

New Beauty in Camelot

EuroPython 2011

Erik Janssens



@erikjanss



<http://www.conceptive.be>

Jeroen Dierckx



@jeroendierckx



<http://shore.be>

-  <http://www.python-camelot.com>
-  <http://groups.google.com/group/project-camelot>



What is Camelot?

Camelot

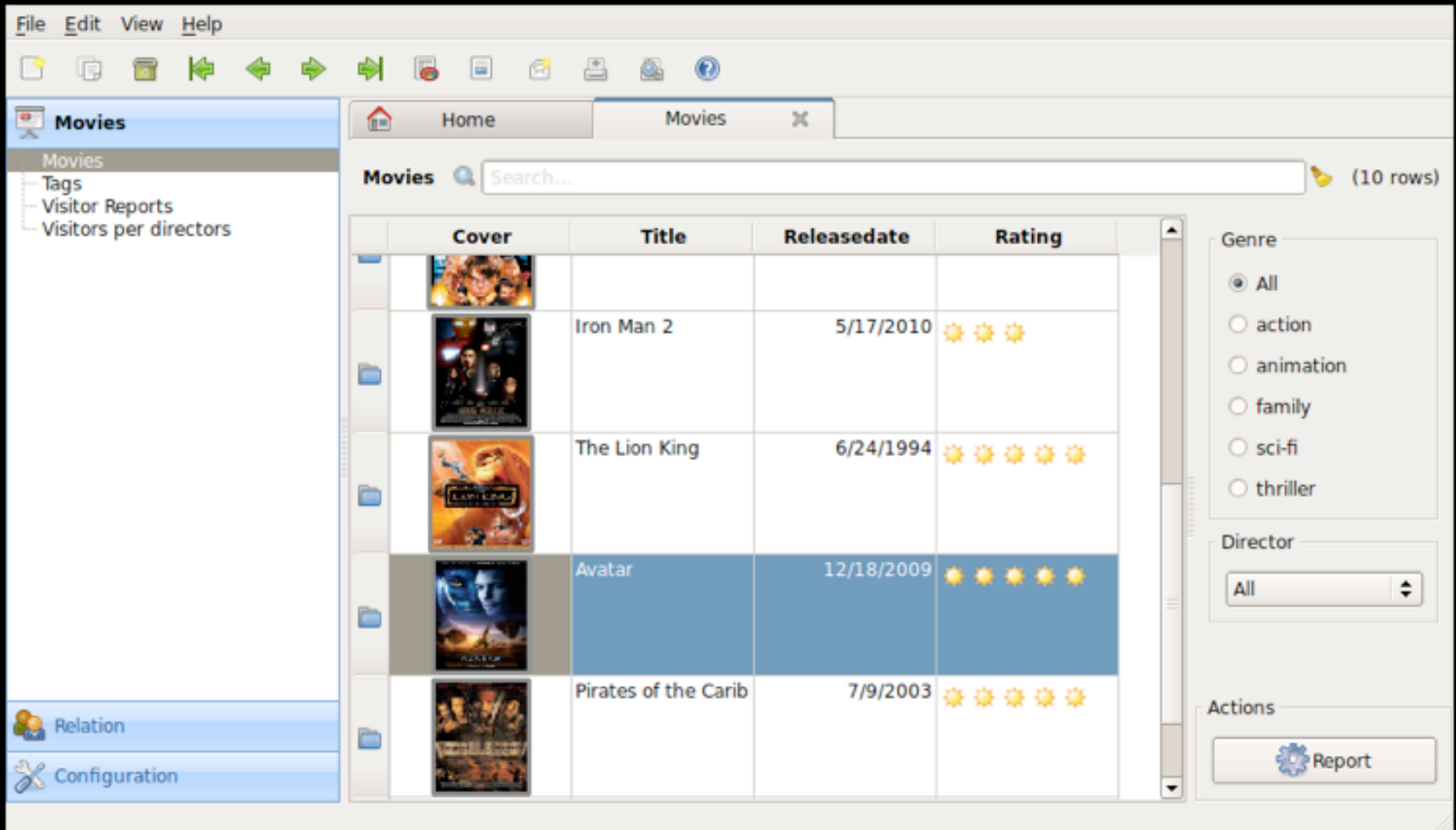
Desktop Application Framework

```
1 class Movie(Entity):
2
3     title = Field(Unicode(60), required=True)
4     short_description = Field(Unicode(512))
5     release_date = Field(Date)
6     genre = Field(Unicode(15))
7
8     class Admin(EntityAdmin):
9
10         list_display = ['title', 'short_description', 'release_date', 'genre']
```

What is Camelot?

Camelot

Desktop Application Framework



Why use Camelot

Camelot

Desktop Application Framework

- * User and Developer friendly
- * Based on Qt and SQLAlchemy
- * Productive environment
- * Multithreaded from the start

What others are saying about Camelot

Camelot

Desktop Application Framework

“Camelot is proving to be a great software development environment.”

- *Jay Foolie*

“From what I've seen, Camelot looks really promising!”

- *Hans-Peter Jansen*

“Camelot is excellent work. It's just fun to use.”

- *Jens*

“I'm already able to build desktop applications at warp speed
- just as promised.”

- *Andreas Kaiser*

“I'm really enjoying it.”

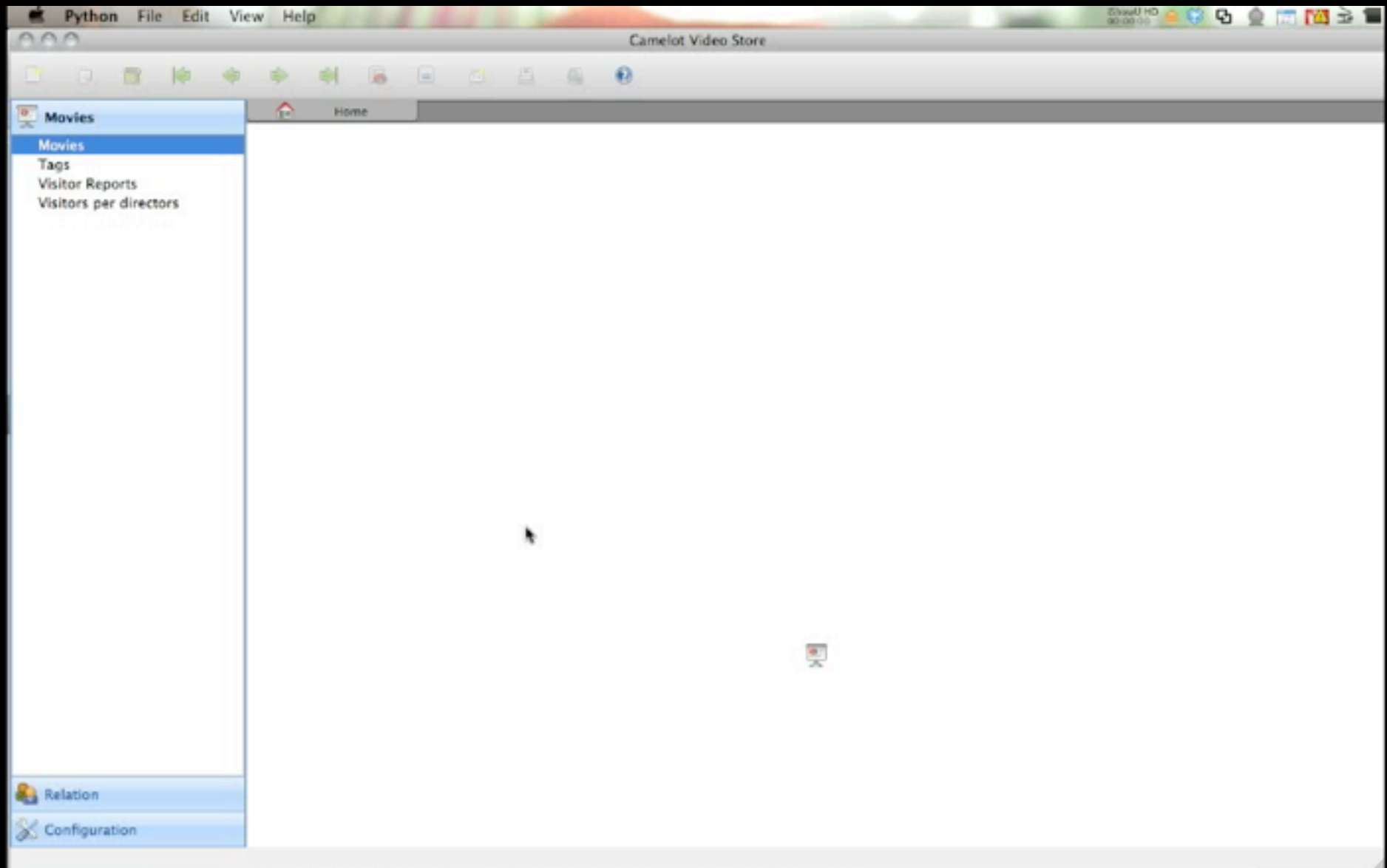
- *Iacopo Spalletti*

- * New Features
 - * Tab based Desktop
 - * Displaying Queries
 - * Table View
 - * Dynamic Field Attributes
 - * Actions with Options
 - * Matplotlib integration
- * Lessons Learned
 - * Deployment
 - * Development model
 - * Tools

- * Intuitive Navigation Pane
- * Application Actions on Home tab
- * Double click on title bar to maximize

Tab based Desktop

Camelot
Desktop Application Framework

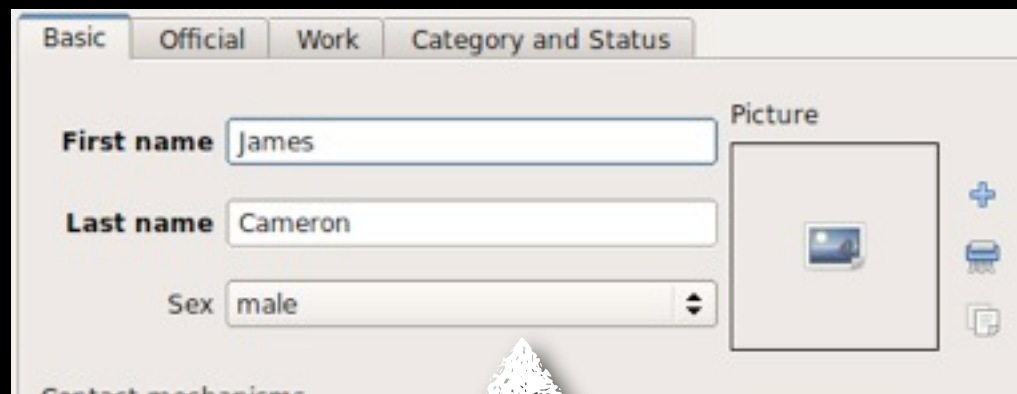


Displaying queries

Camelot

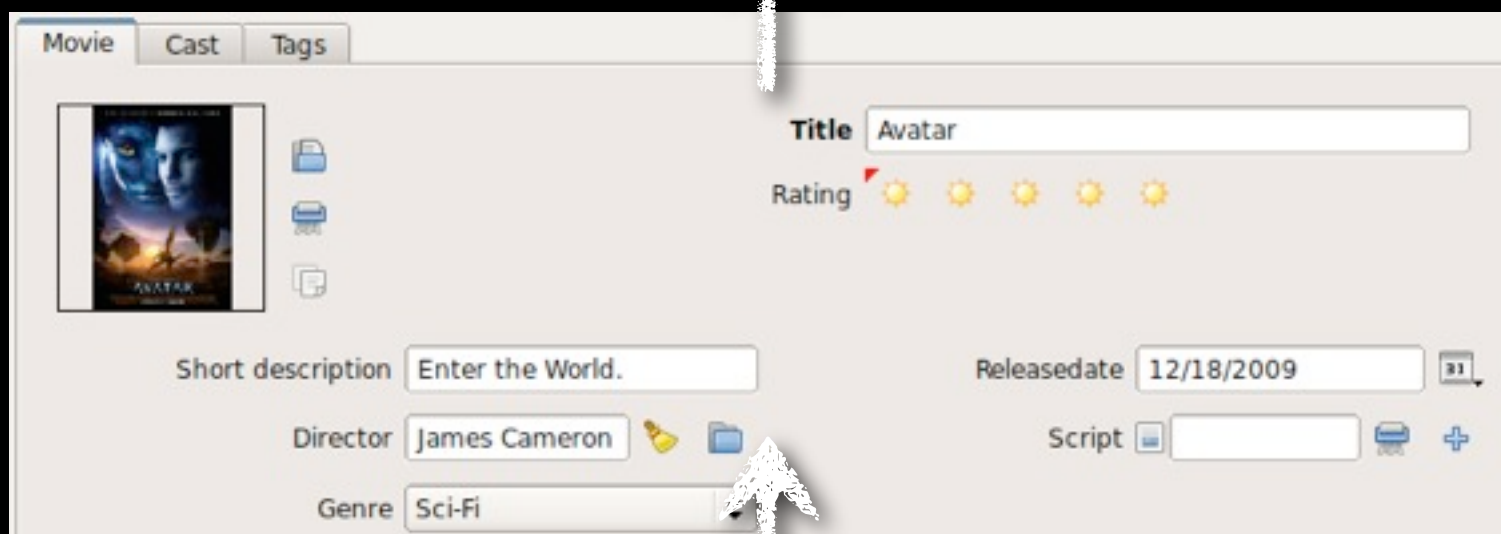
Desktop Application Framework

Person



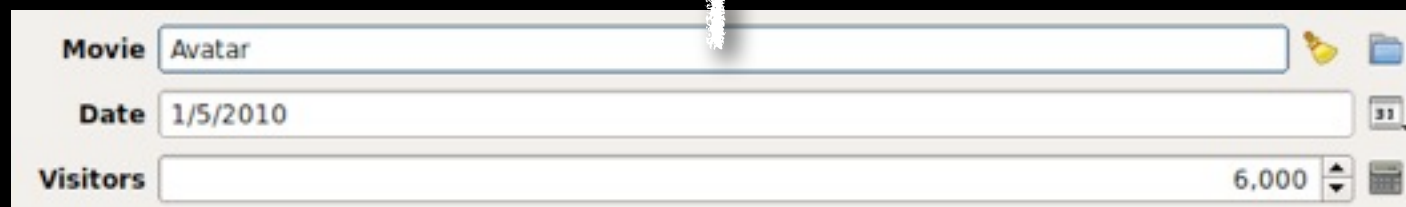
A form for editing a person's information. It has four tabs: 'Basic' (selected), 'Official', 'Work', and 'Category and Status'. The 'Basic' tab contains three text input fields: 'First name' with the value 'James', 'Last name' with the value 'Cameron', and 'Sex' with a dropdown menu showing 'male'. To the right of these fields is a 'Picture' section with a placeholder image and three icons: a plus sign, a printer, and a document. Below the 'Sex' field, the text 'Contact mechanisms' is partially visible.

Movie



A form for editing a movie's information. It has three tabs: 'Movie' (selected), 'Cast', and 'Tags'. The 'Movie' tab contains a movie poster on the left. To the right of the poster are three icons: a folder, a printer, and a document. Further right are the 'Title' field with the value 'Avatar', a 'Rating' section with five yellow stars, a 'Short description' field with the value 'Enter the World.', a 'Releasedate' field with the value '12/18/2009' and a calendar icon, a 'Director' field with the value 'James Cameron' and a folder icon, a 'Script' field with a folder icon and a plus sign, and a 'Genre' field with the value 'Sci-Fi' and a dropdown arrow.

VisitorReport



A form for displaying visitor report information. It contains three rows: 'Movie' with the value 'Avatar' and a folder icon, 'Date' with the value '1/5/2010' and a calendar icon, and 'Visitors' with a text input field containing the value '6,000' and a spinner control.

Displaying queries

Camelot

Desktop Application Framework

Person


Basic Official Work Category and Status

First 1 `class Person(Entity):`
2 `first_name = Field(Unicode(40), required = True)`
Last 3 `last_name = Field(Unicode(40), required = True)`

Sex male

Movie

Movie Cast Tags

 Title Avatar

1 `class Movie(Entity):`
2 `title = Field(Unicode(60), required=True)`
3 `director = ManyToOne('Person')`

Short description Enter the World. Releasedate 12/18/2009

Director James Cameron Script

Genre Sci-Fi

VisitorReport

Movie Av

Date 1/5

Visitors

1 `class VisitorReport(Entity):`
2 `movie = ManyToOne('Movie', required=True)`
3 `date = Field(Date, required=True)`
4 `visitors = Field(Integer, required=True)`

Displaying queries

Camelot

Desktop Application Framework

```
1 class VisitorsPerDirector(object):  
2  
3     class Admin(EntityAdmin):  
4         list_display = ['first_name', 'last_name', 'visitors']  
5
```

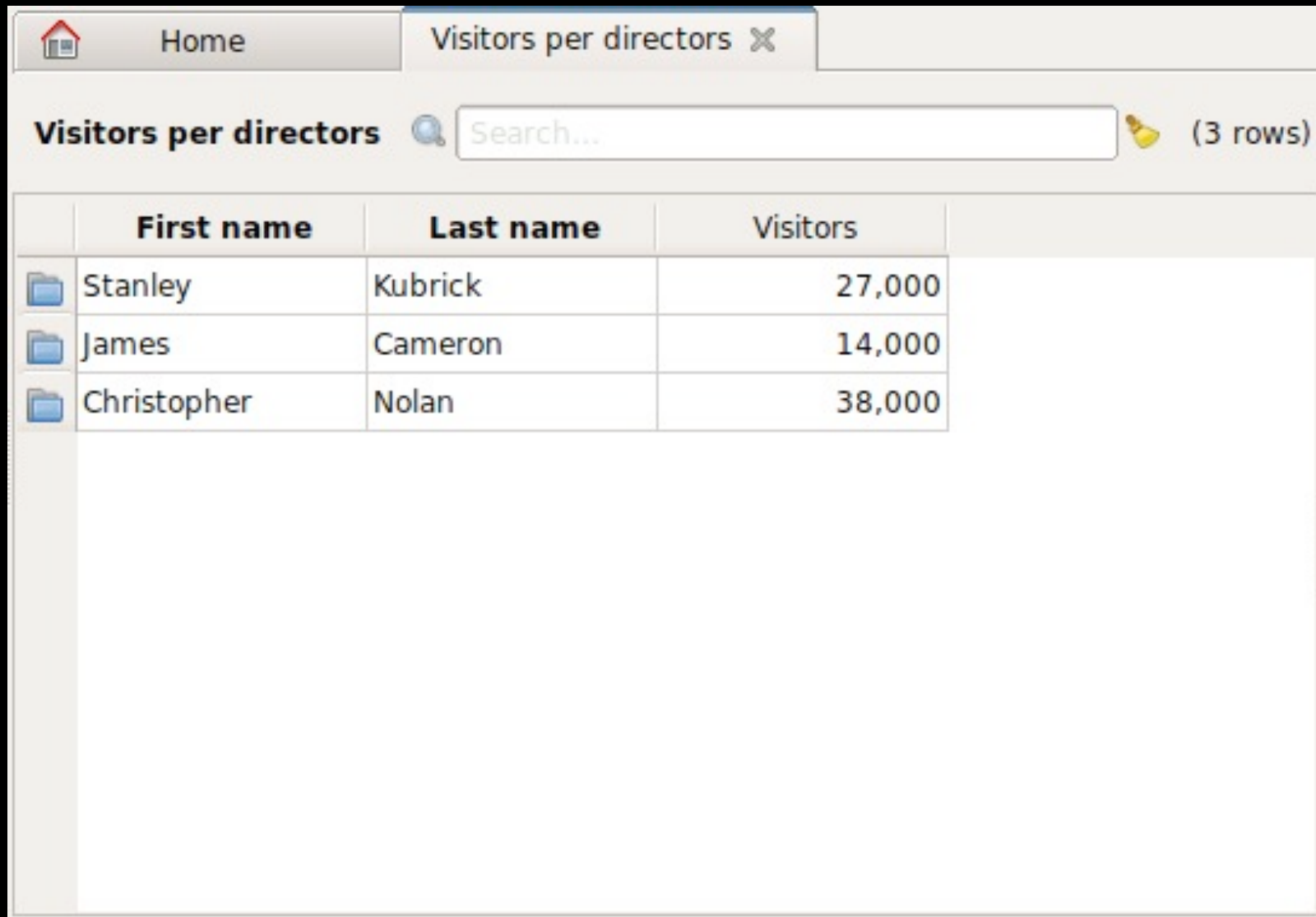
Displaying queries

```
1 def setup_views():
2
3     s = select([
4         Person.party_id,
5         Person.first_name.label('first_name'),
6         Person.last_name.label('last_name'),
7         func.sum( VisitorReport.visitors ).label('visitors'),],
8
9         whereclause = and_( Person.party_id == Movie.director_party_id,
10                             Movie.id == VisitorReport.movie_id),
11
12         group_by = [ Person.party_id,
13                     Person.first_name,
14                     Person.last_name ] )
15
16     s=s.alias('visitors_per_director')
17
18     mapper( VisitorsPerDirector, s, always_refresh=True )
```

Displaying queries

Camelot

Desktop Application Framework



The screenshot shows a web application interface with a tabbed navigation bar at the top. The first tab is 'Home' with a house icon. The second tab is 'Visitors per directors' with a close button (X). Below the tabs, the title 'Visitors per directors' is followed by a search bar with a magnifying glass icon and the text 'Search...'. To the right of the search bar is a yellow bell icon and the text '(3 rows)'. Below this is a table with three columns: 'First name', 'Last name', and 'Visitors'. The table contains three rows of data, each with a folder icon in the first column. The data is as follows:




	First name	Last name	Visitors
	Stanley	Kubrick	27,000
	James	Cameron	14,000
	Christopher	Nolan	38,000

Table view

Camelot

Desktop Application Framework

frozen
columns

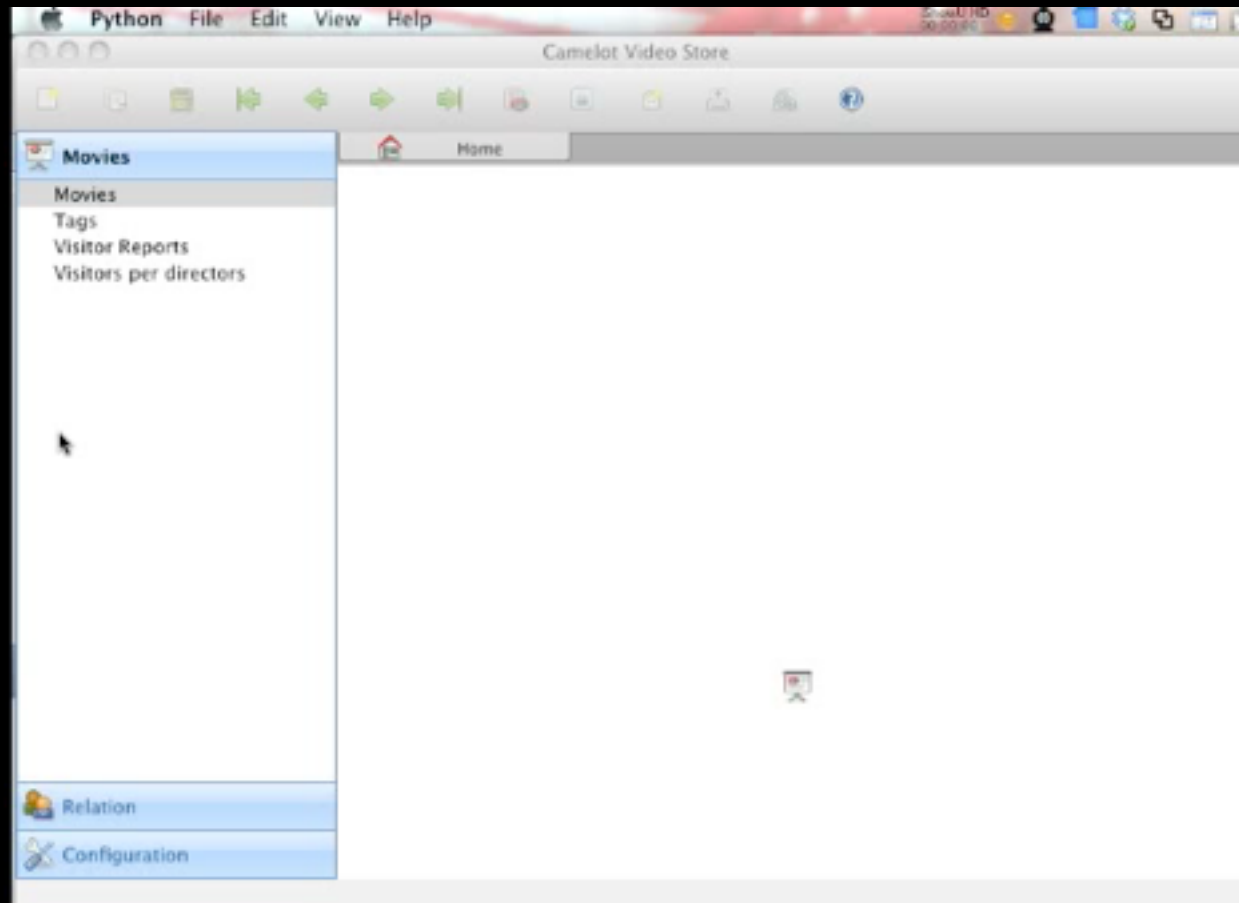


Table view

Camelot

Desktop Application Framework

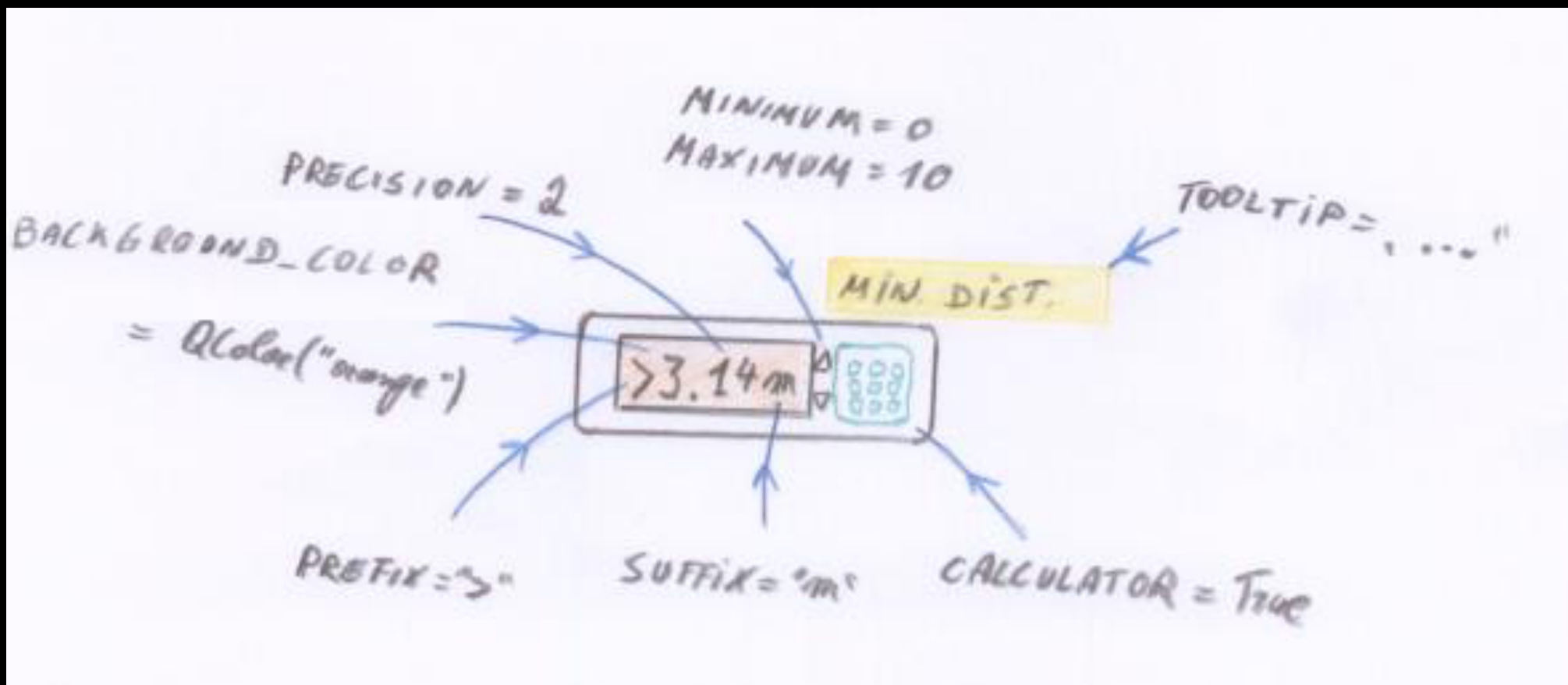
frozen
columns

```
1 class Movie (Entity) :  
2     ...  
3  
4     class Admin (EntityAdmin) :  
5         ...  
6         list_columns_frozen = 1
```

Dynamic Field Attributes

Camelot

Desktop Application Framework



Dynamic Field Attributes

Camelot

Desktop Application Framework

Product definition 1 : Python Investments

Definition Accounting Rules Features Settings Financial Insurances Funds Clauses Notifications Master Brokers

Features

Type	Value	From amount	Thru amount	From agreed duration	Thru agreed duration	From passed duration	Thru passed d
Interest rate	210,100000 %	14,00	11,00	13	12.402	11	
Financed commissions rate	6,000000 %	0,00		0	2.400	0	
Cooling off period	30 days	0,00		0	2.400	0	
Investment delay	7 days	0,00		0	2.400	0	
Financed commissions deduction rate	0,330000 %	0,00		0	2.400	0	
Financed commissions deduction rate	0,420000 %	0,00		0	2.400	0	
Financed commissions deduction rate	0,830000 %	0,00		0	2.400	0	
Financed commissions deduction rate	1,250000 %	0,00		0	2.400	0	
Financed commissions deduction rate	1,670000 %	0,00		0	2.400	0	
Financed commissions deduction rate	2,080000 %	0,00		0	2.400	0	
Financed commissions deduction rate	2,500000 %	0,00		0	2.400	0	
Financed commissions deduction rate	0,080000 %	0,00		0	2.400	0	
Financed commissions deduction rate	0,250000 %	0,00		0	2.400	0	
Financed commissions deduction rate	0,170000 %	0,00		0	2.400	0	
Financed commissions interest rate	0,550000 %	0,00		0	2.400	0	

Dynamic Field Attributes

Camelot

Desktop Application Framework

Applicable Feature 1 : interest_rate (9.100000)

Vaste jaarlijkse interest voet

Type	Interest rate	Value	9,100000 %
From amount	14,00	Thru amount	11,00
From agreed duration	13	Thru agreed duration	12.402
From passed duration	11	Thru passed duration	12.400
From attributed duration	12	Thru attributed duration	12.400
Premium from date	31-12-2009	Premium thru date	31-12-2400
Apply from date	31-12-2009	Apply thru date	31-12-2400
Automated clearing	True	Overrule required	<input type="checkbox"/>
Premium period type	Monthly		

Comment

17:12
19-6-2011

Dynamic Field Attributes

Camelot

Desktop Application Framework

```
1 features = [ (1, 'intrest_rate'),  
2              (2, 'tax_rate'),  
3              (3, 'entry_fee'), ]  
4  
5 class Feature(Entity):  
6     product = ManyToOne('Product')  
7     name = Field( Enumeration(features) )  
8     value = Field( Numeric() )  
9  
10    class Admin(EntityAdmin):  
11        list_display = ['name', 'value']
```

Dynamic Field Attributes

Camelot

Desktop Application Framework

```
1 features = [ (1, 'intrest_rate', '%'),
2               (2, 'tax_rate',      '%'),
3               (3, 'entry_fee',     'Euro'), ]
4
5 feature_enumeration = dict( (pf[0],pf[1]) for pf in features)
6 feature_suffixes    = dict( (pf[1],pf[2]) for pf in features)
7
8 def value_suffix(feature):
9     return feature_suffixes[feature.name]
10
11 class Feature(Entity):
12     product = ManyToOne('Product')
13     name = Field( Enumeration(feature_enumeration) )
14     value = Field( Numeric() )
15
16     class Admin(EntityAdmin):
17         list_display = ['name', 'value']
18         field_attributes = {'value' : {'suffix':value_suffix} }
```

Dynamic Field Attributes

Camelot

Desktop Application Framework

```
1 features = [ (1, 'intrest_rate', '%', 10),
2              (2, 'tax_rate', '%', 30),
3              (3, 'entry_fee', 'Euro', 100), ]
4
5 feature_thresholds = dict( (pf[1],pf[3]) for pf in features)
6
7 def value_bg_color(feature):
8     from camelot.view.art import ColorScheme
9     if feature.value > feature_thresholds[feature.name]:
10         return ColorScheme.orange_1
11
12 class Feature(Entity):
13
14     class Admin(EntityAdmin):
15         list_display = ['name', 'value']
16         field_attributes = {'value' : {'suffix':value_suffix,
17                                       'background_color':value_bg_color} }
```

Context

- * List
- * Form
- * Application

Thread

- * Model
- * GUI

Actions - DocxFormAction

Camelot

Desktop Application Framework

Financial Account 4 : 152000000004

Account Items Brokers Assets Invoicing Settings Documents and Notifications Agreements Tasks Status

Account number

Full account number 152000000004

Product Python Investments

Roles

Described by	Rank	Summability	Natuurlijke persoon	Rechtspersoon	From date	Thru date
Subscriber	1	0,00	Márcos Cêris		3-2-2010	31-12-2400
Subscriber	2	0,00	Flater Guust		3-2-2010	31-12-2400

Premium schedules

Premium amount	Period type	Direct debit	From date	Thru date	Premiums to customer	Premiums to account	Premiums attributed to
2.500,00	Single	<input type="checkbox"/>	3-2-2010	3-2-2210	1	1	
1.000,00	Yearly	<input type="checkbox"/>	3-2-2010	3-2-2020	0	0	

Actions

Activate Account

Summary

Account state

Account movements

Create Customer

Actions - DocxFormAction

Camelot

Desktop Application Framework

```
1 class FinancialAccountState (DocxFormAction) :
2
3     def get_environment(self, obj):
4         fileloader = FileSystemLoader('path/to/templates')
5         return Environment(loader=fileloader, auto_escape=True)
6
7     def get_template(self, account):
8         return 'state_template.xml'
9
10    def get_context(self, account):
11        return {
12            'today' : datetime.date.today(),
13            'account': account,
14            'subscribers': account.subscribers,
15            'correspondent': account.correspondents,
16        }
```


Actions - DocxFormAction

Camelot

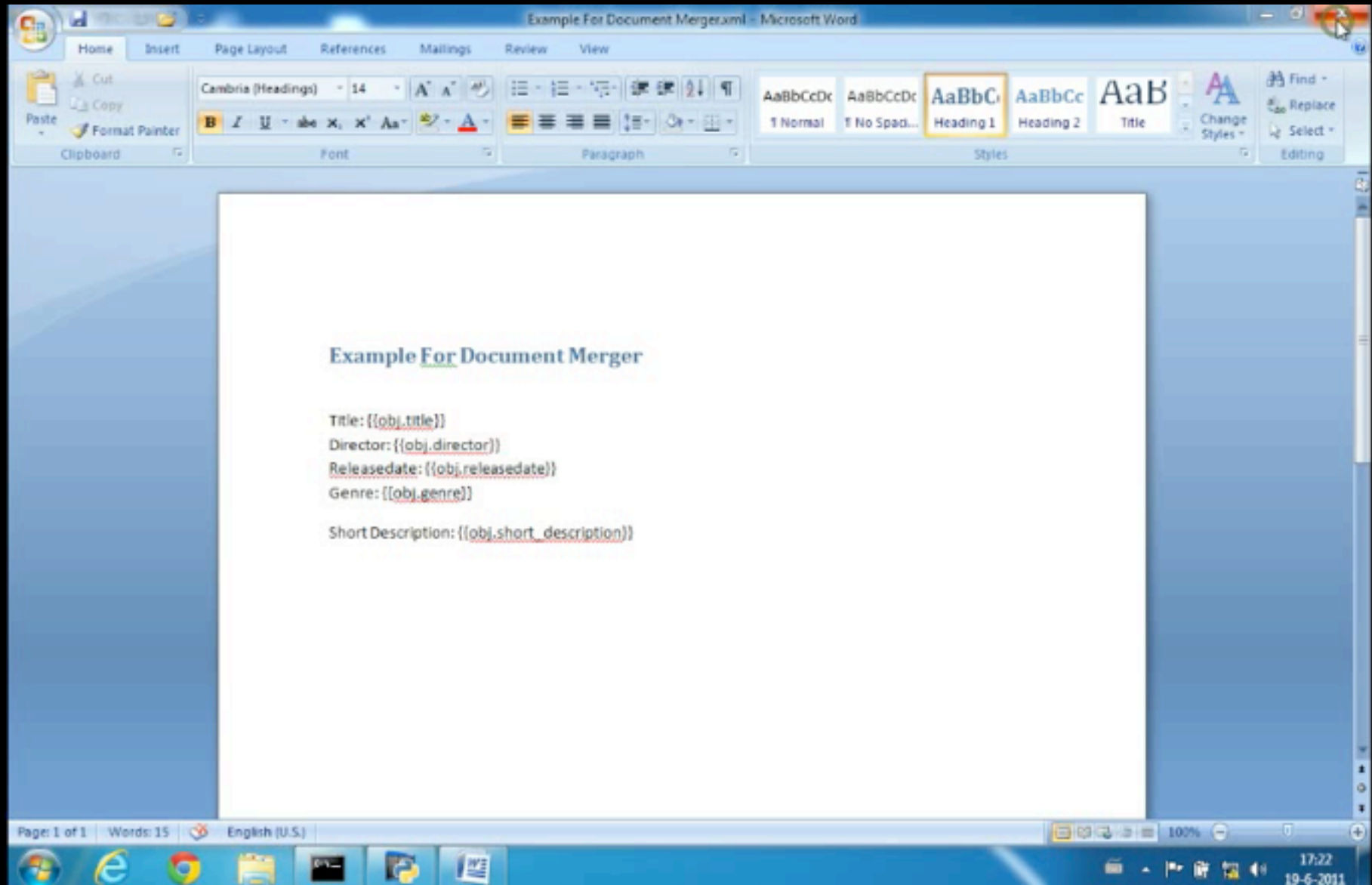
Desktop Application Framework

```
1 <w:r w:rsidR="00ED5055">
2   <w:rPr>
3     <w:rFonts w:ascii="News Gothic Std" w:hAnsi="News Gothic Std"/>
4     <w:lang w:val="en-US"/>
5   </w:rPr>
6   <w:t xml:space="preserve"> on :  {{today|date}} </w:t>
7 </w:r>
```

Actions - Document Merge

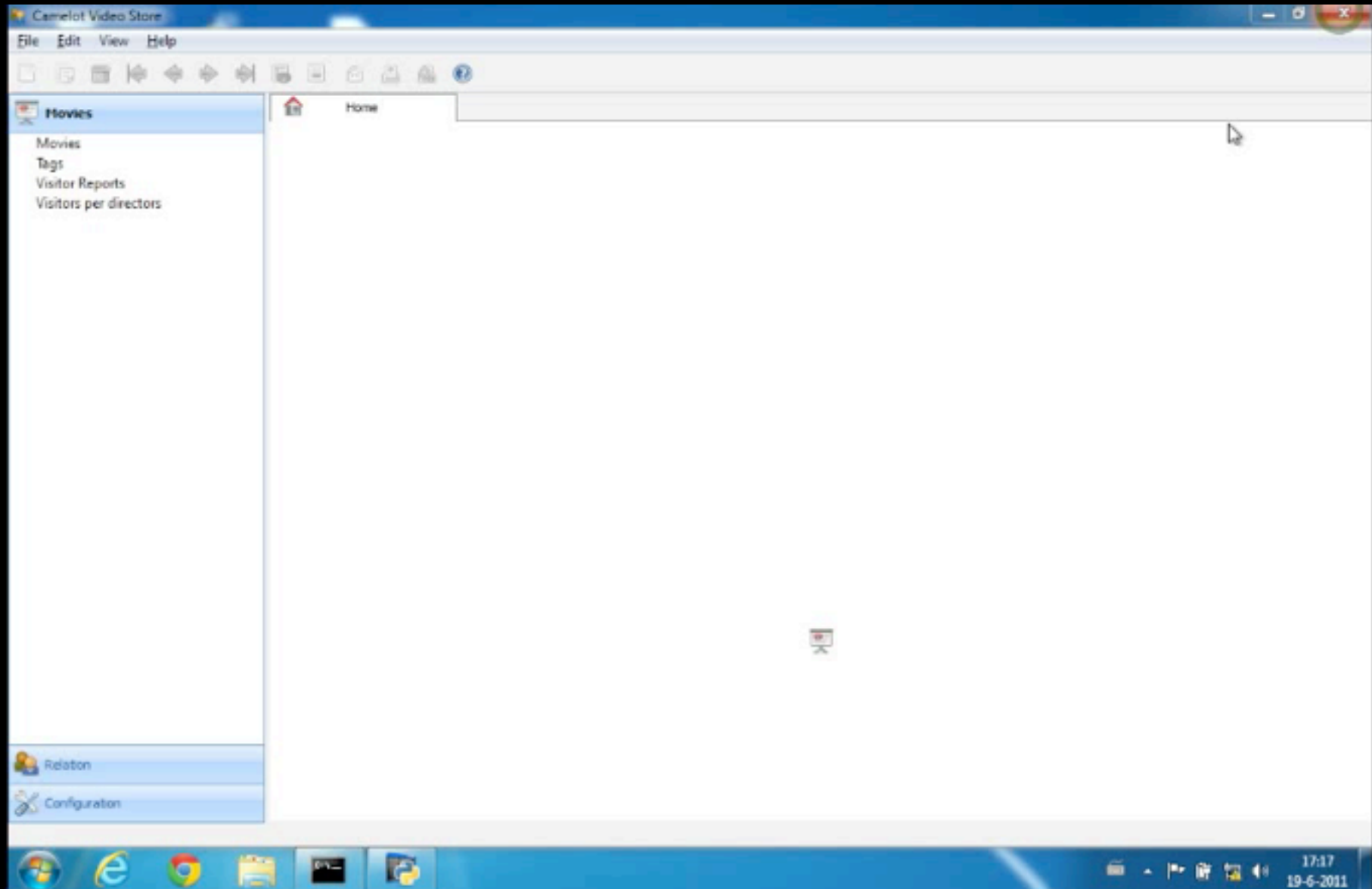
Camelot

Desktop Application Framework



Actions - List Action w/ Options

Camelot
Desktop Application Framework



Actions - List Action w/ Options

```
1 class ListActionFromModelFunction( ListAction ):
2
3     def model_run(self, collection, selection, options):
4         """runs in the model thread"""
5         pass
6
7     def run( self, collection_getter, selection_getter ):
8         """runs in the GUI thread"""
9         pass
```

Actions - List Action w/ Options

Camelot

Desktop Application Framework

```
1 class MyListAction( ListActionFromModelFunction ):
2
3     def model_run(self, collection, selection, options):
4         """runs in the model thread"""
5         for obj in selection:
6             print obj
7
8 class Movie(Entity):
9     ...
10    class Admin(EntityAdmin):
11        list_actions = [MyListAction(_('Report'))]
```

Actions - List Action w/ Options

Camelot

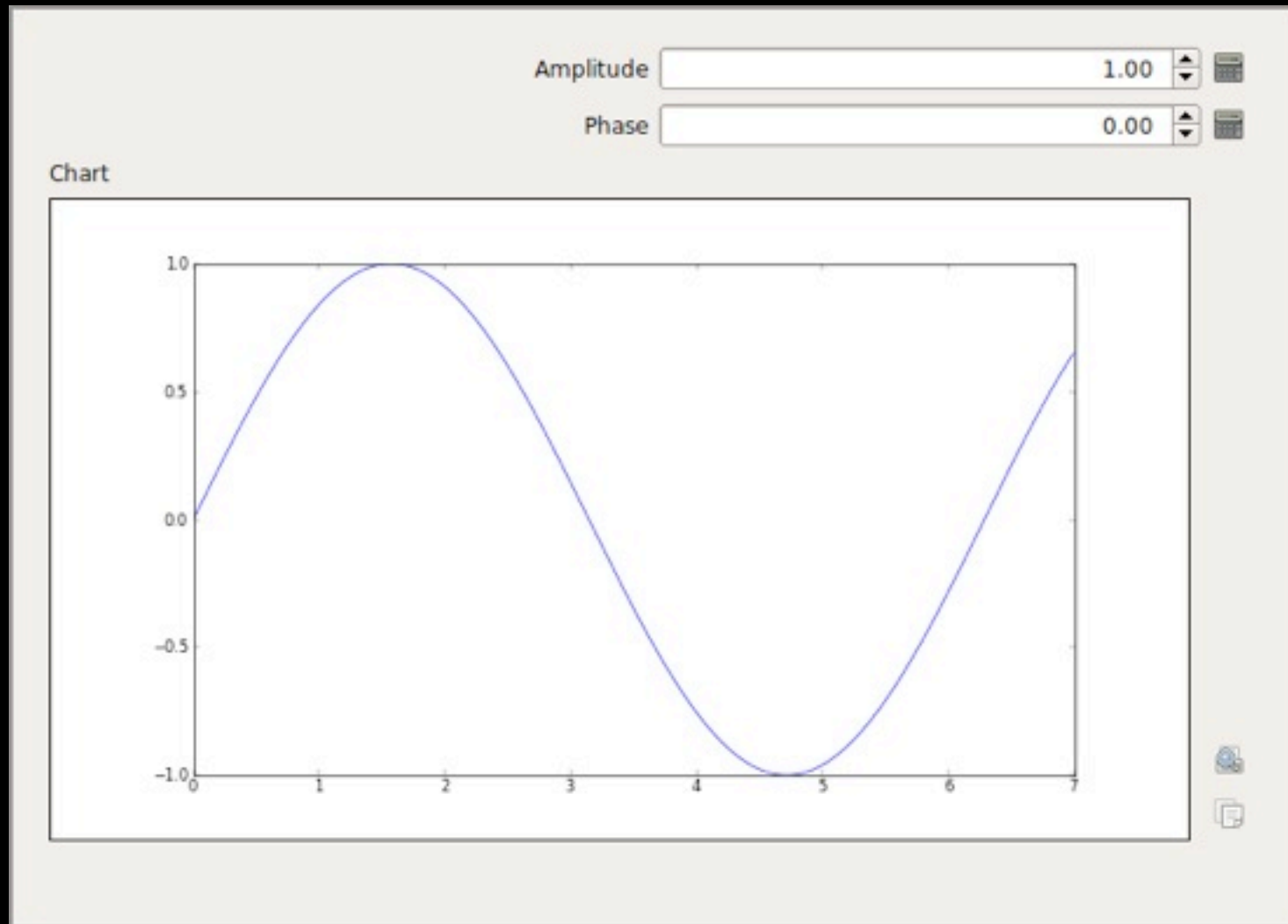
Desktop Application Framework

```
1 class MyListAction( ListActionFromModelFunction ):
2
3     class Options(object):
4
5         def __init__(self):
6             self.only_selected = False
7
8         class Admin( ObjectAdmin ):
9             form_display = ['only_selected']
10             field_attributes = {'only_selected':{'delegate':BoolDelegate,
11                                                'editable':True}}
12
13     def model_run(self, collection, selection, options):
14         """runs in the model thread"""
15         if options.only_selected:
16             for obj in selection:
17                 print obj
18         else:
19             for obj in collection:
20                 print collection
```

Matplotlib Integration

Camelot

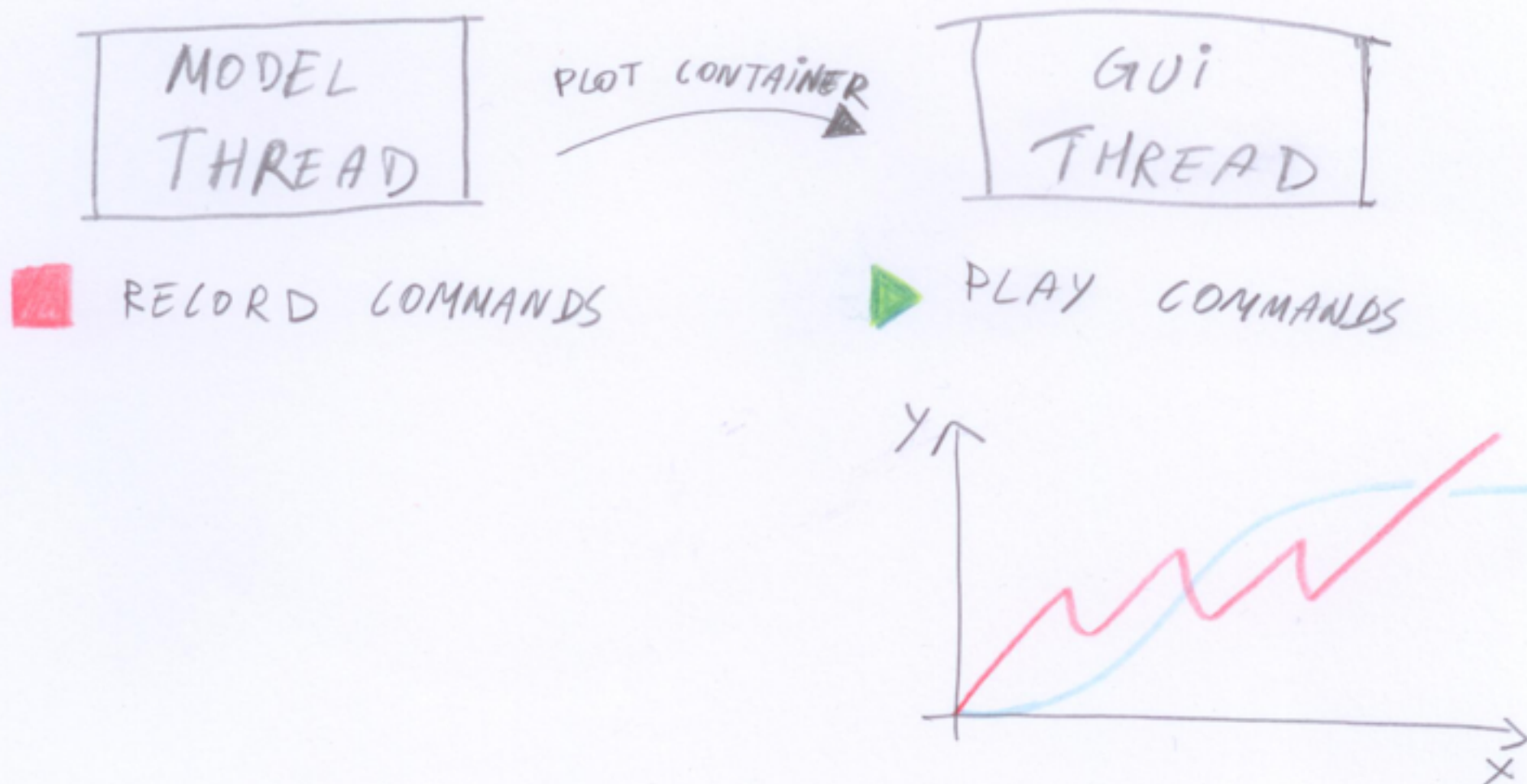
Desktop Application Framework



Matplotlib Integration

Camelot

Desktop Application Framework

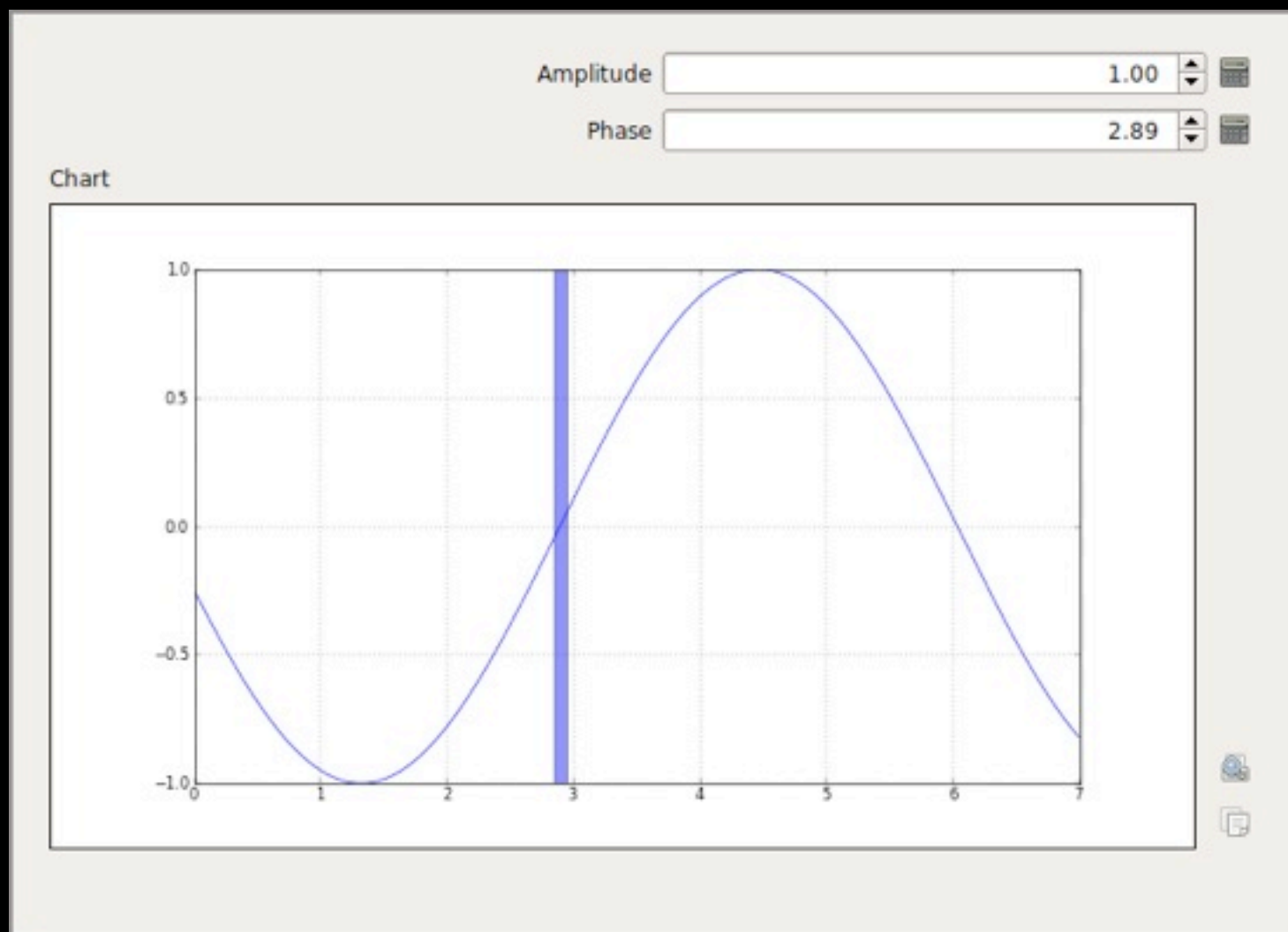



```
1 @property
2 def chart(self):
3     import math
4     x_data = [x/100.0 for x in range(1, 700, 1)]
5     y_data = [self.amplitude * math.sin(x - self.phase) for x in x_data]
6     return PlotContainer( x_data, y_data )
7
8 class Admin(ObjectAdmin):
9     field_attributes = dict(chart = dict(delegate=delegates.ChartDelegate))
```

Matplotlib Integration

Camelot

Desktop Application Framework



```
1 @property
2 def chart(self):
3     import math
4     axes = AxesContainer()
5     x_data = [x/100.0 for x in range(1, 700, 1)]
6     y_data = [self.amplitude * math.sin(x - self.phase) for x in x_data]
7     axes.plot( x_data, y_data )
8     axes.grid( True )
9     axes.axvspan(self.phase-0.05, self.phase+0.05, facecolor='b', alpha=0.5)
10    return axes
11
12 class Admin(ObjectAdmin):
13     field_attributes = dict(chart = dict(delegate=delegates.ChartDelegate))
```

That's us



*This is us,
after a 100 deployments*

*Notice how cool we're still,
though*

How??





Only
€ 45

Conceptive Python SDK

- * GUI toolkits such as Qt and PyQt
- * Database drivers
- * Crypto and SSL modules
- * Spyder IDE

For more information, check out
<http://www.python-camelot.com>

Development Model

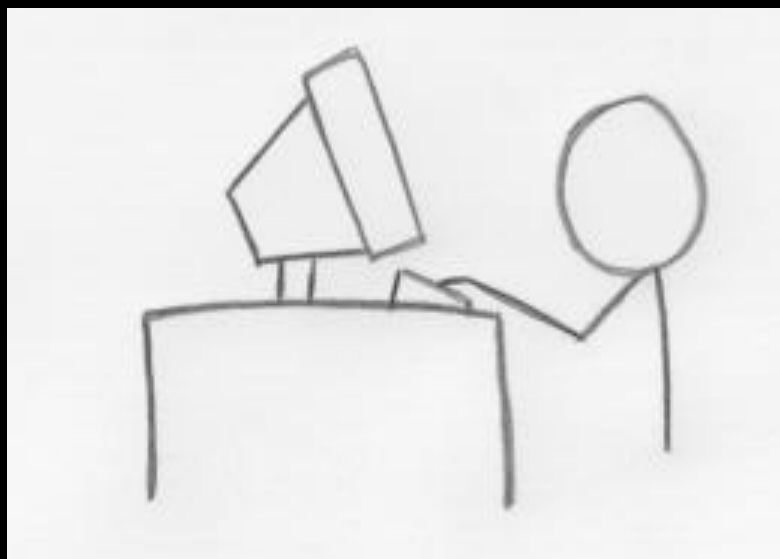
Camelot

Desktop Application Framework

DEVELOPER

DOMAIN
EXPERT

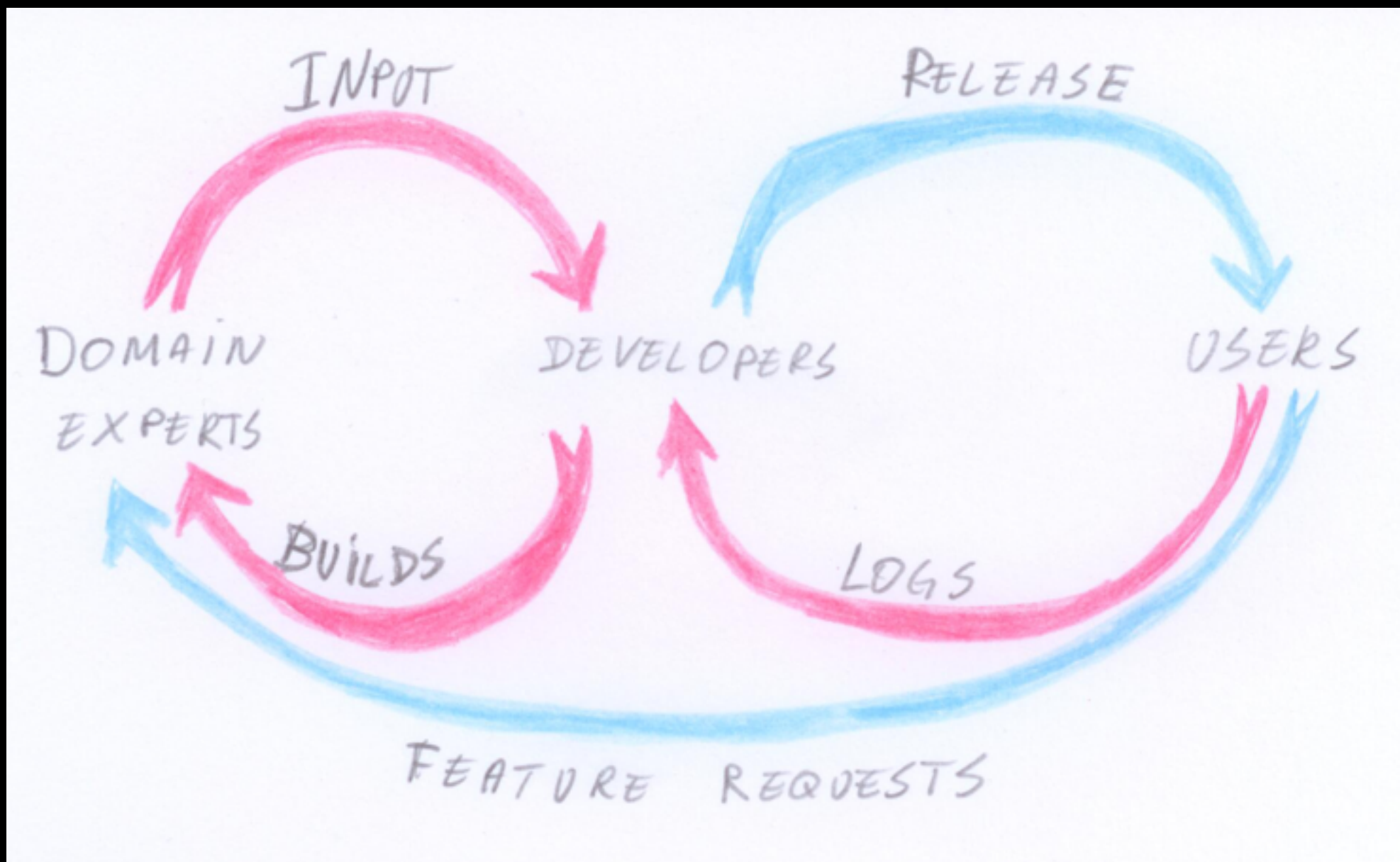
USER



Development Model

Camelot

Desktop Application Framework



Deployment tool - Cloudlaunch

Camelot

Desktop Application Framework

```
1 $ python setup.py bdist_cloud
2 creating 'dist/cloud/videostore-py2.7.egg' and adding 'build/bdist.linux-i686/egg' to it
3
4 $ python setup.py bdist_cloud wininst_cloud
5 creating 'dist/cloud/videostore-py2.7.egg' and adding 'build/bdist.linux-i686/egg' to it
6 creating 'dist/cloud/videostore-installer.exe' and adding 'build/bdist.linux-i686/egg' to it
```

Deployment tool - Cloudlaunch

Camelot

Desktop Application Framework

```
1 $ python setup.py bdist_cloud upload_cloud
2 creating 'dist/cloud/videostore-py2.7.egg' and adding 'build/bdist.linux-i686/egg' to it
3 upload dist/cloud/videostore.cld to key videostore.cld
4 key updated
5 upload dist/cloud/videostore-py2.7.egg to key videostore-py2.7.egg
6 key updated
7
8 $ python setup.py monitor_cloud
9 [2011-06-14 15:11:10,339 user1      9045 INFO      camelot] move backup file to its final location
10 [2011-06-14 15:16:40,042 user2      9045 ERROR      camelot] exception caught
11 Traceback (most recent call last):
12   File "build/bdist.linux-i686/egg/camelot/view/model_thread/signal_slot_model_thread.py", line 57, in execute
13   File "build/bdist.linux-i686/egg/camelot/view/model_thread/__init__.py", line 57, in wrapper
14   File "build/bdist.linux-i686/egg/camelot/view/proxy/collection_proxy.py", line 884, in _extend_cache
15   File "build/bdist.linux-i686/egg/camelot/view/proxy/collection_proxy.py", line 808, in _add_data
16 TypeError: PyQtSignal must be bound to a QObject, not 'CollectionProxy'
17 [2011-06-14 15:20:25,499 user3      9067 INFO      videostore] old database version detected
18 [2011-06-14 15:20:27,904 user3      9067 INFO      videostore] executing upgrade
```

*Thanks!
Questions?*

New Beauty in Camelot EuroPython 2011

Erik Janssens



@erikjanss



<http://www.conceptive.be>

Jeroen Dierckx



@jeroendierckx



<http://shore.be>

- **SQLAlchemy and Camelot Course**

- <http://www.python-camelot.com>

- <http://groups.google.com/group/project-camelot>



Slides will be online soon ...

