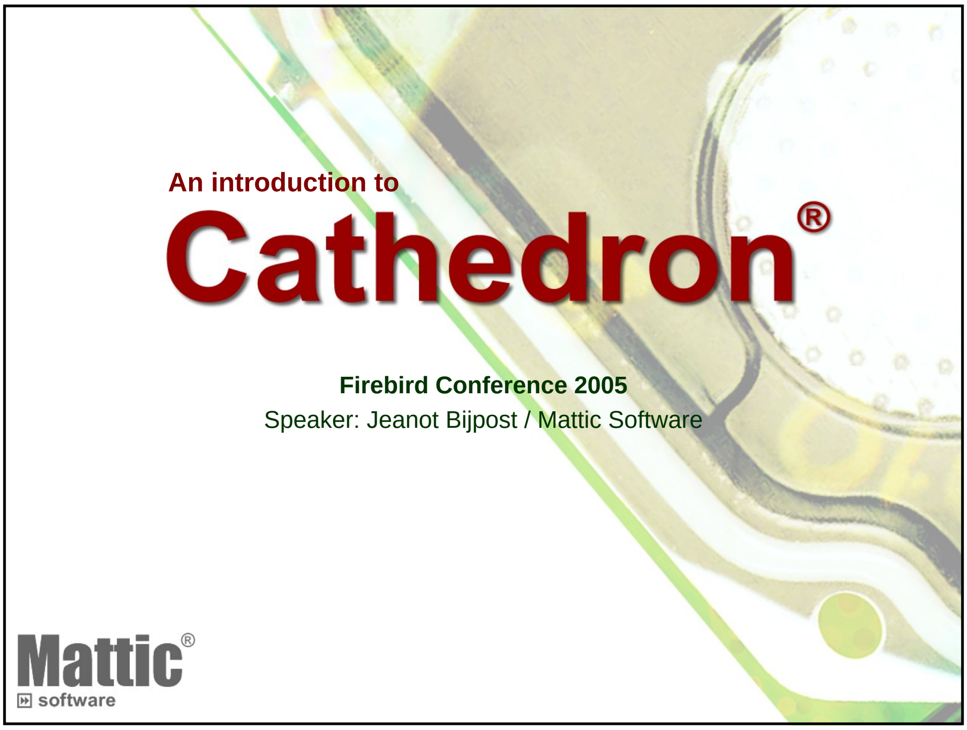


An introduction to

Cathedron®

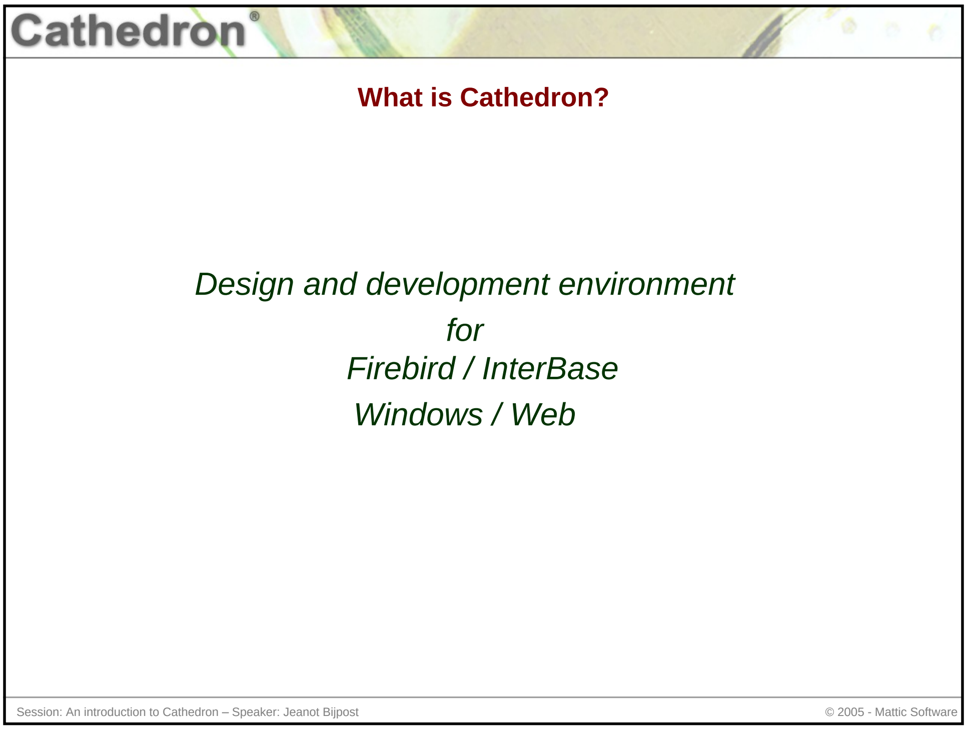
Firebird Conference 2005

Speaker: Jeanot Bijpost / Mattic Software

Mattic® software

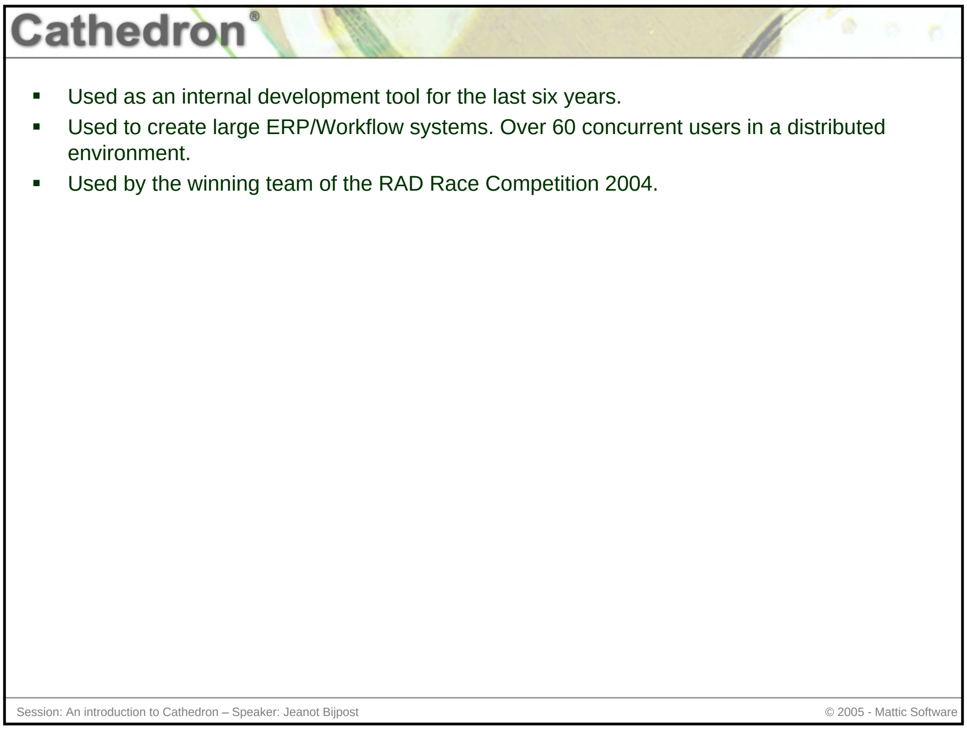
Cathedron®

- Duration of the original presentation: 30 minutes.
- More information can be found on www.cathedron.com.



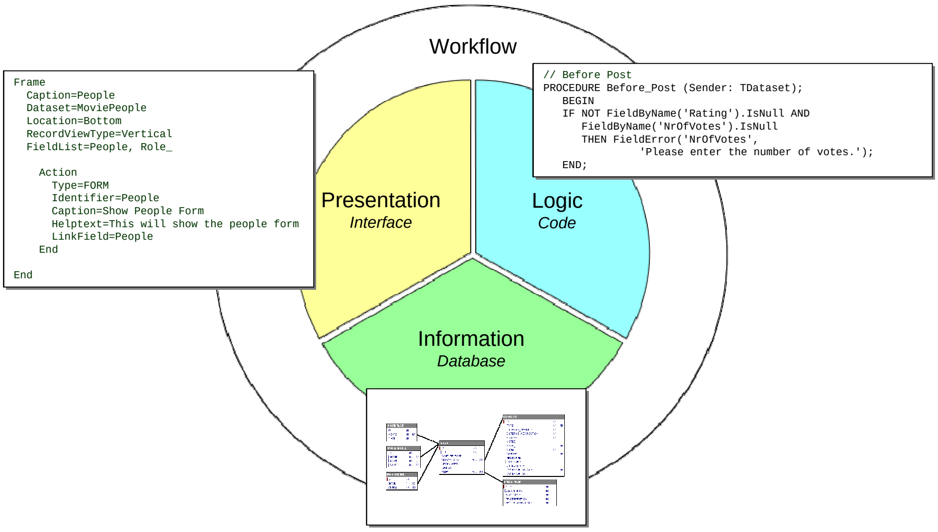
What is Cathedron?

Design and development environment
for
Firebird / InterBase
Windows / Web



- Used as an internal development tool for the last six years.
- Used to create large ERP/Workflow systems. Over 60 concurrent users in a distributed environment.
- Used by the winning team of the RAD Race Competition 2004.

The Big Picture



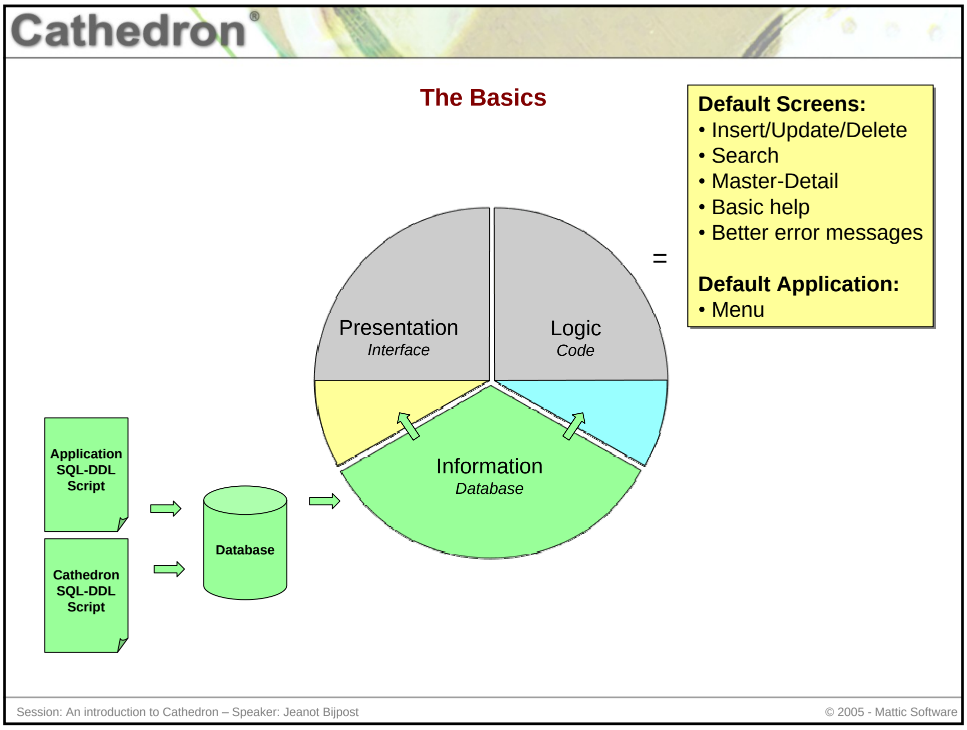
```
Frame
  Caption=People
  Dataset=MoviePeople
  Location=Bottom
  RecordViewType=Vertical
  FieldList=People, Role_

    Action
      Type=FORM
      Identifier=People
      Caption=Show People Form
      Helptext=This will show the people form
      LinkField=People
    End

End
```

```
// Before Post
PROCEDURE Before_Post (Sender: TDataset);
   BEGIN
   IF NOT FieldByName('Rating').IsNull AND
      FieldByName('NrOfVotes').IsNull
      THEN FieldError('NrOfVotes',
               'Please enter the number of votes.');
   END;
```

- The core of an information system can be seen as the combination of “Information”, “Presentation” and “Logic”.
- The workflow layer is an optional addition spanning the whole core.

In Cathedron:

- Information:
  - In the current version of Cathedron the information model is ‘reverse engineered’ from the database. (You create a database using SQL-DDL or some other tool, Cathedron will examine the rdb$... meta tables and reconstruct a ERD/UML-like model from this structure).
  - In future versions we provide an editable ERD/UML model and construct/change the database from this model.
- Logic:
  - Logic is written using a procedural third generation language (Pascal).
- Presentation:
  - Interfaces are designed using a high-level interface definition language.

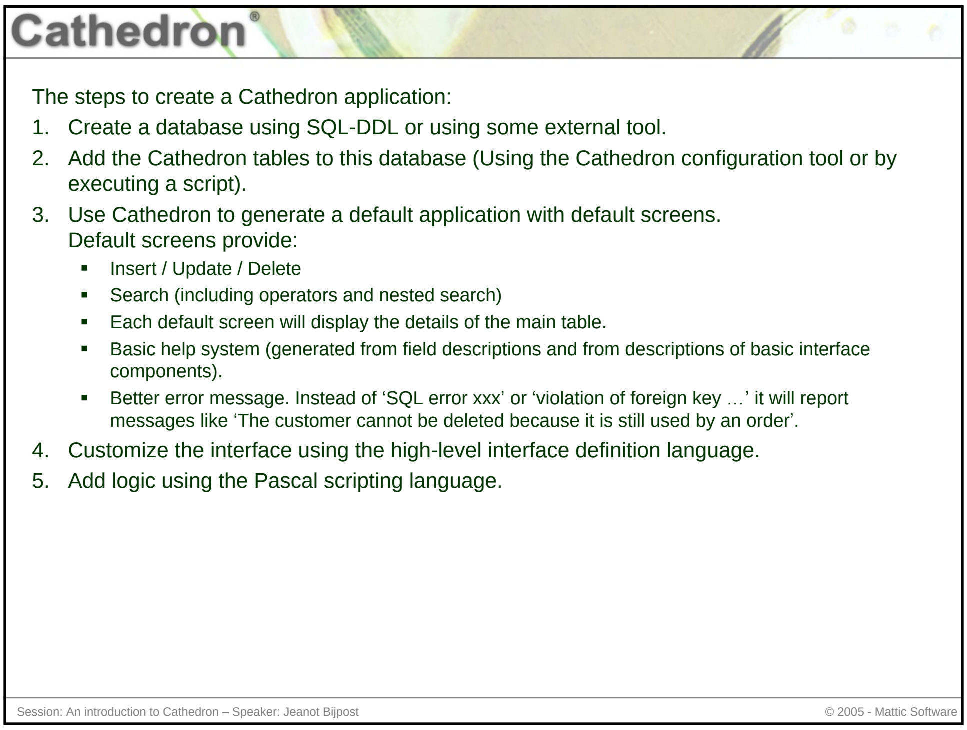


Cathedron®

The Basics

Default Screens:
- Insert/Update/Delete
- Search
- Master-Detail
- Basic help
- Better error messages

Default Application:
- Menu

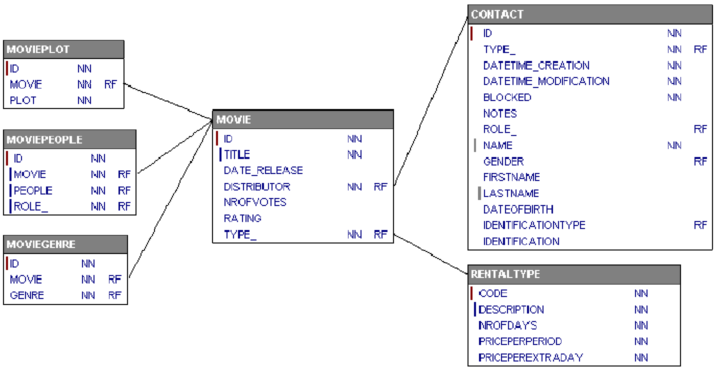


Cathedron®

The steps to create a Cathedron application:

1. Create a database using SQL-DDL or using some external tool.
2. Add the Cathedron tables to this database (Using the Cathedron configuration tool or by executing a script).
3. Use Cathedron to generate a default application with default screens.
   Default screens provide:
   - Insert / Update / Delete
   - Search (including operators and nested search)
   - Each default screen will display the details of the main table.
   - Basic help system (generated from field descriptions and from descriptions of basic interface components).
   - Better error message. Instead of ‘SQL error xxx’ or ‘violation of foreign key …’ it will report messages like ‘The customer cannot be deleted because it is still used by an order’.
4. Customize the interface using the high-level interface definition language.
5. Add logic using the Pascal scripting language.

Information/Structure/Database Model



MOVIEPLOT		
ID	NN	
MOVIE	NN	RF
PLOT	NN	

MOVIEPEOPLE		
ID	NN	
MOVIE	NN	RF
PEOPLE	NN	RF
ROLE_	NN	RF

MOVIEGENRE		
ID	NN	
MOVIE	NN	RF
GENRE	NN	RF

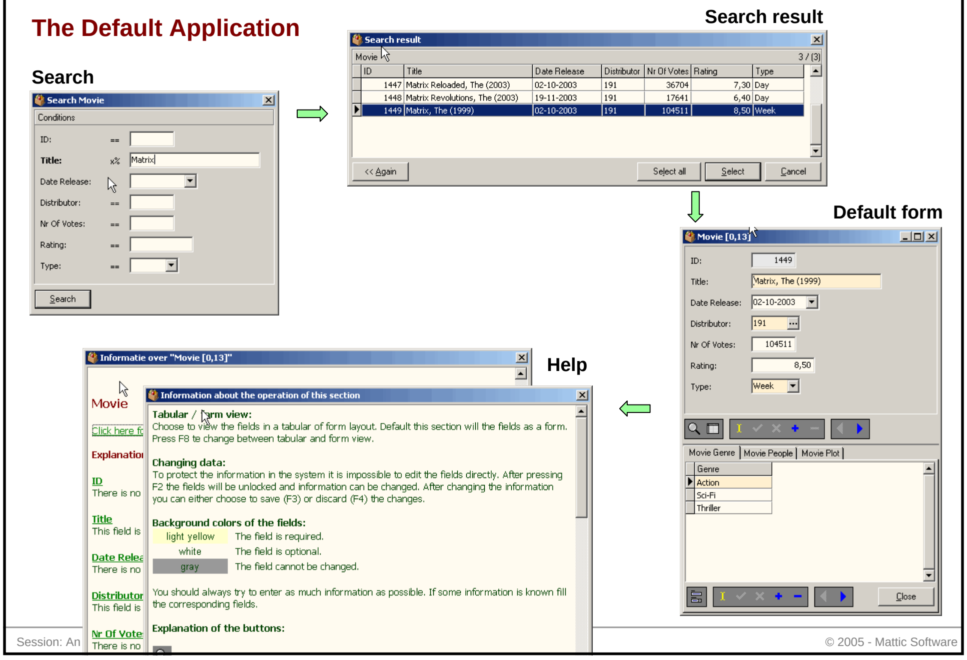
MOVIE		
ID	NN	
TITLE	NN	
DATE_RELEASE		
DISTRIBUTOR	NN	RF
NROFVOTES		
RATING		
TYPE_	NN	RF

CONTACT		
ID	NN	
TYPE_	NN	RF
DATETIME_CREATION	NN	
DATETIME_MODIFICATION	NN	
BLOCKED	NN	
NOTES		
ROLE_		RF
NAME	NN	
GENDER		RF
FIRSTNAME		
LASTNAME		
DATEOFBIRTH		
IDENTIFICATIONTYPE		RF
IDENTIFICATION		

RENTALTYPE	
CODE	NN
DESCRIPTION	NN
NROFDAYS	NN
PRICEPERPERIOD	NN
PRICEPEREXTRADAY	NN

- This model displays the structure of the example that is used in the following slides.
- The example is a simplified application with some example data from the Internet Movie Database (www.imdb.com).

The Default Application



Search

Search Movie

Conditions

ID:	==	
Title:	x%	Matrix
Date Release:		
Distributor:	==	
Nr Of Votes:	==	
Rating:	==	
Type:	==	

Search result

Search result

Movie 3 / (3)

ID	Title	Date Release	Distributor	Nr Of Votes	Rating	Type
1447	Matrix Reloaded, The (2003)	02-10-2003	191	36704	7,30	Day
1448	Matrix Revolutions, The (2003)	19-11-2003	191	17641	6,40	Day
1449	Matrix, The (1999)	02-10-2003	191	104511	8,50	Week

Default form

Movie [0,13]

ID:	1449
Title:	Matrix, The (1999)
Date Release:	02-10-2003
Distributor:	191
Nr Of Votes:	104511
Rating:	8,50
Type:	Week

Genre
Action
Sci-Fi
Thriller

Help

Informatie over "Movie [0,13]"

Movie

Click here fo

Explanatio

ID
There is no

Title
This field is

Date Relea
There is no

Distributor
This field is

Nr Of Vote
There is no

Information about the operation of this section

Tabular / form view:
Choose to view the fields in a tabular of form layout. Default this section will the fields as a form. Press F8 te change between tabular and form view.

Changing data:
To protect the information in the system it is impossible to edit the fields directly. After pressing F2 the fields will be unlocked and information can be changed. After changing the information you can either choose to save (F3) or discard (F4) the changes.

Background colors of the fields:

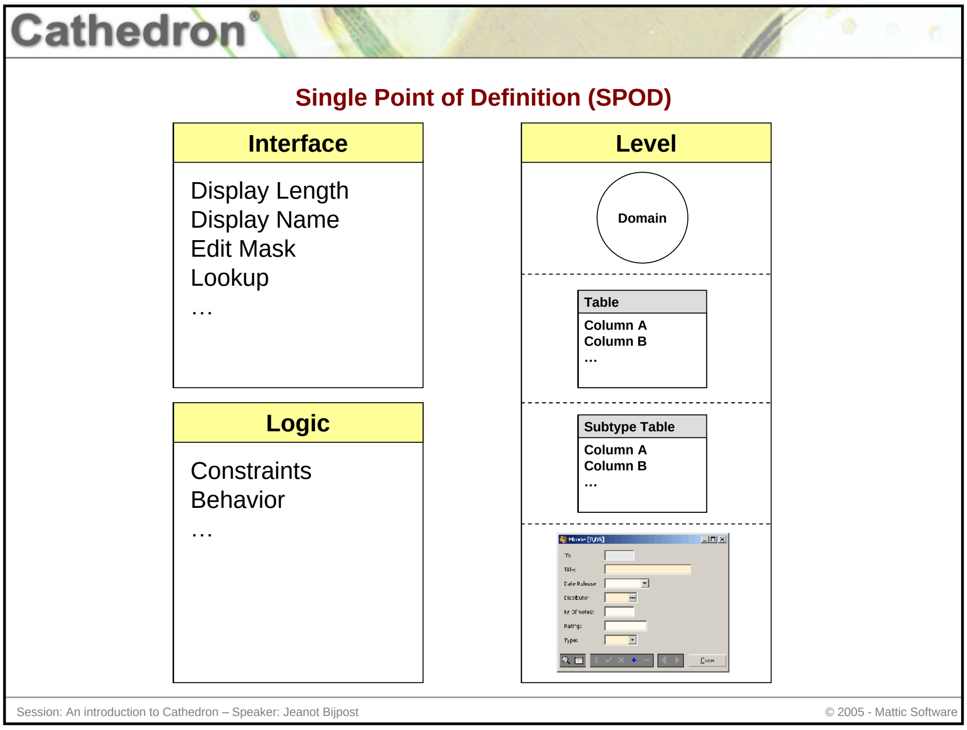
light yellow	The field is required.
white	The field is optional.
gray	The field cannot be changed.

You should always try to enter as much information as possible. If some information is known fill the corresponding fields.

Explanation of the buttons:

Starting in the upper left corner following the arrows:

- We open the movie screen, press F5 to open the search dialog and enter the phrase ‘matrix’ in the edit for the movie title.
  - Multiple arguments can be entered. Before the edit you will find a button indicating the selected operator. As you can see each field has a default operator.
- Cathedron will show the search results in a separate screen. We can either select all movies or just a single movie.
- After selecting a movie Cathedron displays the information in the default screen.
  - Field colors indicate the status of a field: gray = read only, light yellow = required, white = optional.
  - Cathedron scans the data types and tries to link appropriate controls to the fields. As a result financial fields will have a calculator, date fields will have a date time picker etc.
  - The type field uses a combo box with the values ‘week’ and ‘day’. After scanning the database Cathedron found out this column was referencing the ‘rentaltype’ table. As you can see in the database diagram this table contains two fields: code and description. Code is the primary key and description the secondary. Because of this structure Cathedron created a combo box and cached the values of this combo box to prevent unwanted network traffic.
  - The distributor references the contact table. In the example database this table contains over 10.000 records. The primary key of contact is the ID field, there is no secondary key. As a result Cathedron ‘did not know’ how to display a contact other that using its ID.
- Before we fix the contact field and extend the application there is some more theory…



Single Point of Definition (SPOD)

- Single point of definition = Avoid Redundancy.

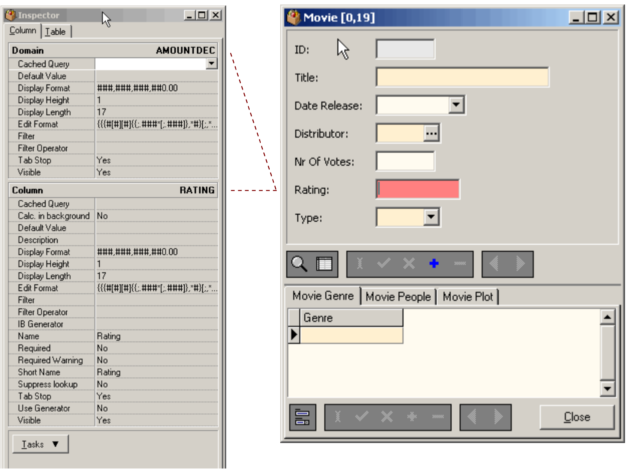
- For databases this means a redundancy free (normalized) structures.
- For a ‘normal’ programming language this means creating classes to reuse code.
- For Cathedron this means ‘apply a rule or definition on the highest applicable level’.

- Within Cathedron logic and interface properties can be set at three different levels:
  - Domain
  - Table/Column (including subtypes)
  - Screen

- In the following slides we will make adjustments to the application using a property inspector…

Cathedron®

Interface Inspector



Inspector

Column | Table

Domain	AMOUNTDEC
Cached Query	
Default Value	
Display Format	###,###,###,##0.00
Display Height	1
Display Length	17
Edit Format	{{{#[#][#]{{;,###*[;,###]},*#}[;,*...
Filter	
Filter Operator	
Tab Stop	Yes
Visible	Yes

Column	RATING
Cached Query	
Calc. in background	No
Default Value	
Description	
Display Format	###,###,###,##0.00
Display Height	1
Display Length	17
Edit Format	{{{#[#][#]{{;,###*[;,###]},*#}[;,*...
Filter	
Filter Operator	
IB Generator	
Name	Rating
Required	No
Required Warning	No
Short Name	Rating
Suppress lookup	No
Tab Stop	Yes
Use Generator	No
Visible	Yes

Using the inspector we can change interface-related properties of domains, columns and tables. Such as displaynames, editmasks, lookups etc.

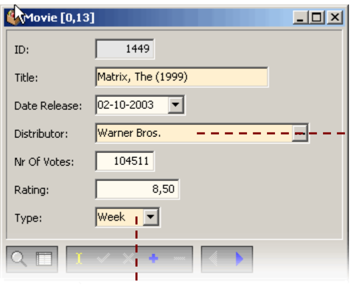
In this example the rating field is selected in the form and the inspector shows us the properties for the rating column and its domain ‘AmountDec’.

Cathedron®

- Example: The rating of a movie should be displayed using only one decimal instead of two.

- The rating appears to be defined as an AmountDec. Now we have four options to make this change:
  - Change the display format of the AmountDec domain. Thereby forcing Cathedron to display every column that has the AmountDec domain to display with a single decimal. - Two decimals seems a good default for a normal decimal field, so we will reject this option.
  - Change the display format of the Rating field. Thereby forcing Cathedron to display the rating column of the movie table with a single decimal (this will be enforced in every screen displaying a movie). - This seems a reasonable option.
  - Create a new domain for ratings, change the rating column to use this domain and apply a display format to this domain. - Probably more elegant compared to the second option.
  - Override the display format using the interface scripting language. (Cannot be done using the inspector). - This is a bad option. If we display the rating in an other screen we would have to override the display format again.

Lookups



Movie [0,13]

ID: 1449

Title: Matrix, The (1999)

Date Release: 02-10-2003

Distributor: Warner Bros.

Nr Of Votes: 104511

Rating: 8,50

Type: Week

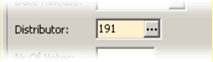
Lookups are a mechanism to change the way a foreign key field is displayed. The can be assigned to a domain or field using the inspector.

Manually added lookup:

```
SELECT Name
FROM Contact
WHERE ID=:KeyValue
```

This lookup is called a live lookup. Every time we display a distributor the name will be retrieved and displayed instead of the ID. (Once retrieved it will be temporarily cached within the form).

Without this lookup the field would simply display the ID of the distributor:



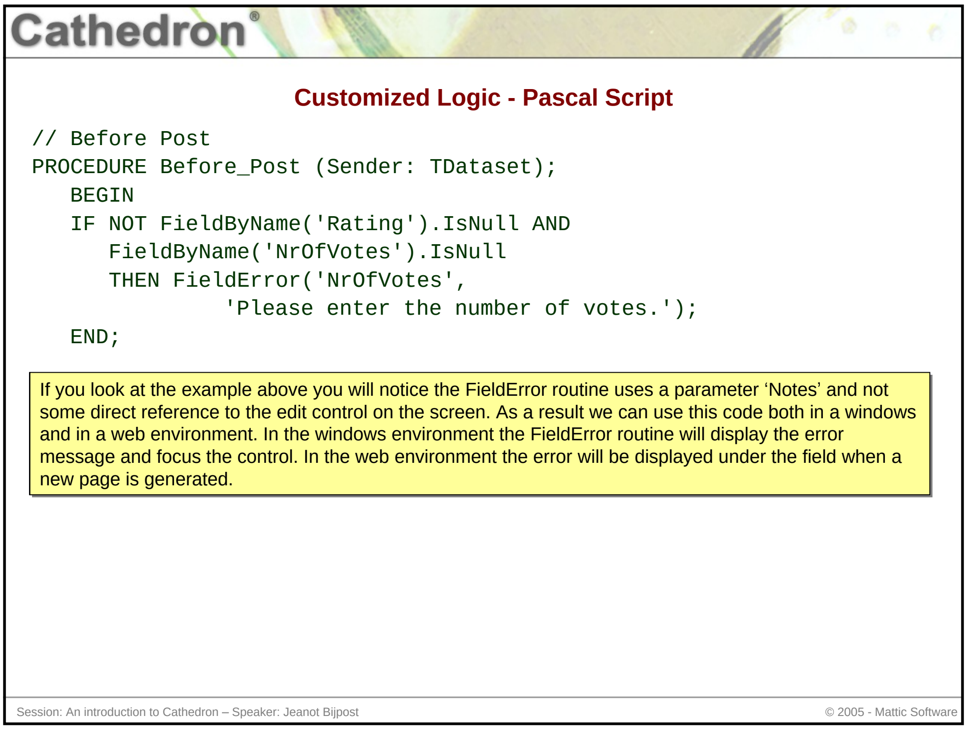
Distributor: 191

Automaticly added lookup:

```
SELECT Code, Description
FROM RentalType
```

This lookup was automatically created by Cathedron. The RentalType table was recognized as a 'Lookup' table. Since there is a limited amount of rental types, the values for lookup are cached on the client.

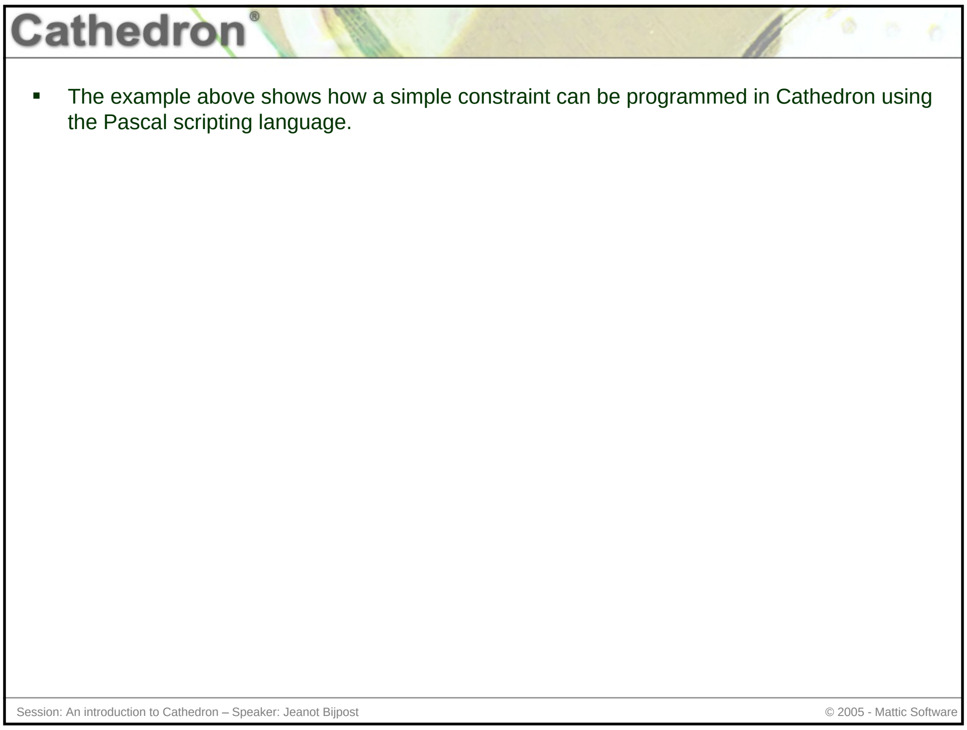
- Example: Change the way the distributor is displayed.
- The distributor column of the movie table references the contact table. So a distributor is in fact a contact.
- We can use Cathedron to define lookups. There are several different types of lookups:
  - Not cached
    - This lookup uses a query to map a value onto a display value. Example: SELECT Code, Description FROM RentalType. The result of this query is not cached.
  - Cached on create
    - Uses exactly the same query as the 'Not cached' type. Only this time the result of the query is cached as soon as the lookup is created and updated and redistributed when the rental type table changes. When a client connects it retrieves a cache file from the server containing all the lookups. This file is stored on the client and only retrieved from the database after an update.
  - Cached on demand
    - Uses exactly the same query as the 'Cache on demand' type. This time the result of the query is cached as soon as the user accesses the lookup.
  - Live
    - This lookup uses an other query. Example: SELECT Name FROM Contact WHERE ID=:KeyValue. This query could be the solution to change the way our distributor is displayed. Cathedron will execute this query each time it has to display a contact. Since the contact table contains over 10.000 records we do not prefer caching neither do we prefer a combo box. Cathedron replaces the combo box with a default search button.
  - Fixed value
    - This lookup uses a fixed table an can be used for domains such as: CREATE DOMAIN YesNo AS Char(1) CHECK (Value IN ('Y', 'N') OR Value IS NULL); If there is no table defining a description for the codes Y and N we can use a fixed lookup to translate those into Yes and No.
- Another way to solve our display problem is to use a view instead of a table. In this view we will join the contact name and display the name instead of the contact id.



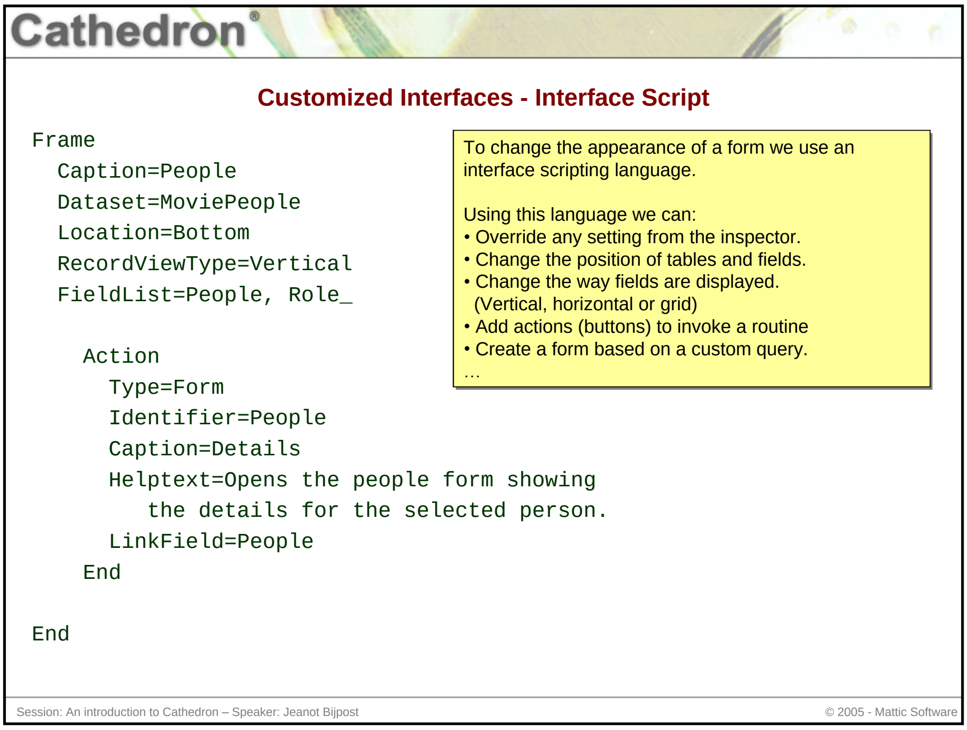
Customized Logic - Pascal Script

```
// Before Post
PROCEDURE Before_Post (Sender: TDataset);
   BEGIN
   IF NOT FieldByName('Rating').IsNull AND
      FieldByName('NrOfVotes').IsNull
      THEN FieldError('NrOfVotes',
               'Please enter the number of votes.');
   END;
```

If you look at the example above you will notice the FieldError routine uses a parameter ‘Notes’ and not some direct reference to the edit control on the screen. As a result we can use this code both in a windows and in a web environment. In the windows environment the FieldError routine will display the error message and focus the control. In the web environment the error will be displayed under the field when a new page is generated.



- The example above shows how a simple constraint can be programmed in Cathedron using the Pascal scripting language.



Cathedron®

Customized Interfaces - Interface Script

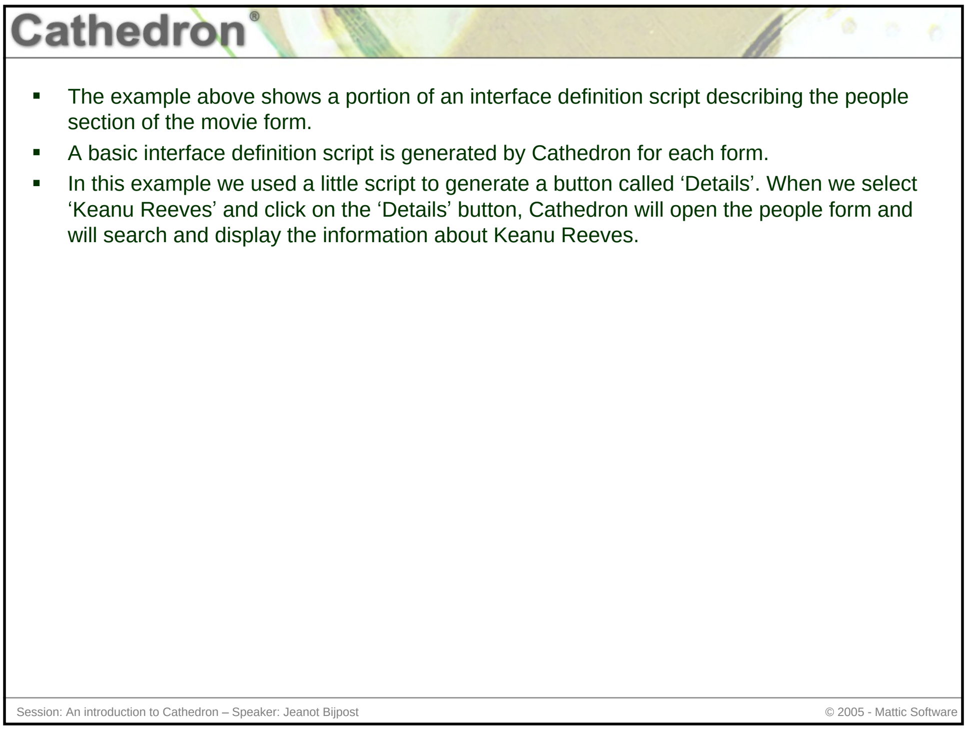
```
Frame
  Caption=People
  Dataset=MoviePeople
  Location=Bottom
  RecordViewType=Vertical
  FieldList=People, Role_

    Action
      Type=Form
      Identifier=People
      Caption=Details
      Helptext=Opens the people form showing
         the details for the selected person.
      LinkField=People
    End

End
```

To change the appearance of a form we use an interface scripting language.

Using this language we can:
- Override any setting from the inspector.
- Change the position of tables and fields.
- Change the way fields are displayed. (Vertical, horizontal or grid)
- Add actions (buttons) to invoke a routine
- Create a form based on a custom query.

…

Session: An introduction to Cathedron – Speaker: Jeanot Bijpost © 2005 - Mattic Software

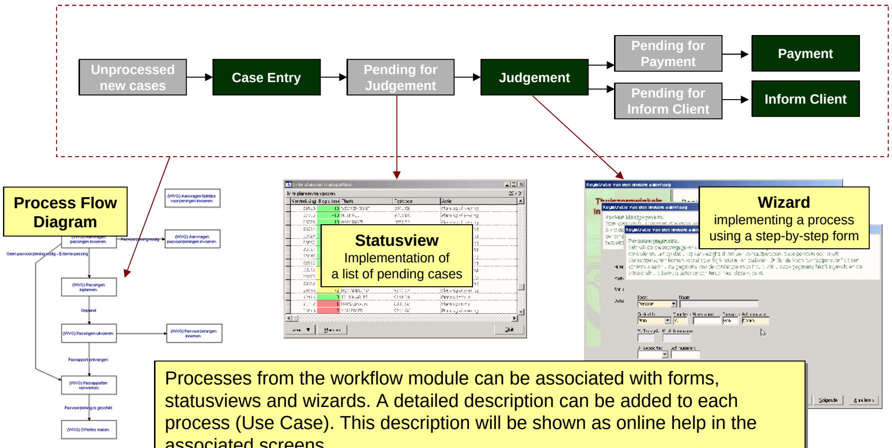


Cathedron®

- The example above shows a portion of an interface definition script describing the people section of the movie form.
- A basic interface definition script is generated by Cathedron for each form.
- In this example we used a little script to generate a button called ‘Details’. When we select ‘Keanu Reeves’ and click on the ‘Details’ button, Cathedron will open the people form and will search and display the information about Keanu Reeves.

Session: An introduction to Cathedron – Speaker: Jeanot Bijpost © 2005 - Mattic Software

Workflow



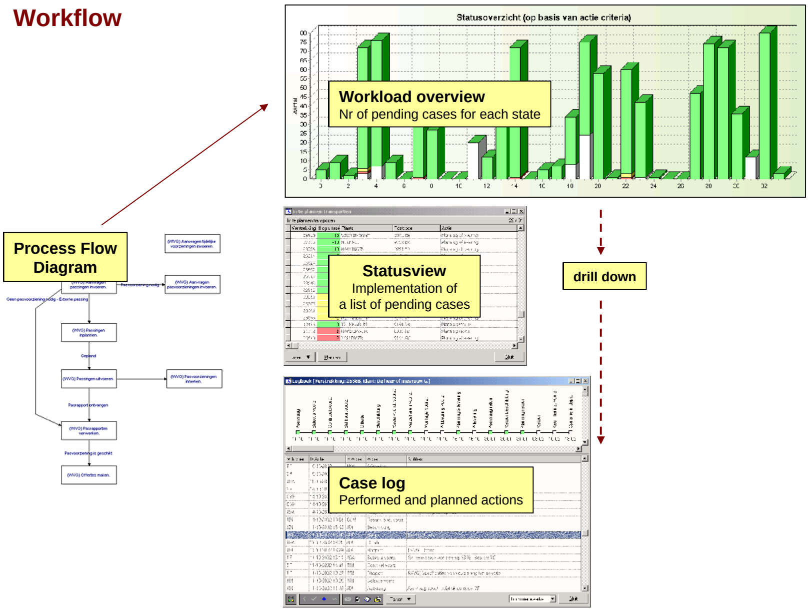
Processes from the workflow module can be associated with forms, statusviews and wizards. A detailed description can be added to each process (Use Case). This description will be shown as online help in the associated screens.

Groups of users (roles) can be authorized to execute a process. This information is used by Cathedron to create a customized menu for each user.

Workflow in Cathedron:

- Adding workflow to your Cathedron application is optional.
- You can use Cathedron to:
  - Create a simple workflow diagram.
  - Add descriptions to each process in this diagram.
    - Can be used as Use Cases during the design, can be updated to become online instructions for the final application.
  - Create ‘statusviews’ displaying the pending records at a certain point in the workflow. Using deadlines and colors to indicate the status of a case with respect to its deadline.
  - Create ‘wizards’ to implement a process and guide a user step-by-step to complete a process.
- Note: Statusviews and wizards are not default screens. They are not generated automatically. You can however benefit from generation because both statusviews and wizards use the interface definition language and the Pascal scripting language.

Workflow



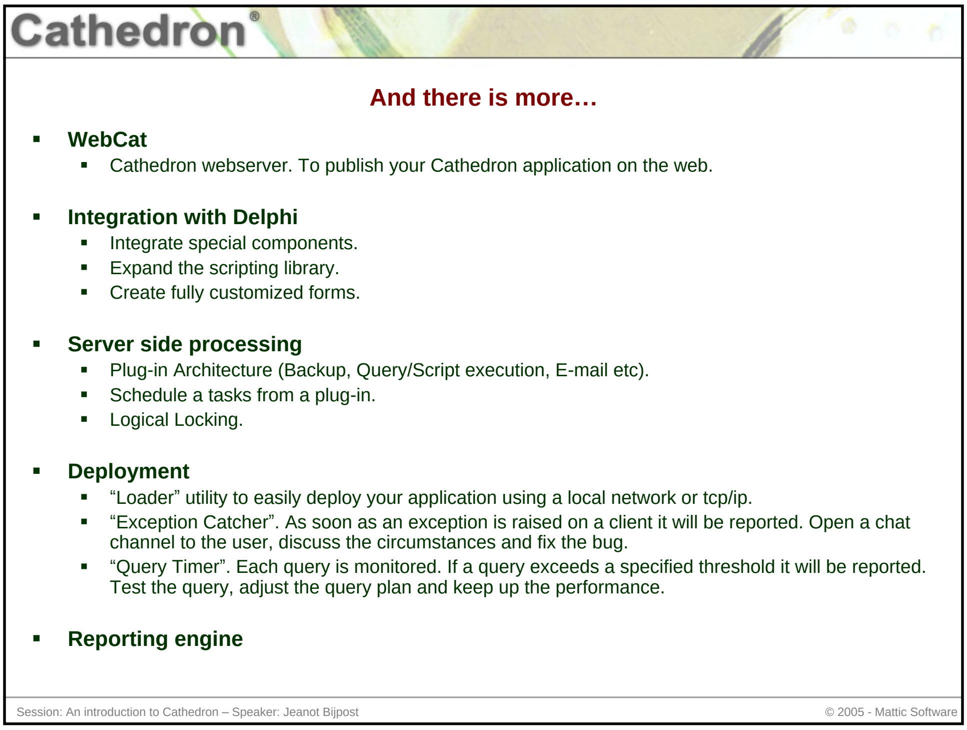
Workload overview
Nr of pending cases for each state

Process Flow Diagram

Statusview
Implementation of
a list of pending cases

Case log
Performed and planned actions

- We have used workflow to drilldown from aggregated information to a specific case:
  - At the top we find the workload chart (indicating the workload for every queue in the workflow)
  - From the workflow chart we can drilldown into a statusview as each statusview corresponds to exactly one queue and each queue is a bar in the chart.
  - From the statusview we can drilldown into a specific case.



And there is more…

- **WebCat**
  - Cathedron webserver. To publish your Cathedron application on the web.
- **Integration with Delphi**
  - Integrate special components.
  - Expand the scripting library.
  - Create fully customized forms.
- **Server side processing**
  - Plug-in Architecture (Backup, Query/Script execution, E-mail etc).
  - Schedule a tasks from a plug-in.
  - Logical Locking.
- **Deployment**
  - “Loader” utility to easily deploy your application using a local network or tcp/ip.
  - “Exception Catcher”. As soon as an exception is raised on a client it will be reported. Open a chat channel to the user, discuss the circumstances and fix the bug.
  - “Query Timer”. Each query is monitored. If a query exceeds a specified threshold it will be reported. Test the query, adjust the query plan and keep up the performance.
- **Reporting engine**

Cathedron®

Questions? – Questions!

- **Feedback**
- **Testers**

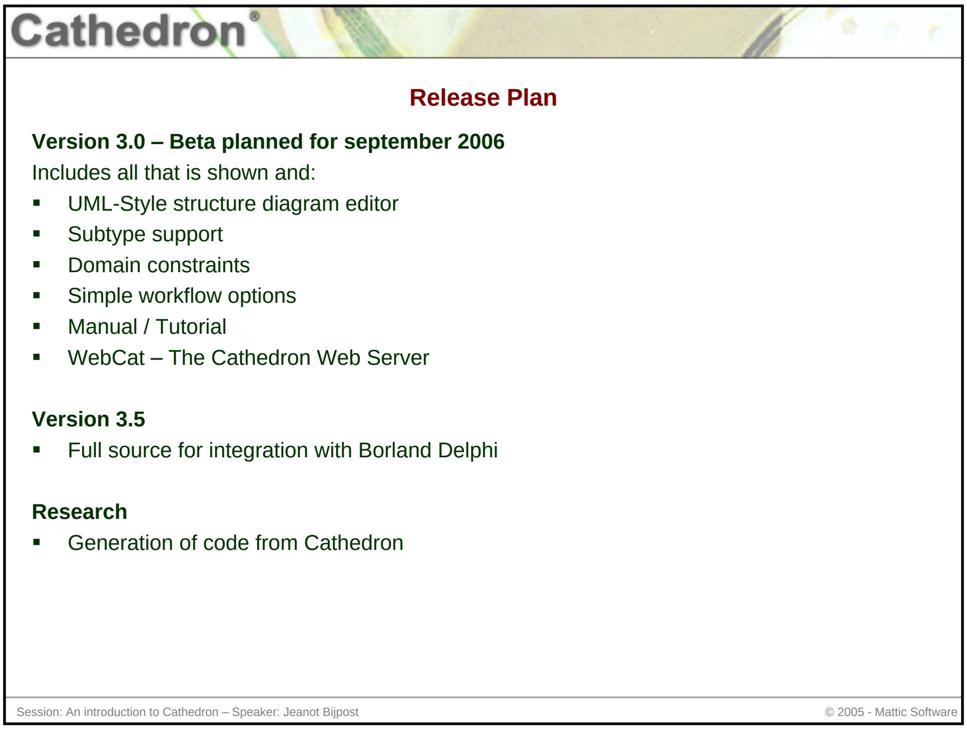
Session: An introduction to Cathedron – Speaker: Jeanot Bijpost

© 2005 - Mattic Software

Cathedron®

Session: An introduction to Cathedron – Speaker: Jeanot Bijpost

© 2005 - Mattic Software



Release Plan

Version 3.0 – Beta planned for september 2006

Includes all that is shown and:

- UML-Style structure diagram editor
- Subtype support
- Domain constraints
- Simple workflow options
- Manual / Tutorial
- WebCat – The Cathedron Web Server

Version 3.5

- Full source for integration with Borland Delphi

Research

- Generation of code from Cathedron