

Generic Component Framework

A Qt based Component Framework



-

Prashanth N Udupa

Email: [prashanth @ vcreatelogic . com](mailto:prashanth@vcreatelogic.com)

Key Rules in Software Design

- Divide and Rule
- Establish a Constitution
- Similarity in Diversity
- Bring them together

Survey of Component Frameworks

- KDE's KParts
- Microsoft's COM
- Mozilla's XPCOM
- SOAP
- CORBA
- and today let me introduce another one :)

Introducing GCF....



What is GCF?

- A C++ framework for Qt applications. Based on Qt 4.4
- Helps compose modules of an application as a dynamically loadable entities
- Provides GUI merging capabilities
- Provides object/functionality discovery services
- Provides support for inter-process-communication.
- Provides in-built components for common tasks
- Foundation for several commercial software and the open-source VTK Designer 2

Winner of Qt Centre Programming Contest 2007



Gallery.....

The screenshot displays the VTK Designer interface. The main canvas shows a pipeline diagram with the following components and connections:

- vtkPNGReader14** (vtkPNG Reader) is connected to **vtkImageToTerrain16** (vtkImageToTerrain) via a green line.
- vtkImageToTerrain16** is connected to **vtkPolyDataNormals17** (vtkPolyDataNormals) via a red line.
- vtkPolyDataNormals17** is connected to **vtkDataSetMapper18** (vtkDataSetMapper) via a red line.
- vtkDataSetMapper18** is connected to **vtkActor19** (vtkActor) via a blue line.
- vtkTexture15** (vtkTexture) is connected to **vtkActor19** via a blue line.

The **VTK Output** window on the right shows a 3D visualization of a terrain, rendered with a color gradient from blue (low) to red (high).

The **Message Area** at the bottom left contains the following log:

```

Loading component "CScriptEditorComponent"
Loading component "CVtkVisSystemComponent"
Loading component "CVtkFullscreenComponent"
Loading component "CDataFileLoaderComponent"
Successfully loaded file "C:/Projects/VTKDesigner2/bin/Samples/gray_scale_terrain_image.png"
Successfully loaded file "C:/Projects/Documentation/VTKData/DICOM/BREBIX/BREBIX/CT 10 ponction foie/DEF FOIE ART. - 107198/IMG-0001-0001.dcm"
Successfully loaded file "C:/Projects/VTKDesigner2/bin/Samples/gray_scale_terrain_image.png"
  
```

The **Pipeline Progress** window at the bottom right shows the following table:

VTK Class	Progress
vtkPNGReader14	Finished in 1146 ms...
vtkTexture15	Not started
vtkImageToTerrain16	Finished in 96 msec
vtkPolyDataNormals17	Finished in 358 msec

The **Message Area** also includes tabs for Debug Messages, Warning Messages, Critical Messages, and Fatal Messages. The copyright notice at the bottom reads: Copyright VCreate Logic (P) Ltd. www.vcreatelogic.com

Gallery....

The screenshot displays the SequelGUI software interface. The main window is titled "C:/Projects/SequelGUI/bin/SequelGUIProjects/matrix_converter.sqproj - SequelGUI". The interface includes a menu bar with options like Project, Solver, Edit, Zoom Tools, Graphs, and Script. Below the menu bar is a toolbar with various icons for file operations, solving, editing, and zooming. The central area is the Circuit Editor, showing a complex circuit diagram with various components and connections. On the left side, there is an Element Explorer panel with a "Look For" dropdown and a list of components. On the right side, there is a Configuration Panel with a table of parameters and a Property Editor. At the bottom, there is a Message Area with a log of system messages.

Element Explorer

- Look For
- ammeter.ece
- ammeter_fb.ece
- c.ece
- cccs.ece
- ccvs.ece
- clock.ece
- cmptr_1.ece
- cmptr_2.ece
- connector_e.ece
- dac1.ece
- dac1a.ece
- dac4.ece
- diode_r.ece
- diode_r_c.ece
- filter_1.ece
- gate_pulse_1.ece
- ground.ece
- ground_dummy.ece
- indmc.ece
- invtr.ece
- invtr_1.ece
- ipulse.ece
- isrcac.ece
- isrcdc.ece
- isrc_1.ece
- l.ece

Configuration Panel

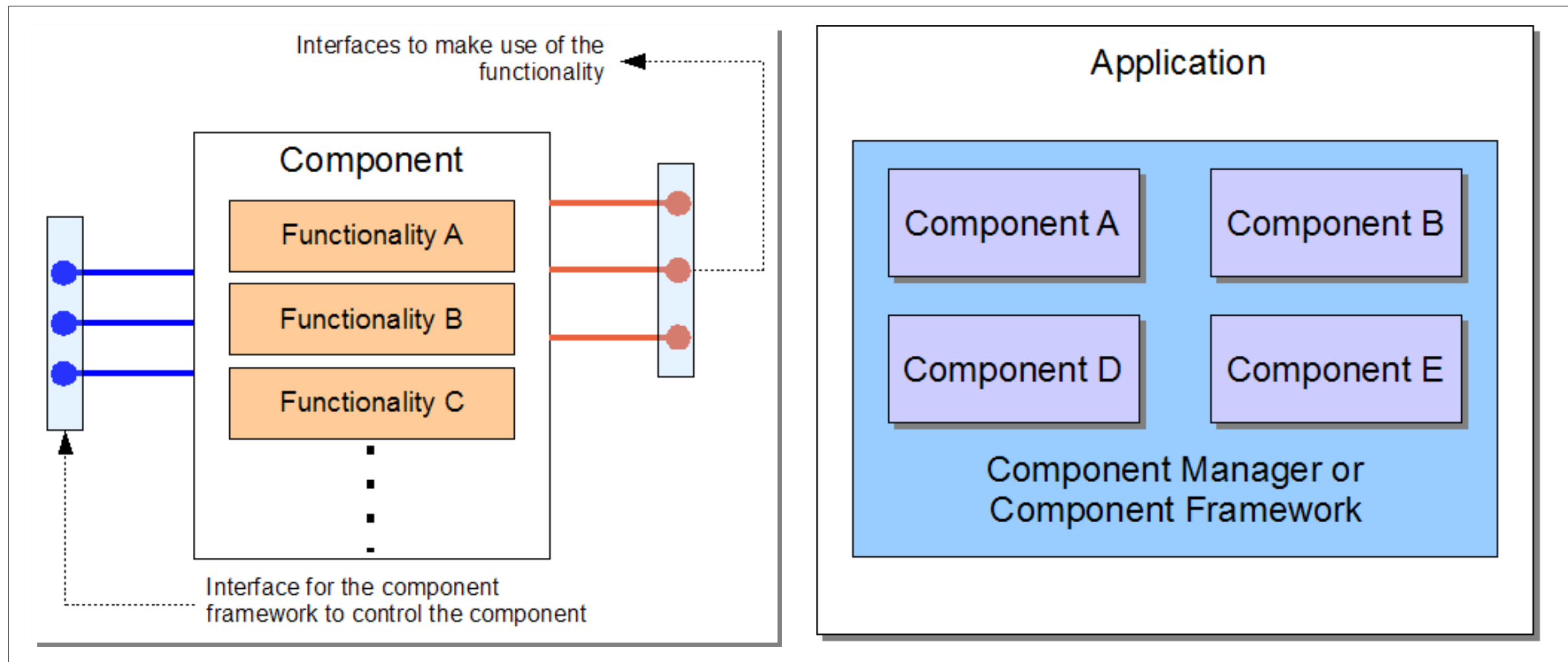
Name	Value
Element	
Comment	
Caption	gce_sm_control
Rotation	Zero
Flip	None
Nodes	
theta_sv	theta_sv
theta_sc	theta_sc
alpha_in	alpha_in
alpha_out	alpha_out
saa	saa
sab	sab
sac	sac
sba	sba
sbb	sbb
sbc	sbc
sca	sca
scb	scb
scc	scc
rparams	
m	0.866
b1	0.523

Message Area

```

Registering circuit file Part writer - Solve blocks
Loading component "CModelEditorComponent"
Loading component "CSymbolEditorComponent"
Loading component "CScriptEditorComponent"
Loading component "CClipboardComponent"
Loading component "CHelpSystemComponent"
Help Collection File is: C:/Projects/SequelGUI/bin/Documentation/SequelGUI.qhc
  
```


Overview



How to Divide and Rule

- Package functionality into “GCF Components”
- A GCF component offers
 - One or more widgets (subclasses of QWidget)
 - One or more objects (subclasses of QObject)
 - User interface elements (QAction, QMenu etc)

GCF's Constitution for Applications

- Have a singleton component class for each component
- Provide a GUI-XML file for each component declaring the objects, widgets, actions etc exposed by the application and their merge hints
- Serve objects, widgets, actions, menus, toolbars etc from components
- Expose component functionality via interfaces
- Have one or more objects within your object implement them

Similarity in Diversity

- IComponent – base class of GCF components
- IComponentPlugin – offers a component from a plugin
- CComponentFactory – dictionary of all components (instantiated, uninstantiated, active and inactive)
- Each component's GUI XML exposes one or more objects belonging to the component
- Interfaces implemented by exposed objects can be searched

The IComponent Class

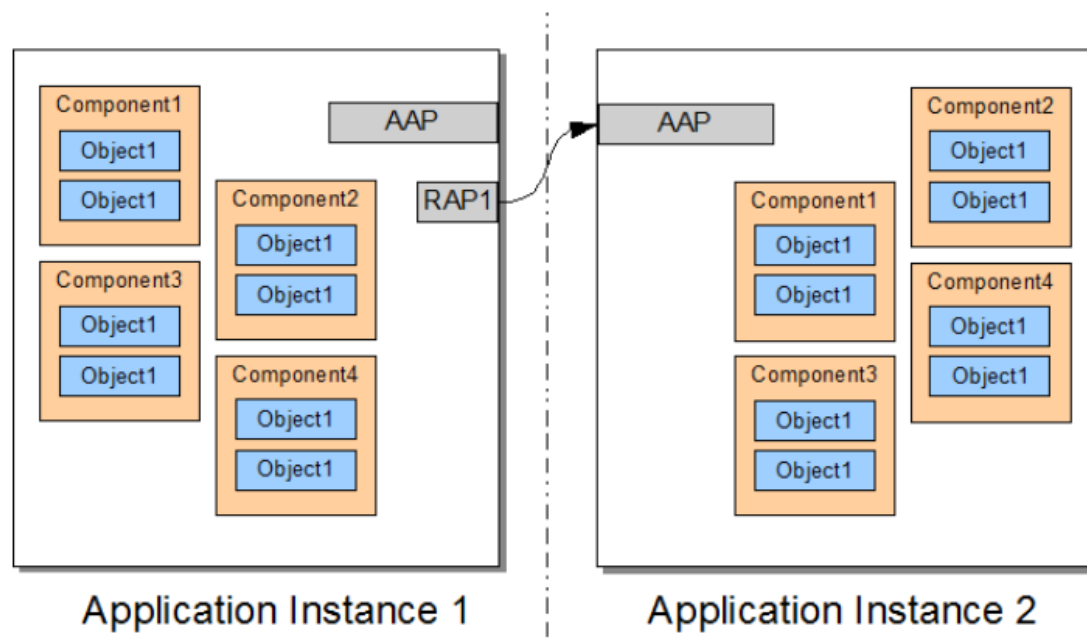
- Interface Methods
 - Methods called by GCF
 - Used for GUI creation and merging
 - Used for handling activation/deactivation of components
 - Reporting component “attendance”
- Service Methods
 - Methods called by components
 - Used for object/service/component discovery
 - Used for establishing and releasing component dependency

Bring them together...

- CComponentLoaderComponent and CComponentGui
- GUI XML File
 - Declares objects, widgets, actions, menus, menu-strip groups
 - All declared objects are called “exposed” objects.
 - For each “exposed” object in the GUI XML file, a corresponding create() method is called on the IComponent.
- Configuration capabilities
 - Can invoke methods, set property values, make connections etc..

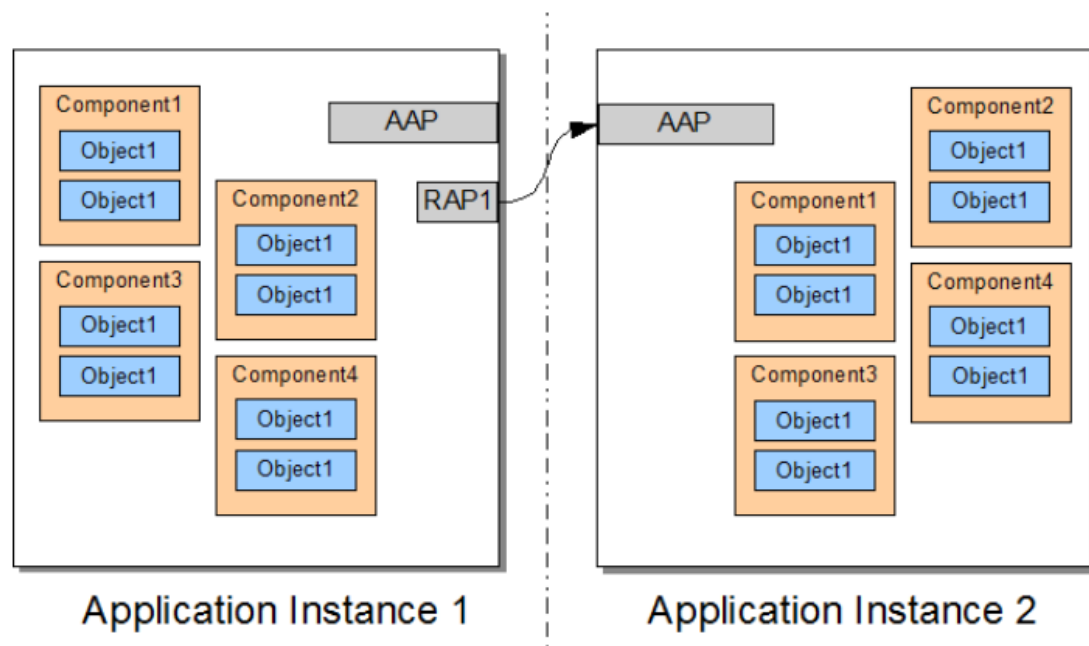
Demo #1

Interprocess Communication



- **Application Access Point**
 - Access to components and its objects are provided through this.
- **Remote Access Point**
 - Helps access objects in a remote application
- **Remote Object**
 - Represents a remote object

Interprocess Communication



- CAppAccessPoint
- CRemoteAppAccessPoint
- CRemoteObject
- CRemoteAppDiscovery
- CMessage
- CRemoteApp
- Based on TCP/IP

Demo #2

Getting and Installing GCF

- From www.vcreatelogic.com/oss/gcf
- SVN <https://svn2.hosted-projects.com/vcreatelogic/GCF>
- Supported compilers: g++, MSVC 2005, MinGW
- GCF designed, developed, maintained by VCreate Logic.
- Is distributed under GPL v2 and GPL v3
- Contributions are most definitely welcome.
- KDCF (commercial GCF) can be purchased from KDAB.
For more information write to sales@kdab.net

Documentation and How to learn?

Qt Assistant - VCreate Logic

File Edit View Go Bookmarks Help

Contents

- Qt Assistant Manual
- Qt Designer Manual
- Qt Linguist Manual
- QMake Manual
- Qt Reference Documentation
- GCF Manual**
 - Articles
 - Basic Concepts
 - Introduction to Compon...
 - Introduction to Generic ...
 - GUI XML Format
 - Interprocess Communic...
 - Tools
 - GCF Help-Qt Assistant in...
 - CreateAppemplate
 - DoxytagToQHP
 - Prepackaged Components
 - Help System
 - Message Log
 - Project Manager
 - Property Editor
 - Script Editor
 - Main Window
 - Utils
 - Clipboard

Search GCF: Generic Component Framework

GCF
Generic Component Framework

Home | All Classes | Modules | Articles | GCF Home | VCreate Logic Home | VCreate Logic
visual meaning to concepts

Generic Component Framework

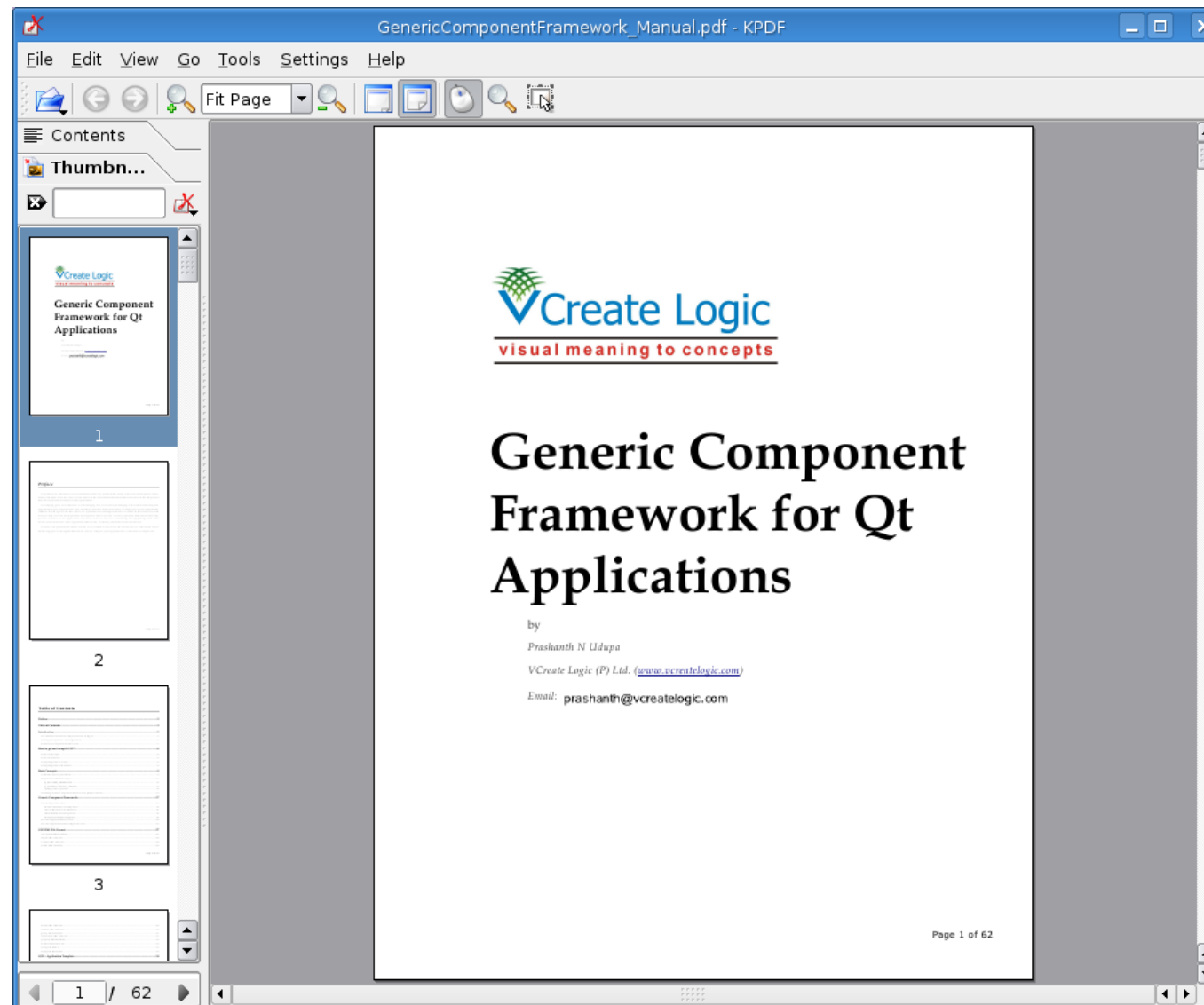
1.6.0

Developing good user interfaces is a challenging task. Even more challenging is the task maintaining and upgrading huge UI applications. Any developer who has been involved in writing large scale UI application software would agree with this. There are a plethora of UI Widget Libraries available in the software world today to help ease out UI application development. There are also several application frameworks that help provide

Articles	Inbuilt Components
Basic Concepts	Standard Interfaces
Introduction to Component Frameworks	Help System Component
Generic Component Framework Brief	Message Log Component
GUI XML Format	Project Manager Component
Interprocess Communication	Property Editor Component
	Script Editor Component
	Main Window Component
	Clipboard Component
	Utils Component

Contents Index Bookmarks

GCF Manual



Whats next? - GCF's Roadmap....

- Support for KPart components
- Support for ActiveX components
- Support DBUS based IPC
- Support for IPC via shared-memory
- Desktop-class component deployments
- Auto-Update mechanisms
- More built-in components

Thank You

Speaker: Prashanth N Udupa



Email: prashanth @ vcreatelogic . com