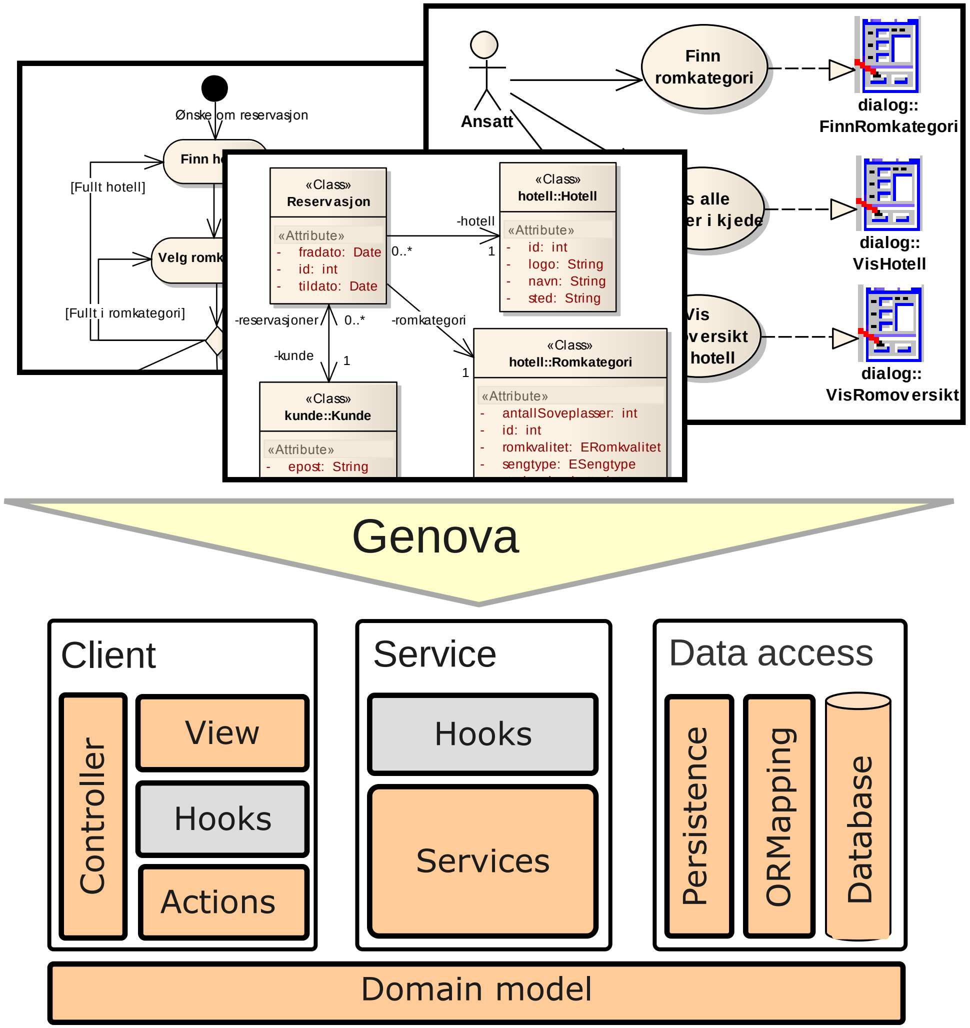
Genova in an MDA context

- Model Driven Architecture
- Domain Driven Design
- Model Driven Development


PLATFORM INDEPENDENT MODEL
PLATFORM SPECIFIC MODEL
Genova
MODEL TO CODE TRANSFORMATION
MODEL TO MODEL TRANSFORMATION
JAVA
.NET
RUBY
JAVA CODE
.NET CODE
RUBY CODE

esito

Bringing Life to Models

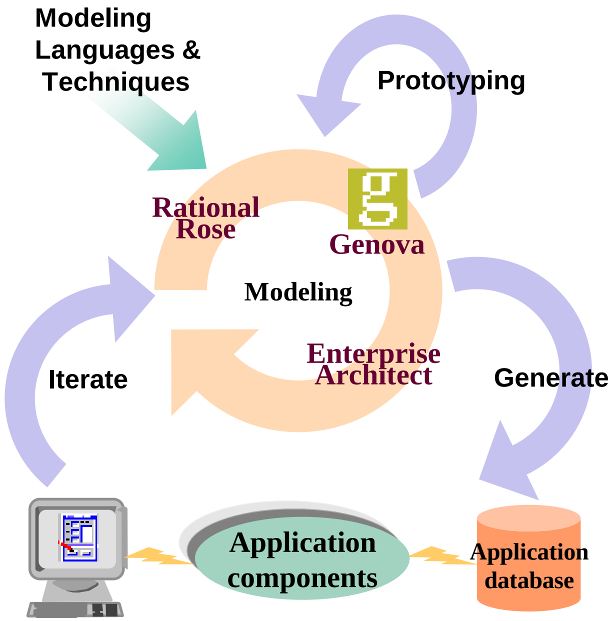


🟧 Generated code

⬜ Business logic

Genova is a tool

- for modeling and generating data intensive applications
- for maintaining applications with long life cycle
- giving added value in all phases of the development life cycle



Genova supports

- UML modeling
  - Integrated with Enterprise Architect and Rational Rose
- modeling of user interfaces
- template based code generation
  - User defined templates/generator
  - User defined ”tagged values”
- prototyping of user interfaces
- generation of user interfaces, services and databases for different targets

Applications built using Genova

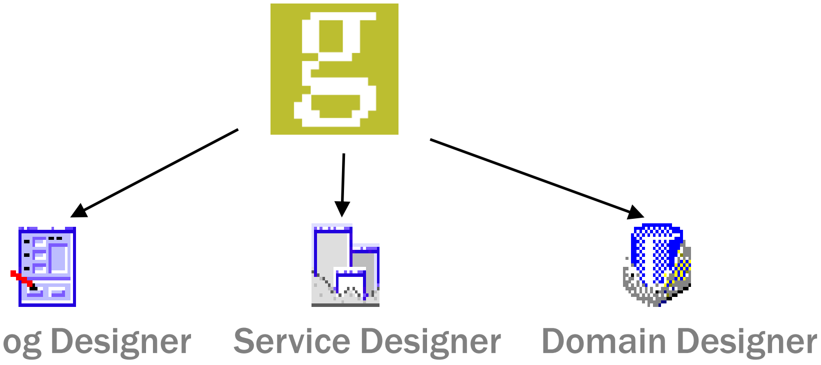
- Transaction oriented and data intensive applications
  - Main tasks are retrieval and storage of data
- Mission critical administrative systems
- Use models as source of program execution
- Survive paradigm changes

Contents

- Presentations
- Applications built using Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets

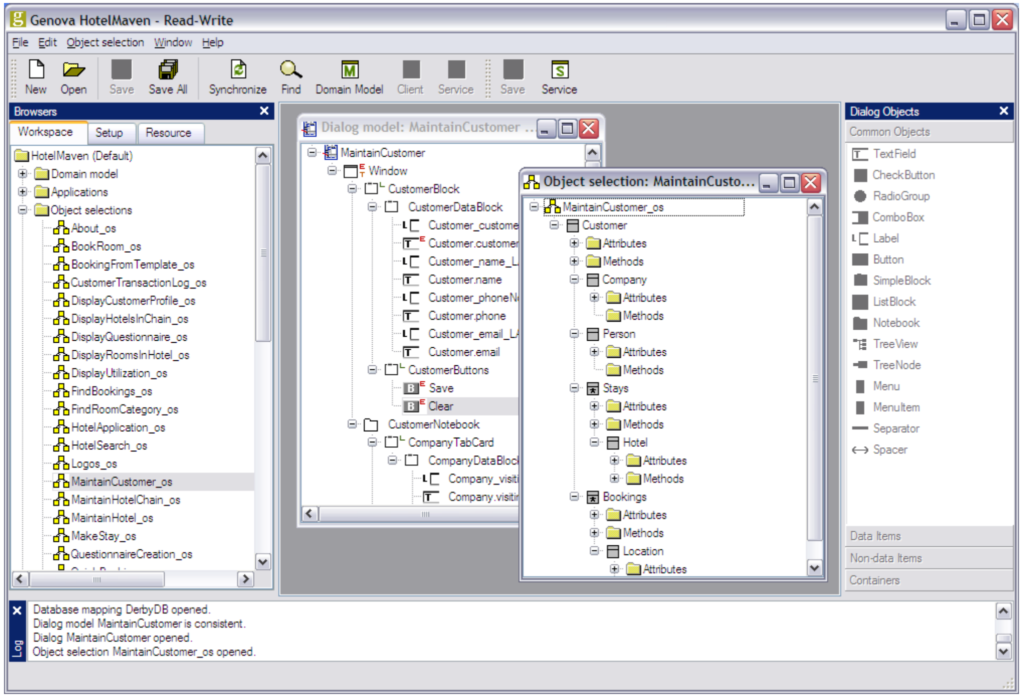
Genova modules

- Synchronizing with UML tool
- Domain Designer
- Service Designer
- Dialog Designer
- Generators



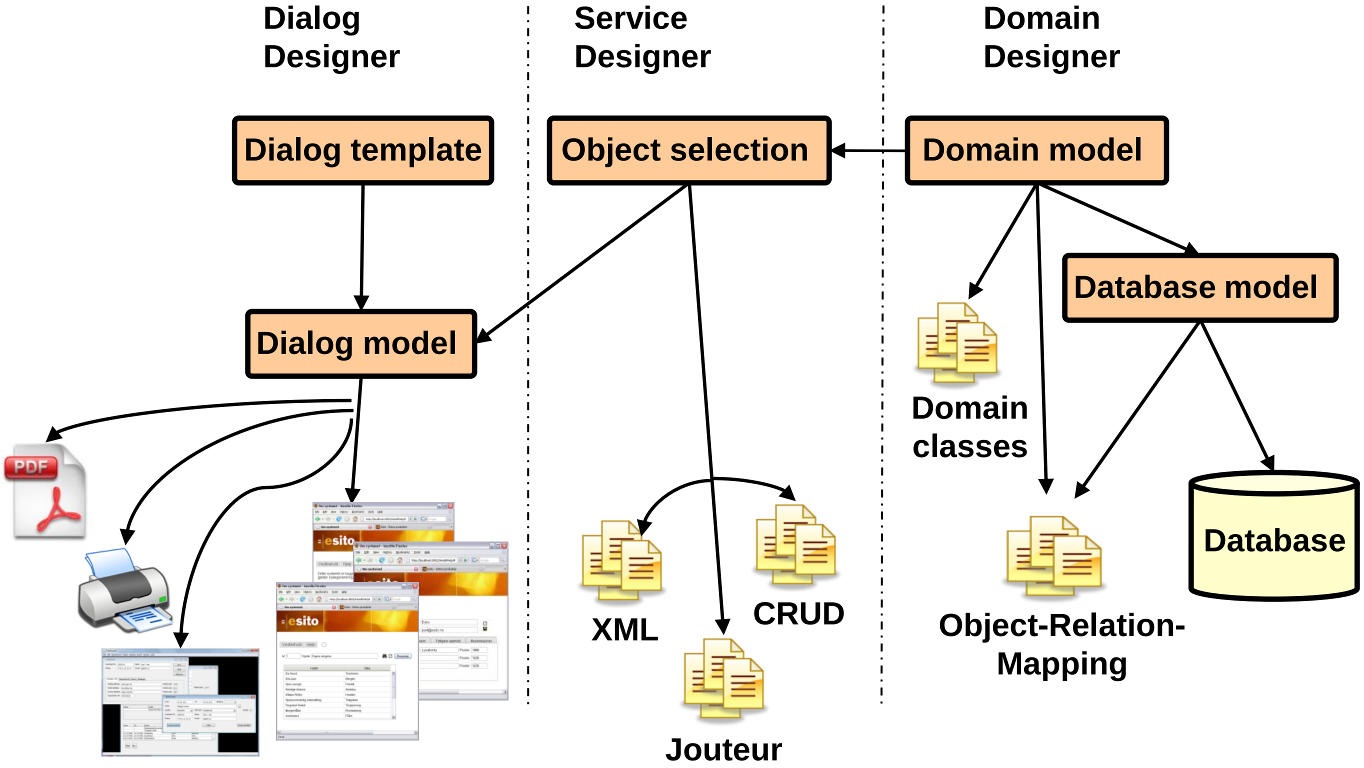
Demo: Genova walkthrough

- Showing tools and interaction
- Generating from an existing workspace
- Running the application



```
Database mapping DerbyDB opened.
Dialog model MaintainCustomer is consistent.
Dialog MaintainCustomer opened.
Object selection MaintainCustomer_os opened.
```

Genova models



Dialog Designer

Service Designer

Domain Designer

Dialog template

Object selection

Domain model

Database model

Dialog model

Domain classes

XML

CRUD

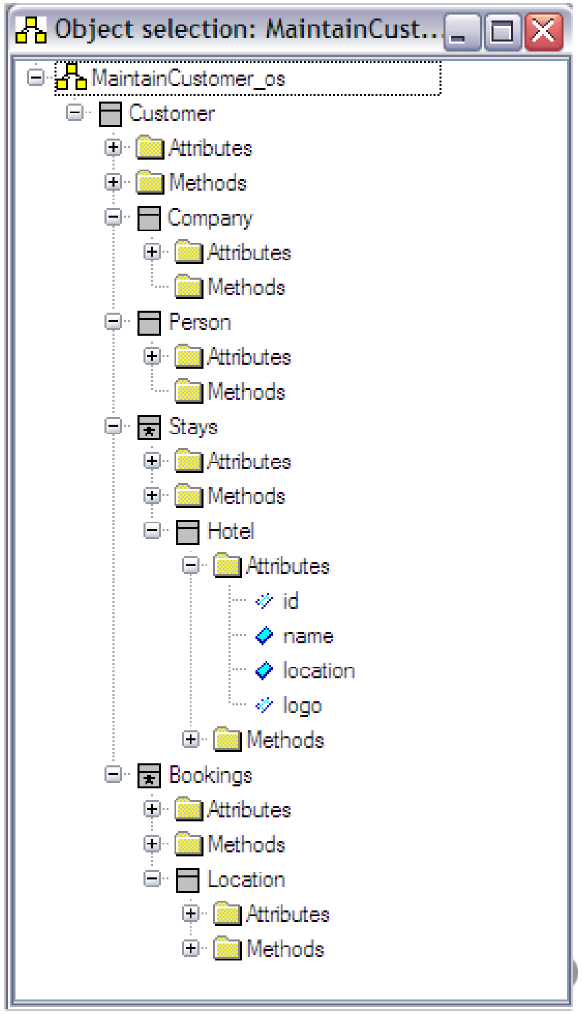
Jouteur

Object-Relation-Mapping

Database

Model View/Object selection

- Selection of classes from domain model
  - Solving a task
  - Organized as hierarchies
  - Object instances
  - Roles
- Foundation for
  - dialog model
  - server- and application code

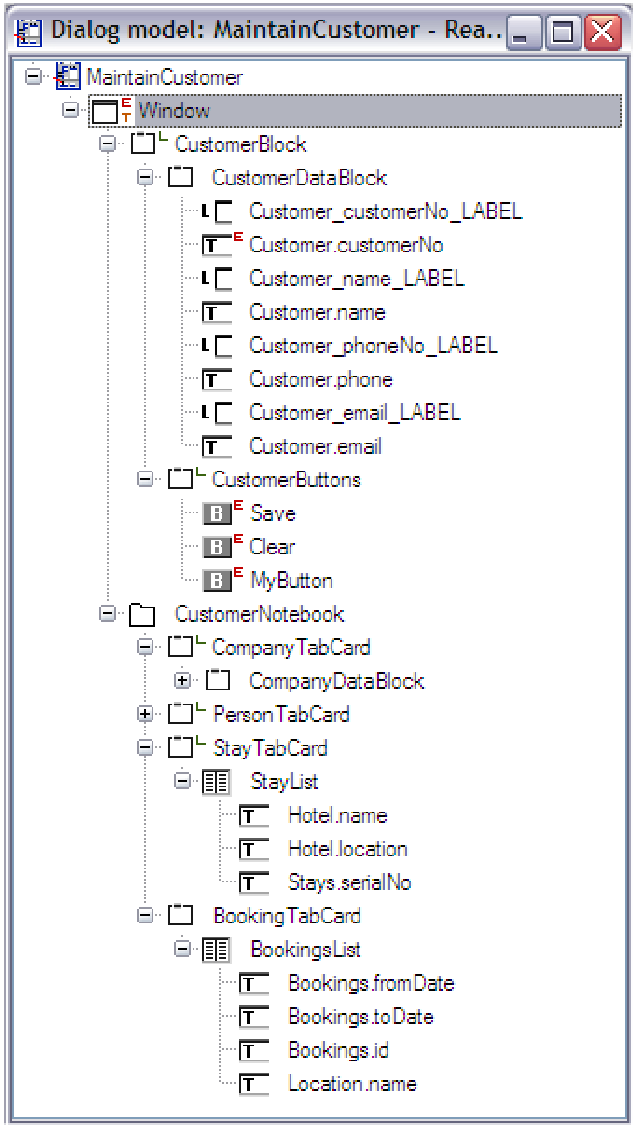


Object selection: MaintainCust...

- MaintainCustomer_os
  - Customer
    - Attributes
    - Methods
    - Company
      - Attributes
      - Methods
    - Person
      - Attributes
      - Methods
    - Stays
      - Attributes
      - Methods
      - Hotel
        - Attributes
          - id
          - name
          - location
          - logo
        - Methods
    - Bookings
      - Attributes
      - Methods
      - Location
        - Attributes
        - Methods

Dialog model

- Based on object selection
  - Hierarchy of user interface components
  - Events and actions
- Foundation for
  - generation of view and controller code

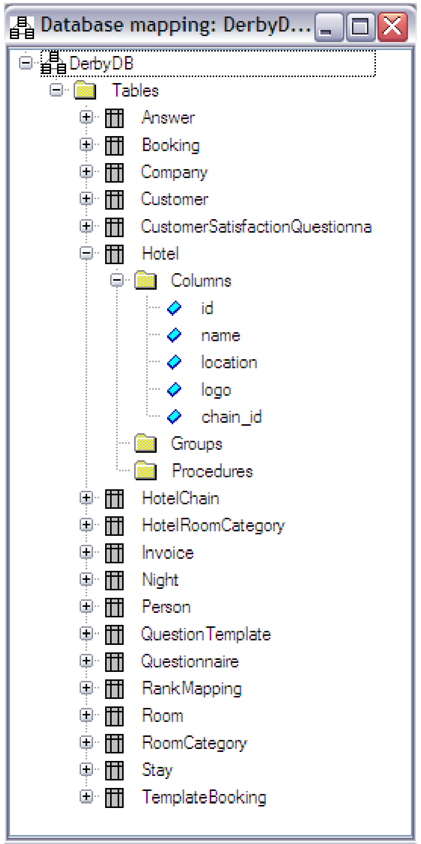


Dialog model: MaintainCustomer - Rea..

```
MaintainCustomer
  Window
    CustomerBlock
      CustomerDataBlock
        Customer_customerNo_LABEL
        Customer.customerNo
        Customer_name_LABEL
        Customer.name
        Customer_phoneNo_LABEL
        Customer.phone
        Customer_email_LABEL
        Customer.email
      CustomerButtons
        Save
        Clear
        MyButton
    CustomerNotebook
      CompanyTabCard
        CompanyDataBlock
      PersonTabCard
      StayTabCard
        StayList
          Hotel.name
          Hotel.location
          Stays.serialNo
      BookingTabCard
        BookingsList
          Bookings.fromDate
          Bookings.toDate
          Bookings.id
          Location.name
```

Database model

- Based on domain model
  - Persistent classes
  - Mapping logical attribute types to database types
  - Resolves OO structures
    - Implements heritage in RDBMS
- Foundation for
  - generation of database schema
  - generation of object relation mapping



Database mapping: DerbyD...

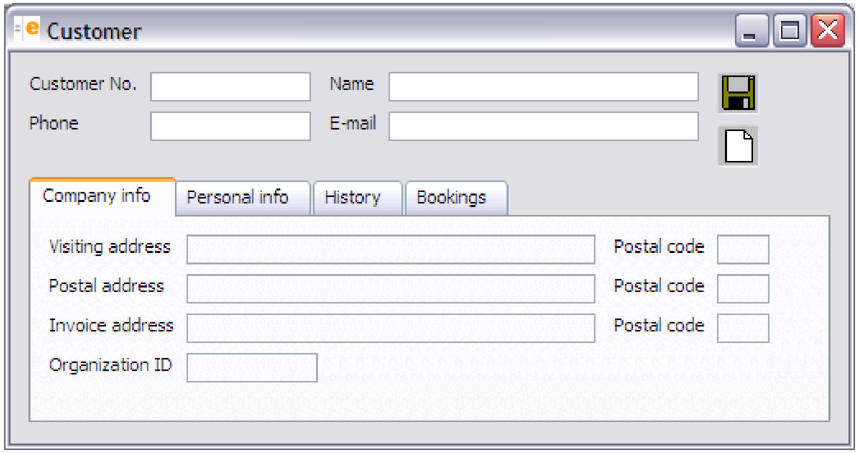
- DerbyDB
  - Tables
    - Answer
    - Booking
    - Company
    - Customer
    - CustomerSatisfactionQuestionna
    - Hotel
      - Columns
        - id
        - name
        - location
        - logo
        - chain_id
      - Groups
      - Procedures
    - HotelChain
    - HotelRoomCategory
    - Invoice
    - Night
    - Person
    - QuestionTemplate
    - Questionnaire
    - RankMapping
    - Room
    - RoomCategory
    - Stay
    - TemplateBooking

Contents

- Esito
- Applications built using Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets

Demo: Add dialog

- UML: add a dialog and synchronize
- Creating Model View (Object Selection)
- Modeling a dialog from scratch
- Generating code for the dialog
- Running the application

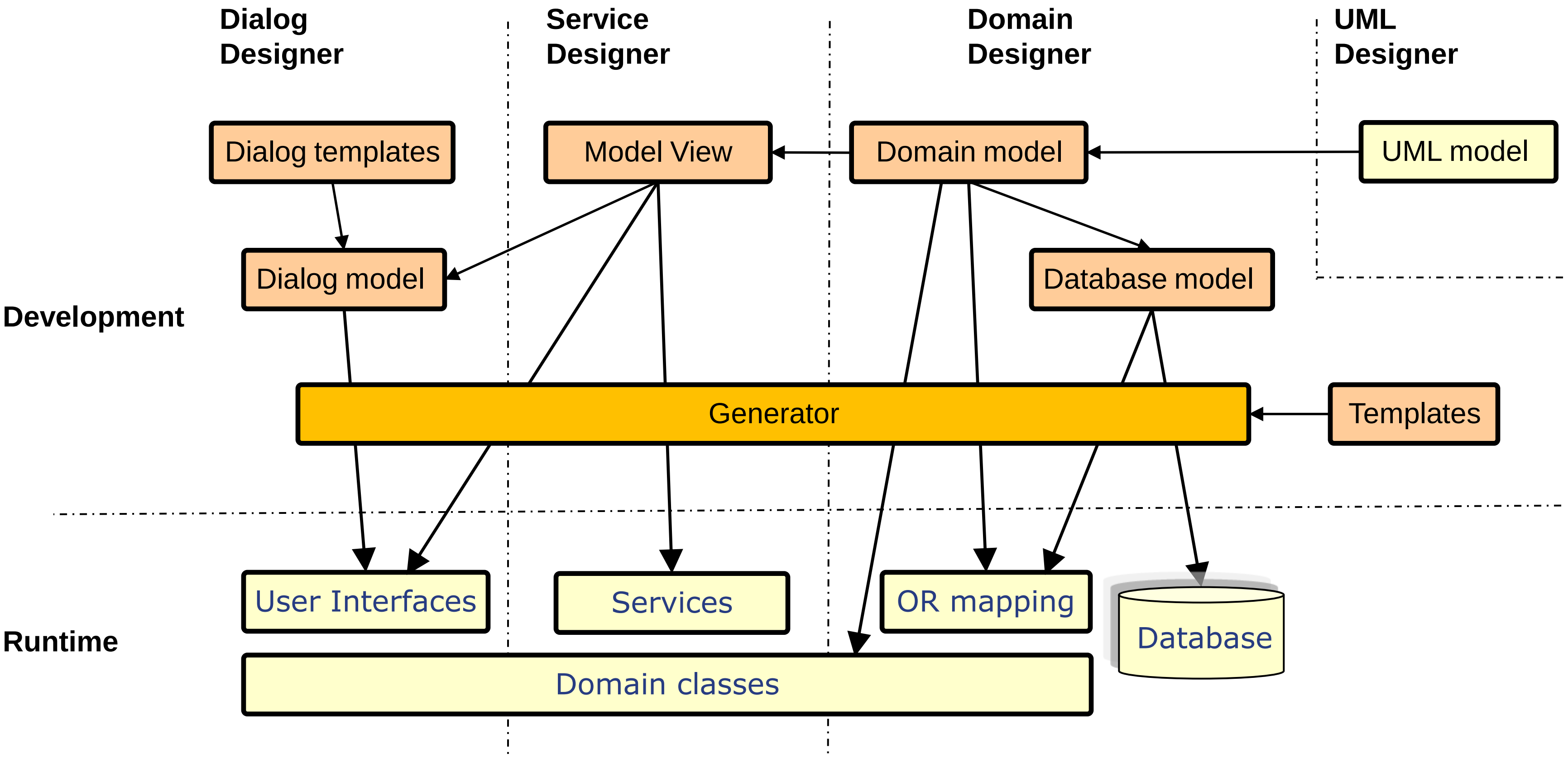


Customer

Customer No.		Name	
Phone		E-mail	

Visiting address		Postal code	
Postal address		Postal code	
Invoice address		Postal code	
Organization ID			

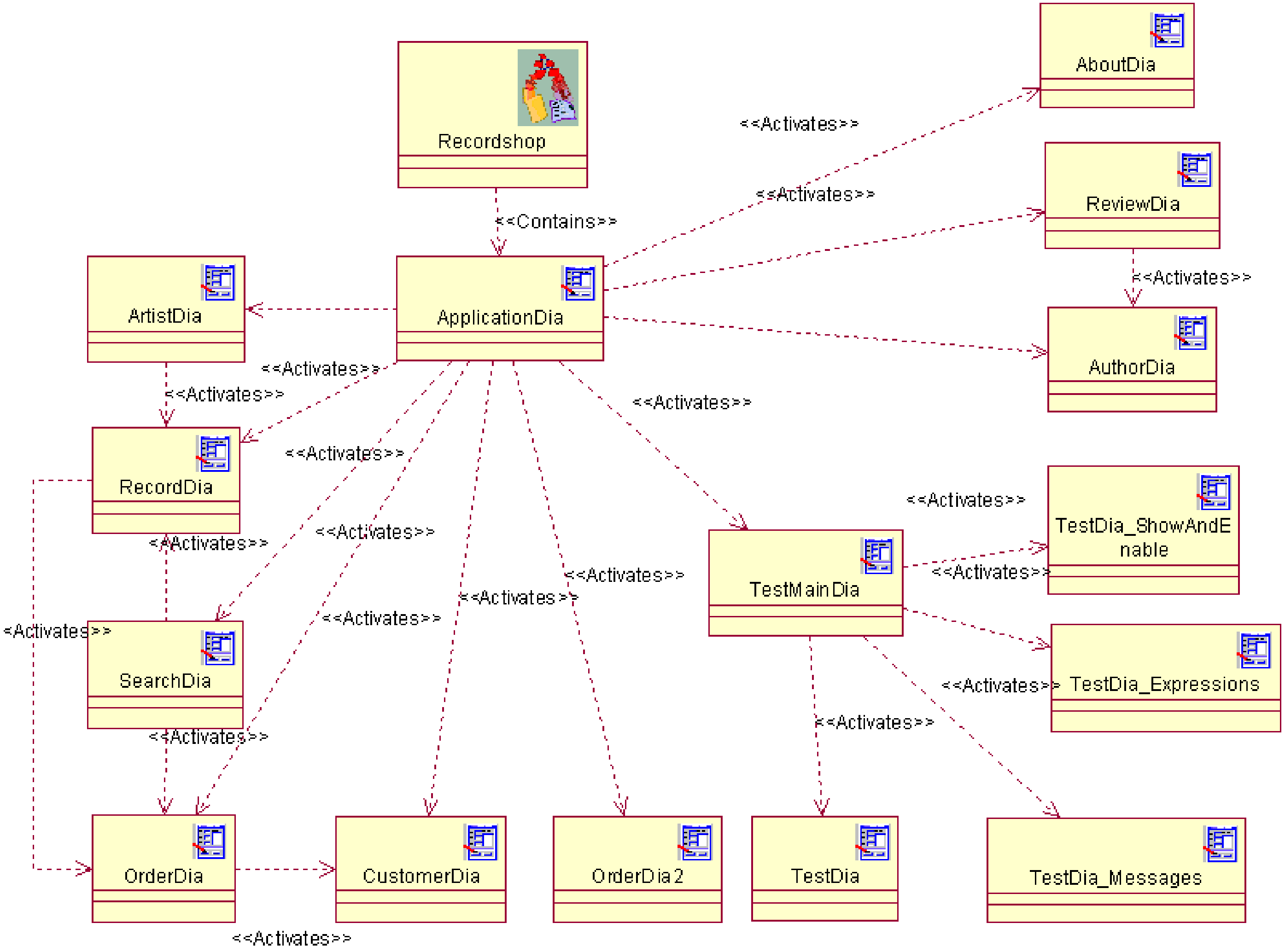
Genova architecture



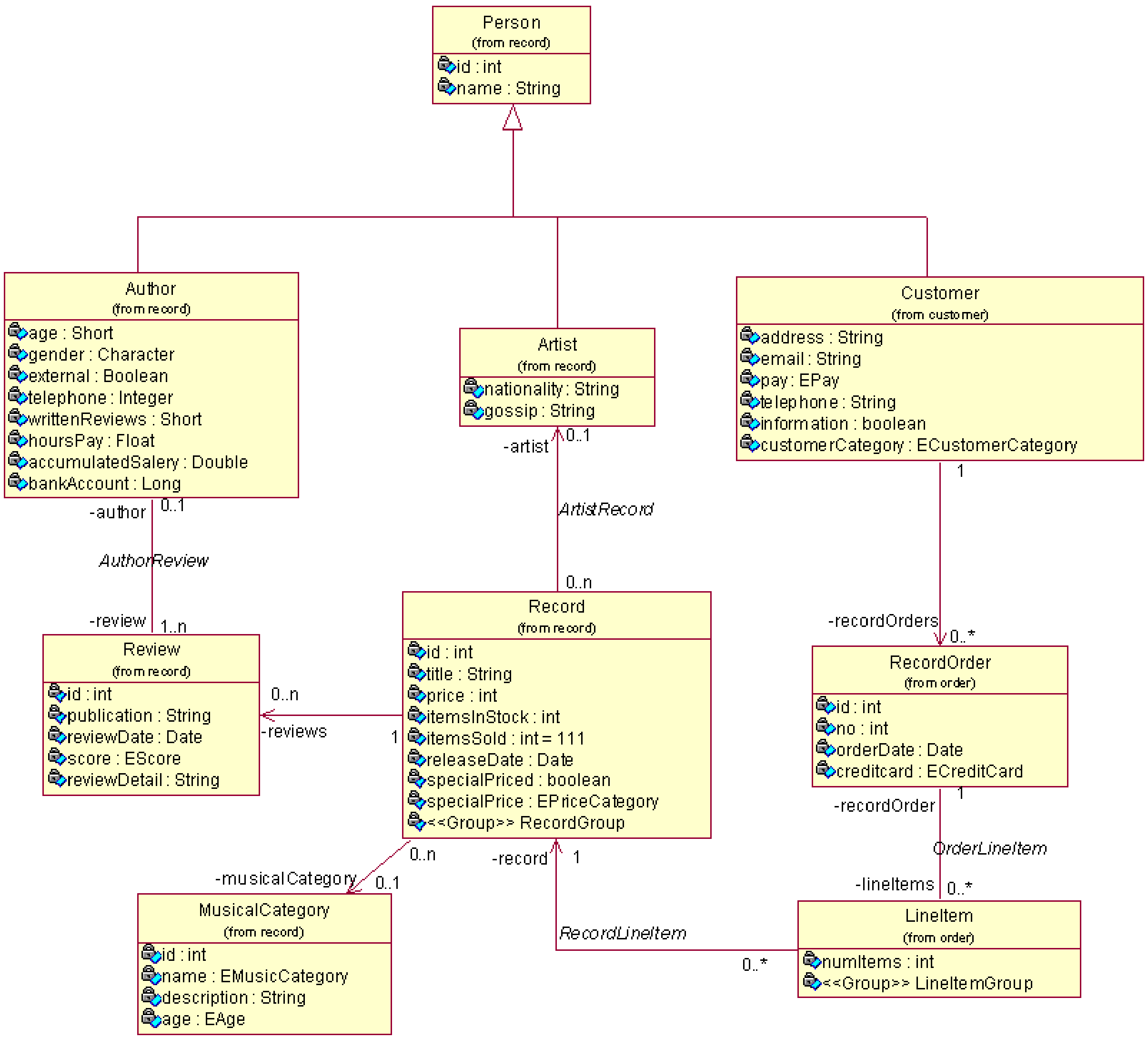
Genova UML profile

- Application Model
  - Class stereotypes
    - Application
    - Dialog
  - Dependency stereotypes
    - Contains
    - Activates
    - Required
- Domain Model
  - Class stereotypes
    - Enumeration
    - Group
    - Domain
    - Converter
  - Association stereotype
    - Association
- Attribute stereotypes
  - PropertyUI
  - PropertyDB
  - PropertyDBKey
  - EnumerationValue
- Generalization stereotypes
  - InheritedAttributes
  - DuplicateObjects
  - OneTable
  - SpecializedAttributes
- Operation stereotype
  - StoredProcedure
- Group Attribute stereotypes
  - Attribute
  - Group
  - Association

Application model



Domain model



Person
(from record)
id : int
name : String

Author
(from record)
age : Short
gender : Character
external : Boolean
telephone : Integer
writtenReviews : Short
hoursPay : Float
accumulatedSalery : Double
bankAccount : Long

Artist
(from record)
nationality : String
gossip : String

Customer
(from customer)
address : String
email : String
pay : EPay
telephone : String
information : boolean
customerCategory : ECustomerCategory

-author 0..1
AuthorReview
-review 1..n

-artist 0..1
ArtistRecord
0..n

1
-recordOrders 0..*

Review
(from record)
id : int
publication : String
reviewDate : Date
score : EScore
reviewDetail : String

0..n
-reviews
1

Record
(from record)
id : int
title : String
price : int
itemsInStock : int
itemsSold : int = 111
releaseDate : Date
specialPriced : boolean
specialPrice : EPriceCategory
<<Group>> RecordGroup

RecordOrder
(from order)
id : int
no : int
orderDate : Date
creditcard : ECreditCard

-recordOrder 1
OrderLineItem
-lineItems 0..*

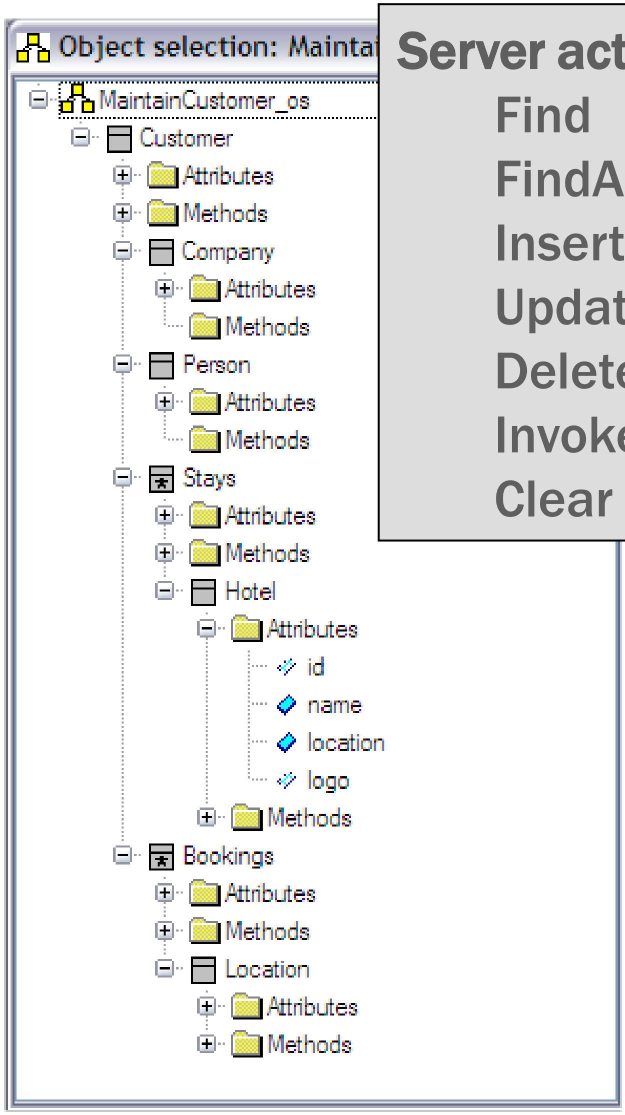
0..n
-musicalCategory 0..1

-record 1
RecordLineItem
0..*

MusicalCategory
(from record)
id : int
name : EMusicCategory
description : String
age : EAge

LineItem
(from order)
numItems : int
<<Group>> LineItemGroup

Client/server communication



Object selection: Maintai

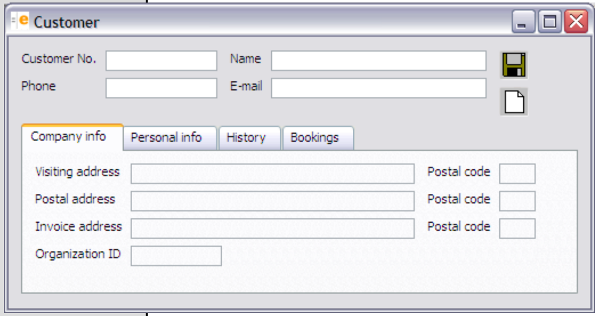
- MaintainCustomer_os
  - Customer
    - Attributes
    - Methods
    - Company
      - Attributes
      - Methods
    - Person
      - Attributes
      - Methods
    - Stays
      - Attributes
      - Methods
      - Hotel
        - Attributes
          - id
          - name
          - location
          - logo
        - Methods
    - Bookings
      - Attributes
      - Methods
      - Location
        - Attributes
        - Methods

Server actions:

- Find
- FindAll
- Insert
- Update
- Delete
- Invoke
- Clear

Client actions:

- Open
- Close
- Show
- Hide
- Enable
- Disable
- InsertRow
- RemoveRow
- ClearObject



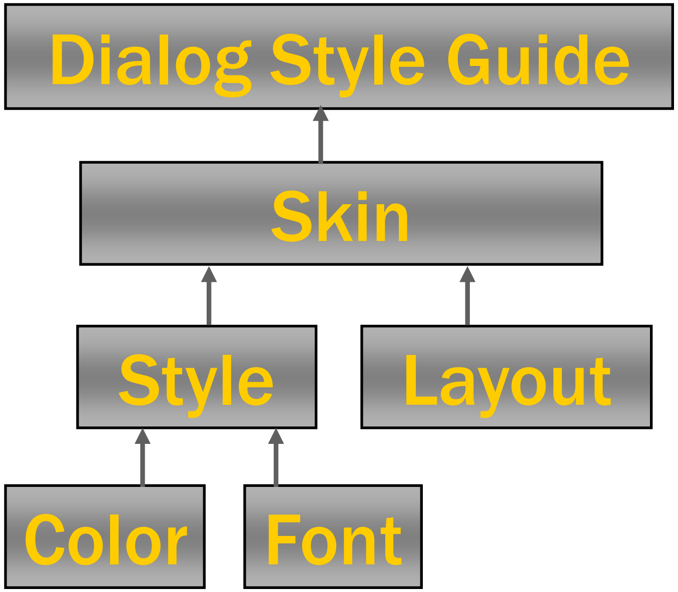
Customer

Customer No.		Name	
Phone		E-mail	

Visiting address		Postal code	
Postal address		Postal code	
Invoice address		Postal code	
Organization ID			

Dialog Resources

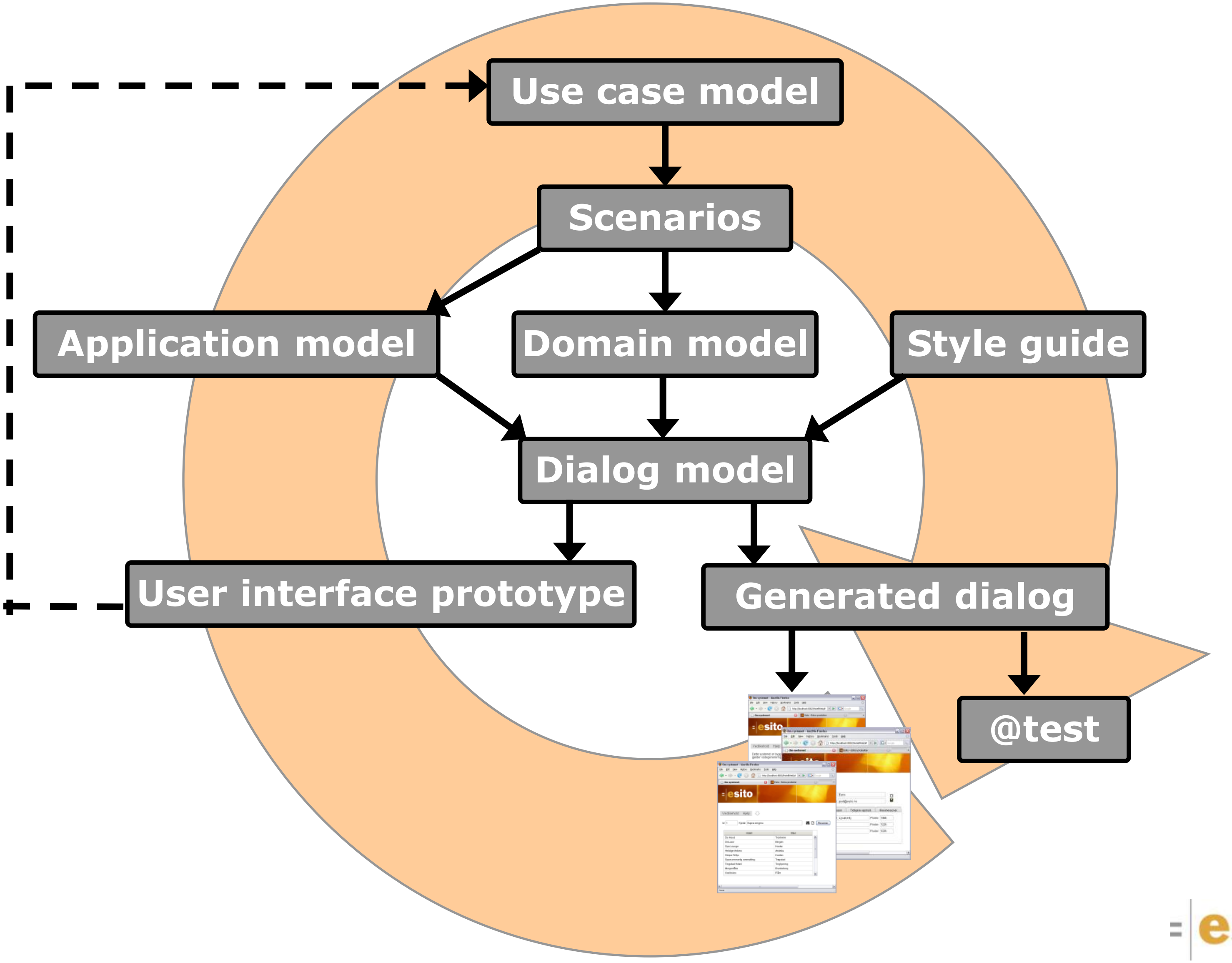
- Style Guide
  - Is a set of rules used in dialog model generation
  - Gives uniform appearance
- Changes to Style Guide properties propagate throughout the dialog model



Contents

- Esito
- Applications built using Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets

Process, methods and techniques



Contents

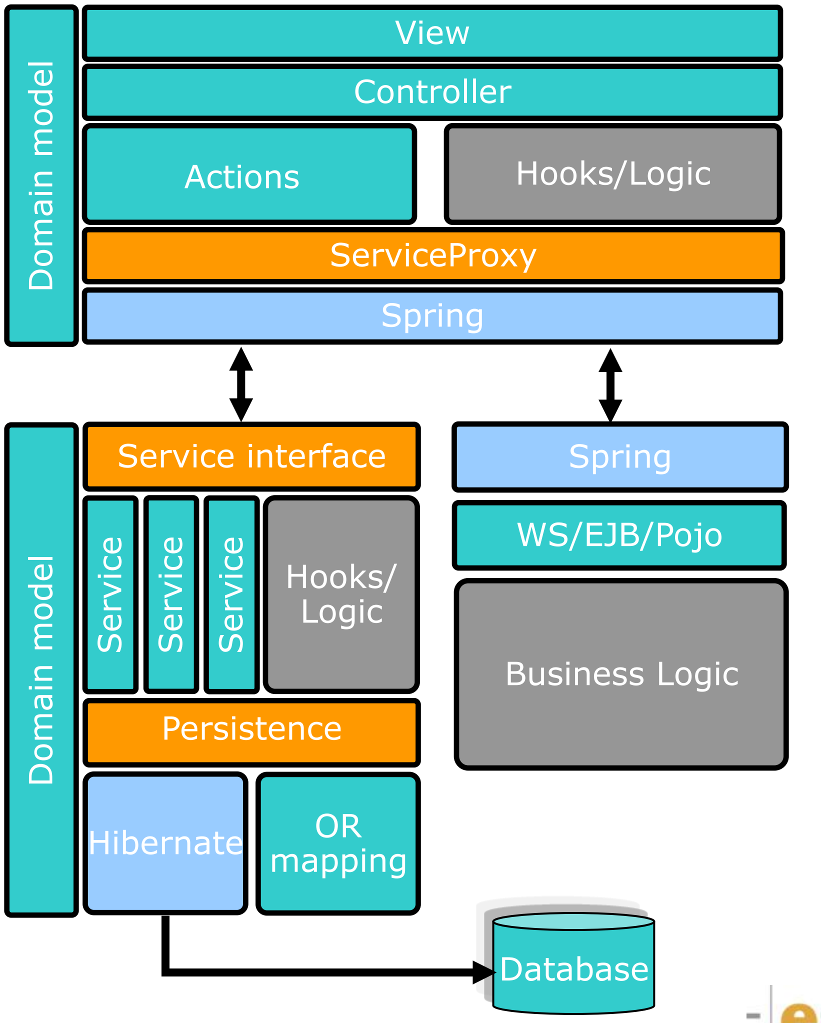
- Esito
- Applications built using Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets

Demo: ICEfaces/JSF

- Look at SpinRecords
- Add code
- Regenerating code for the dialogs
- Running the application
- Code analysis

Application architecture - Java

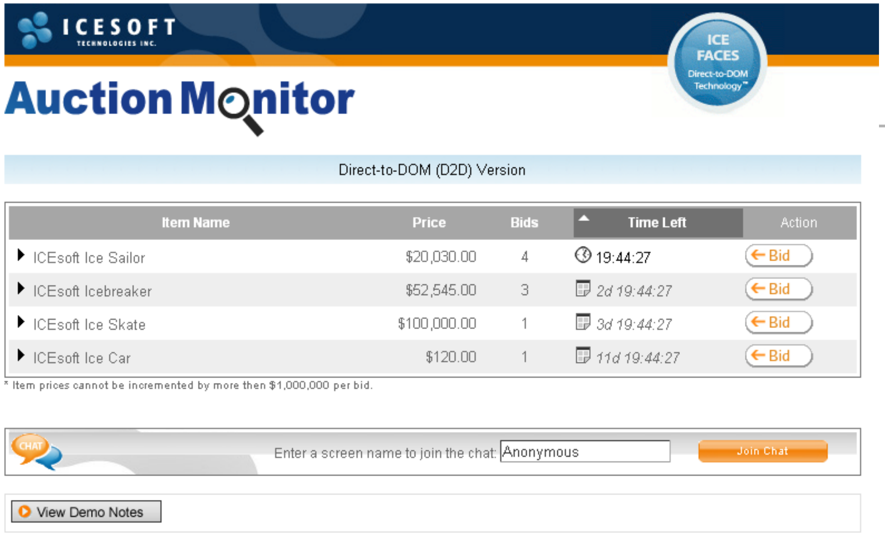
- 🟦 **Generated by Genova**
- 🟧 **Genova framework**
- 🟦 **3. party framework**
- ⬜ **Business Logic**



esito

ICEfaces

- Java Server Faces
- Rich User Experience
- Open Source
- Standards-based
- Easy Ajax



ICESOFT TECHNOLOGIES INC.

Auction Monitor

ICE FACES Direct-to-DOM Technology™

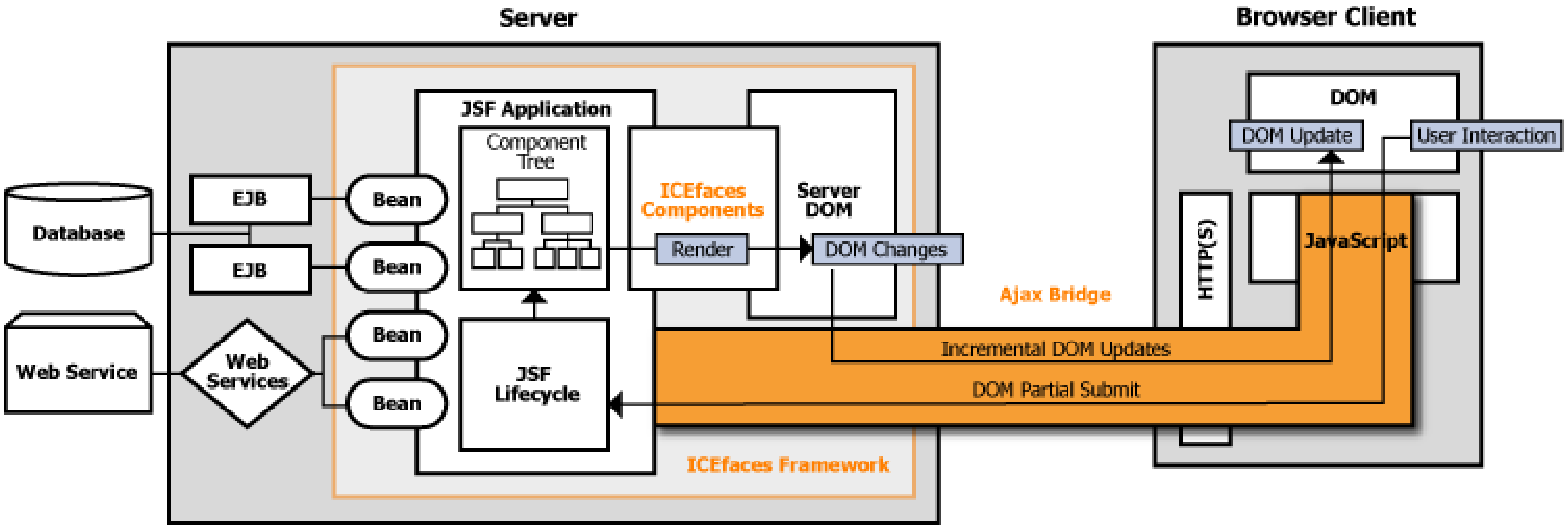
Direct-to-DOM (D2D) Version

Item Name	Price	Bids	▲ Time Left	Action
▸ ICEsoft Ice Sailor	$20,030.00	4	19:44:27	← Bid
▸ ICEsoft Icebreaker	$52,545.00	3	*2d 19:44:27*	← Bid
▸ ICEsoft Ice Skate	$100,000.00	1	*3d 19:44:27*	← Bid
▸ ICEsoft Ice Car	$120.00	1	*11d 19:44:27*	← Bid

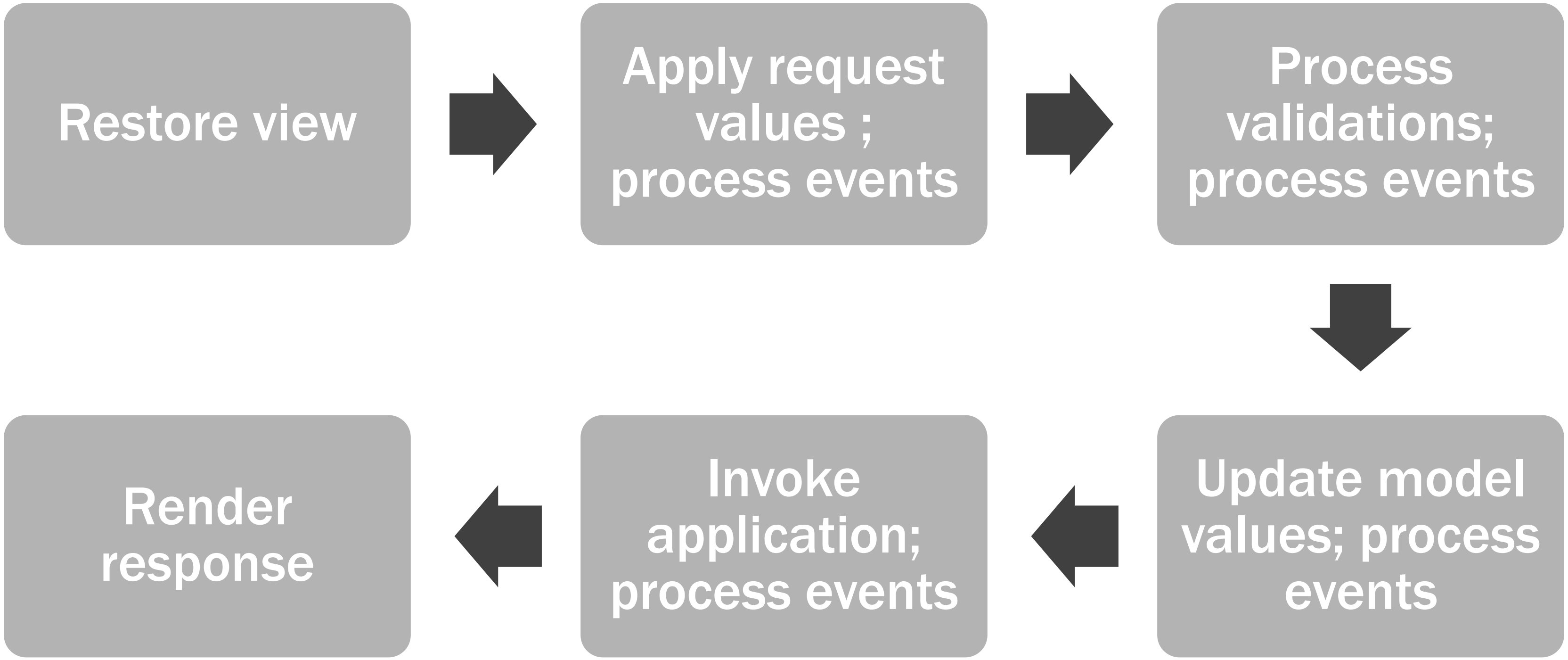
* Item prices cannot be incremented by more then $1,000,000 per bid.

Enter a screen name to join the chat: Anonymous [Join Chat]

[View Demo Notes]



JSF: Lifecycle



Restore view

Apply request values ; process events

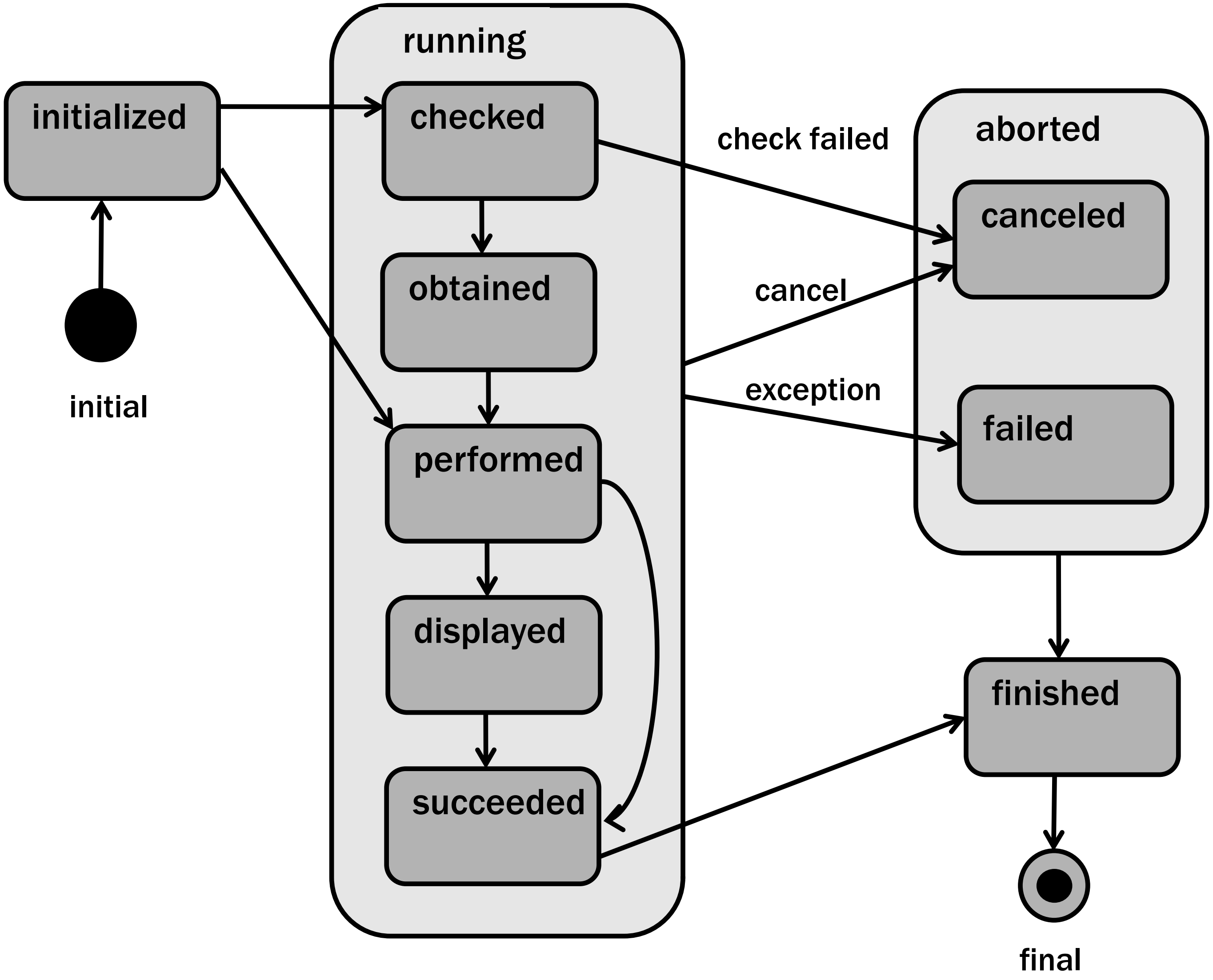
Process validations; process events

Update model values; process events

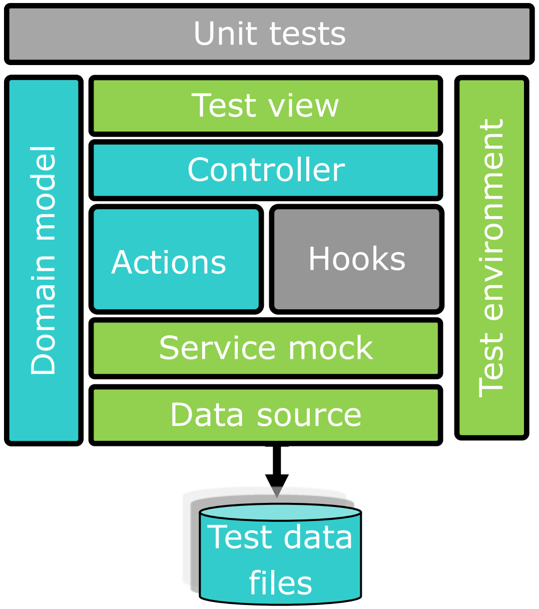
Invoke application; process events

Render response

Genova action life cycle



Unit test architecture of Jouteur



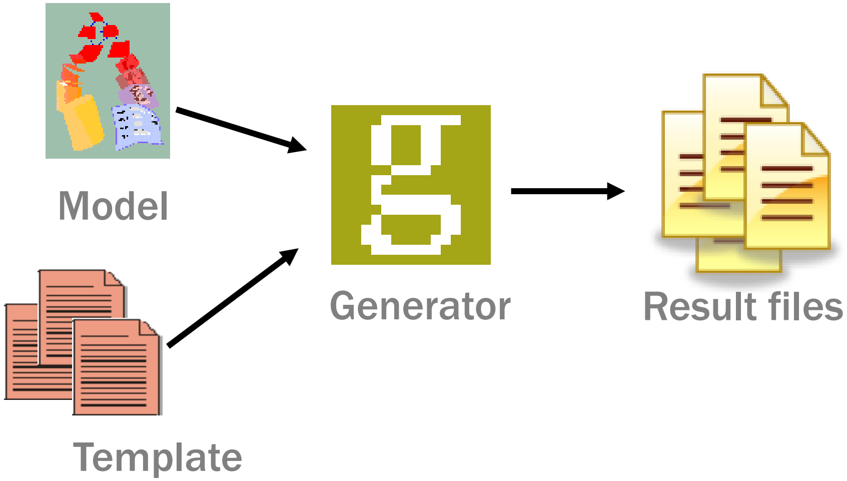
- 🟦 **Generated by Genova**
- ⬜ **Hand-written code**
- 🟩 **Test tools provided by Genova**

Contents

- Esito
- Applications built using Genova
- Genova designers and models
- The Genova tool architecture
- Development method
- Developing running code
  - Application architecture
  - Add code and business logic
  - Test architecture
- Genova template language and targets

Code generator concept

- Generate from models
  - Domain model
  - Object selection
  - Dialog model
  - Database model
- Template based
  - A template consists of directives, substitution variables, macros, expressions, functions and text
- The result is a set of text files



Generator targets

- Dialog Designer
  - Dialog Report (HTML)
  - Java/JFC (Swing)
  - Jasper Report
  - C++/MFC
  - Visual Basic
  - Web/ASP
  - Web/JSP
  - Sysdul/ddf
  - Java/ICEfaces (JSF)
  - Jouteur
- Service Designer
  - Java/EJB
  - Java2XML
  - Sysdul
  - Sysdul2XML
- Domain Designer
  - Java
  - Jouteur
- Data Access
  - Hibernate
  - Genova Data Objects
  - Div EJB2 versions
  - XmlSchema
  - Sysdul
- Database
  - Oracle
  - Sybase
  - SQL Server
  - MySQL
  - Derby
  - Mimer
  - XMLSchema

Template example

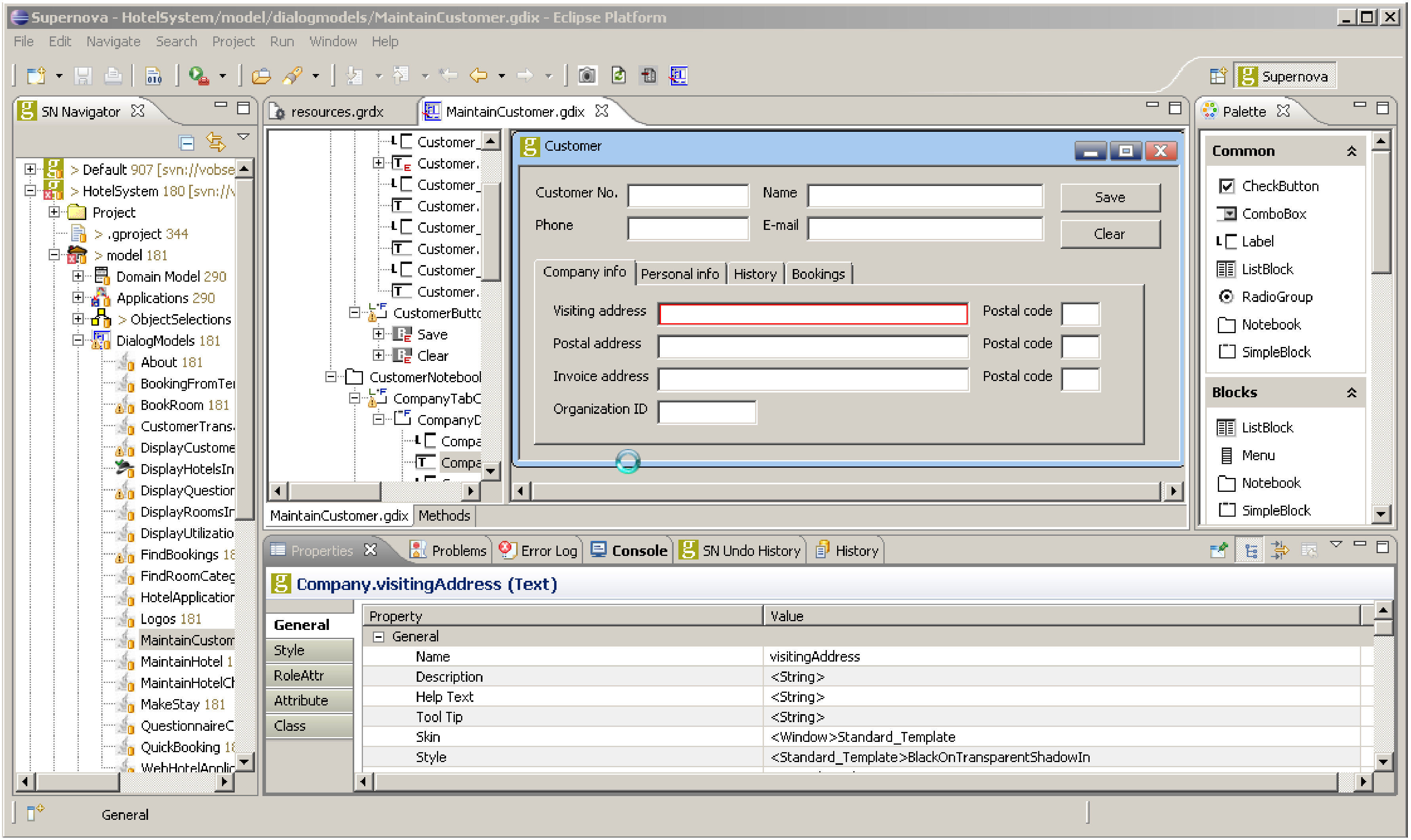
```
%FILE% events.txt
%ITERATE:DialogModel%
    Dialog: %Name%
    Events:
    %ITERATE:DialogObject%
        %Name%
        %ITERATE:Event%
            %EventType%
        %ENDITERATE%
    %ENDITERATE%
%ENDITERATE%
%ENDFILE%
```

- Using the dialog model
- The template prints out the name, of all dialogs, dialog objects and events to the specified file “events.txt“
- Directives:
  - %FILE%, %ENDFILE%
  - %ITERATE%, %ENDITERATE%
- Substitution variables:
  - %Name%, %EventType%
- Plain text:
  - “Dialog:”, “Events: “

In the near future

- Genova as Eclipse plug-in
- Extended automatic Junit testing
- More Genova UML models
  - Components with model views (Object Selection)
  - Components realizes interfaces

Genova Eclipse plug-in



Company.visitingAddress (Text)

Property	Value
⊟ General	
Name	visitingAddress
Description	<String>
Help Text	<String>
Tool Tip	<String>
Skin	<Window>Standard_Template
Style	<Standard_Template>BlackOnTransparentShadowIn

esito

Questions

www.esito.no