Sophisticated visualization with Responsive Design

Patrick Rohloff, 3S Smart Software Solutions

CODESYS Users Conference 2016
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to CODESYS Visualization</td>
</tr>
<tr>
<td>2</td>
<td>Responsive Design</td>
</tr>
<tr>
<td>3</td>
<td>Tips for navigation</td>
</tr>
<tr>
<td>4</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
General information

- Visualization is an integrated part of the programming system
- Includes the following features
  - Visualization
  - Recipes
  - Alarm handling
  - Trace / trend monitoring
- Use cases:
  - Machine or plant handling („HMI“)
  - Commissioning and/or diagnosis of applications
Overview of available platforms
Overview of platforms

- Visualization in the programming system
  - Commissioning and/or diagnosis of applications
  - No requirements as to the controller

- CODESYS TargetVisu
  - Display and operation directly at the controller

- CODESYS WebVisu
  - Integrated web server in the controller
  - Display on all current web browsers
  - Display on the PC and/or on mobile devices
Introduction

- Originally developed for display of web contents on different devices

**Responsive web design (RWD)**

is an approach to web design aimed at crafting sites to provide an optimal viewing and interaction experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from desktop computer monitors to mobile phones).
Introduction

- Transferred to CODESYS Visualization
  - Optimal view and operation, at best without scrolling
  - For a large range of devices
    - Small/large TargetVisu
    - WebVisu on small/large end devices (Smartphone, tablet, PC)
    - Display with the most different resolutions

- Size and quantity of information displayed dependent on
  - Usable resolution
  - Available space
Mechanisms for implementation in the CODESYS IDE

- Simple scaling of the visualization
  - Using available space
  - Amount of information stays the same
Mechanisms for implementation in the CODESYS IDE

- Different starting pages
  - Call of special starting pages defined by the user, combined with scaling
  - Adaptation of size and information to available screen size
Controlling the behavior in the PLC program

- Automatic optimization
  - Visu libraries give information about interfaces
  - Necessary information is provided.
    - Visualization type
    - Available resolution
  - A few lines of code are sufficient to branch out from this point.
  - Typically automatic redirection to certain starting pages
  - Can be combined with scaling
Controlling the behavior in the PLC program

- The ClientManager is located in the system center.
- He knows all visualization clients available.
Controlling the behavior in the PLC program

The ClientManager allows for a registration of the ClientManagerListener.

METHOD AddClientManagerListener
    (pClientManagerListener : POINTER TO IClientManagerListener);

ClientManager

AddClientManagerListener()
Controlling the behavior in the PLC program

The ClientManager informs all registered ClientManagerListeners about opening and closing visualization clients.
Controlling the behavior in the PLC program

Structure of the IClientManagerListener

This variable allows for access to the data of the current VisuClient.
Controlling the behavior in the PLC program

VisuStructClientData provides us with useful information.

- `pClientData^.GlobalData.CurrentUserName` supplies the current user name in case of existing user management.
- `pClientData^.GlobalData.GlobalClientID` is adequate to the variable „CURRENTCLIENTID“.
- `pClientData^.GlobalData.ClientType` informs about the visu client type (e.g. ProgrammingSystem, TargetVisualization, WebVisualization, ...).
Controlling the behavior in the PLC program

**VisuStructClientData delivers useful information**

- \( \text{pClientData}^.\text{rClientRect}.\text{ptBottomRight}.iX \) gives information about the resolution width of the VisuClient.
- \( \text{pClientData}^.\text{rClientRect}.\text{ptBottomRight}.iY \) gives information about the resolution height of the VisuClient.
- In case of a WebVisu, you can identify the IP address of the client by using the FB Visu.Elems.VisuFbClientTagDataHelper.
Controlling the behavior in the PLC program

With IClientManagerListener2 you can determine your preferred start visualization

```plaintext
{attribute 'DG_META_COMMENT' := 'Notification, that a client has set the start visualization after the client was created'}

METHOD StartVisualizationSet : BOOL

VAR_INPUT
  pClientData : POINTER TO VisuElems.VisuStructClientData;
  stStartVisu : STRING;

END_VAR
```
Controlling the behavior in the PLC program

Use the global VisuManager to set the visualization

The method `SetMainVisu` allows for toggling the visualization of the specified VisuClient

```plaintext
METHOD StartVisualizationSet : BOOL
VAR_INPUT
  /* A pointer to a VisuStructClientData structure identifying the client, that has been created*/
  pClientData : POINTER TO VisuElems.VisuStructClientData;
  /* The name of the start visualization*/
  stStartVisu : STRING;
END_VAR
VisuElems.g_VisuManager.SetMainVisu(pClientData, stVisu);
```

String named according to the visualization of your choice
Sample implementation
Sample implementation

<table>
<thead>
<tr>
<th>User Name</th>
<th>ClientID</th>
<th>ClientType</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>TargetVisualization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>WebVisualization</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>WebVisualization</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>WebVisualization</td>
<td>127.0.0.1</td>
</tr>
</tbody>
</table>
Potential requirement

- Identical navigation structure on all visualization clients
- However, individual adaptation of the visualization pages to the terminal used

- Another operating concept may be used.
- Easy handling in case of navigation modification or the adding of new pages
Potential requirement

- Each rectangle corresponds to a visualization page.
- Navigation follows the tree structure.
Possible solution

- Depicting a tree structure in CFC
- Fast integration of new pages
- Easy modification of structure
Possible solution

- Each starting page of a visualization client contains a MainFrame.
- The MainFrame contains all sorts of visualization pages.

- The Frame pages are designed according to the resolution of the client type.
Possible solution

- Selection via the frame property „Switch Frame Variable“
- Implementation of the navigation in a flat list or in a tree structure
- The variable udiID describes the position within the frame.

```plaintext
_aVisu : ARRAY[1..11] OF TVU.VisuTreeNode := [
    (udiID := 1, sName := 'Machine'),
    (udiID := 2, sName := 'Error'),
    (udiID := 3, sName := 'CIP'),
    (udiID := 4, sName := 'Process'),
    (udiID := 5, sName := 'Circulation Pump'),
    (udiID := 6, sName := 'Tank Water'),
    (udiID := 7, sName := 'Tank Caustic'),
    (udiID := 8, sName := 'Tank Acid'),
    (udiID := 9, sName := 'Recipes'),
    (udiID := 10, sName := 'Options'),
    (udiID := 11, sName := 'Help')
];
```
Possible solution

For each VisuClient, the visualization page needs to be defined independently.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elementname</td>
<td>GenElemInst_40</td>
</tr>
<tr>
<td>Type of element</td>
<td>Frame</td>
</tr>
<tr>
<td>Clipping</td>
<td></td>
</tr>
<tr>
<td>Show frame</td>
<td>No frame with offset</td>
</tr>
<tr>
<td>Scaling type</td>
<td>Anisotropic</td>
</tr>
<tr>
<td>Deactivate the background drawing</td>
<td></td>
</tr>
<tr>
<td>References</td>
<td>Configure..</td>
</tr>
<tr>
<td>Position</td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td></td>
</tr>
<tr>
<td>Color variables</td>
<td></td>
</tr>
<tr>
<td>Look variables</td>
<td></td>
</tr>
<tr>
<td>Switch frame variable</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>MainPLC._MultiClientVisuTreeNodeNavigator.aVisuTreeNodeNavigator[CURRENTCLIENTID].udiCurrentID</td>
</tr>
<tr>
<td>State variables</td>
<td></td>
</tr>
<tr>
<td>Input configuration</td>
<td></td>
</tr>
<tr>
<td>Access rights</td>
<td>Not set. Full rights.</td>
</tr>
</tbody>
</table>
You define the maximal number of VisuClients in the VisuManager.
Conclusion

- The CODESYS Visualization is very versatile.
  - Different visualization clients are supported.
  - Scaling options allow for easy use of different screen resolutions.
  - If the resolutions differ widely, you can provide the user with different visualizations based on different handling concepts and optimized to the available size in the IEC application.

- Despite all these requirements, easy maintenance and extendibility of the CODESYS project can still be guaranteed.
Thank you for your attention.
Einzelne „Vorlagen“ zur Kopie
Einzelne „Vorlagen“ zur Kopie

Umblätter- ecke