Real time monitor in SVG, a use case in Machining Technology HMI

SVG Open Day 2008
August 28, 2008, Nuremberg

Giulio Roggero PMP®, Ing

Engineering Intelligence for Customer Productivity
Agenda

- Introduction
- Project objectives
- Architecture
- Live demo
- Questions and Answers
Agenda

- Introduction
- Project objectives
- Architecture
- Live demo
- Questions and Answers
What is an HMI (Human Machine Interface)

Main task of HMI are:

- **Show the live** machine process status graphically.
- **Simplify the management** of the machine.
- Support the user in case of **warnings and alarms**.
- Support the user to **control the machine**.
- **Reduce** the operator errors.
A Mikron Machine – NRG 50
miHMI – a solution for all Mikron machine lines

- reduce the maintenance costs
- consolidate software and improve quality
- reduce the learning cure of the operators
Agenda

- Introduction
- Project objectives
- Architecture
- Live demo
- Questions and Answers
Main objectives of miHMI

miHMI is a product developed by Mikron Agno SA, designed to provide a new HMI framework for Transfer Machines. The objectives are:

- **Adaptability.** miHMI will be installed on four different machine lines with different pages and screen resolutions.
- **Scalability.** miHMI must be accessible on local machine and on remote machine and could monitor more than one machine at the same time.
- **Configurability.** Each installation of miHMI require different graphical pages. The effort to create a new page be less that one day.
- **Reliability.** The system must guarantee a latency from the Machine to the user interface below 300 ms.
Graphical layout objectives

- miHMI is designed to support both touch screens and standard screens

- thanks to SVG can support different resolution without loosing the quality

- is composed by a fixed part that highlight important information and menus (.NET WPF) and a dynamic part that shows miHMI SVG pages

- colors and fonts are studied for improving readability also at distance more than 1 meters
miHMI Layout

**MAIN BAR**
Display important information, alarms, warning and messages. Always on top. (WPF)

**SIDEBAR**
Display menus for SVG and external applications. (WPF)

**CONTENT AREA**
Display SVG Pages, Silverlight Pages or stand-alone applications. (SVG/Silverlight)

**STATUS BAR**
Display some live data always on top. (WPF)
Design and implementation approach

- **Modularity**
  - Each graphic part and server logic is a separate module to speed-up the machine configuration for a customer.

- **Remote Access.**
  - Live data can be shown by a PC connected to network using a web browser.
  - It’s possible to scale and consolidate more than one miHMI using Microsoft Message Queue. (eg: to support production lines).

- **Technologies.**
  - C# 2008
  - SVG
  - .NET Web Services
  - .NET 3.0 WPF
  - Silverlight 2
  - AJAX.NET and YUI (Yahoo Libraries)
  - Visio 2003
  - SQL Express 2005
Agenda

- Introduction
- Project objectives
- Architecture
- Live demo
- Questions and Answers
miHMI Architecture
miHMI Modules - Drivers

- communicate directly with the machine hardware
- translate the machine PLC data structure to miHMI data structure
- allows to manage different machine PLC at the same time
miHMI Modules - Servers

- implement monitor and control logic: production, events, trends, programming ...
- are abstracted from the machine implementation because get data from drivers
- are modular and can be enabled/disabled at runtime
miHMI Modules – Web Server

- expose data of servers and drivers via web services
- instrument and provide SVG pages
- allows the remote access to miHMI
miHMI Modules - Client

- visualize live data connecting via web services to the Web Server
- is composed by two main components:
  - MainBar in WPS that visualize top most data and integrate stand-alone applications
  - Web Browser that visualize SVG and Silverlight pages
SVG management
SVG Management - Page load

IE receives the SVG page and loads it into the SVG player. Then loads the JavaScript for runtime updating and adds, to each SVG element, the events for runtime.

The SVG is loaded from the file system. (The path name is in Alias DB)

IE request, via Web Service, an SVG Page. SVG Instrumentaror coordinates the request getting the Page information from the Alias DB.

IE request, via Web Service, an SVG Page. SVG Instrumentaror coordinates the request getting the Page information from the Alias DB.

SVG Object Model is generated by SVG Instrumentaror during the parsing and is stored in SESSION for runtime SVG updating.

SVG Instrumentaror parses the Page and links the JavaScript needed by IE for updating the SVG at runtime.

SVG Management – Live Data

The browser gets back a string with the list of JavaScript functions to call. Each function modifies the value or format of the Stencil using DOM.

Meanwhile the JavaScript, using AJAX.NET libraries, polls asynchronously, via Web Service, with a timeout of 100ms, the IIS queue and get the new message that contains the new value and its quality.

The Stencil, via JavaScript, subscribes the Aliases invoking the Data Point Web Service.

Router routes the Subscription, via Message Queues, to the Driver.

Each time a Value on the Machine changes (Alias) the Driver sends a message to the Router that routes that message to the subscribers.
SVG Stencils – some examples

TOGGLE BUTTONS

BARS

PIE
miHMI configuration process
SVG Page design using Microsoft Visio

- Visio page with custom stencil
- List of machine elements and pages
- Automatic SVG instrumentation
- HTML page with AJAX calls
SVG Stencil configuration

<table>
<thead>
<tr>
<th>Type</th>
<th>TOGGLEBTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>WC01.KF6107.RTDATA.AIR.STATUS</td>
</tr>
<tr>
<td>OnClickAlias</td>
<td>WC01.KF6107.RTDATA.AIR.SET</td>
</tr>
<tr>
<td>OnClickValue</td>
<td>1</td>
</tr>
<tr>
<td>OnReleaseAlias</td>
<td>WC01.KF6107.RTDATA.AIR.SET</td>
</tr>
<tr>
<td>OnReleaseValue</td>
<td>1</td>
</tr>
<tr>
<td>StatusValues</td>
<td>0,1</td>
</tr>
<tr>
<td>StatusColors</td>
<td>RED, GREEN</td>
</tr>
<tr>
<td>StatusTexts</td>
<td>$$$AIROFF$$$, $$$AIRON$$$,</td>
</tr>
<tr>
<td>OnClickType</td>
<td>Momentary</td>
</tr>
<tr>
<td>OnReleaseType</td>
<td>Momentary</td>
</tr>
</tbody>
</table>
SVG source code generated by Visio
performances over an SVG that monitor 100 Aliases at the same time
Agenda

- Introduction
- Project objectives
- Architecture
- Live demo
- Questions and Answers
miHMI demo – Lego Mindstorm® NXT

miHMI Clients
- Mobile Systems
- Mikron HMI Main Bar
- Web Browser (WB)

miHMI Web Server
- ASP.NET Page Controller (PC)
- SVG Engine (SVG)
- Web Service Interface (WS): Subscribe, Change Of Values, Command
- Change Of Value Queue (CDV) – A queue for each drive

miHMI Server Queue (SQ)
- Message Manager / Message Router

Server Plug-ins
- Maintenance Server
- Production Server
- Event Server
- CDV: Maintenance, Production, Event

Common Machine Driver (OM)
- Bosch Driver
- Siemens Driver
- Fanuc Driver
- Other Drivers...

LEGEND
- CI Components
- Queue
- External Comp.
- Machine Driver

Live
Effort to implement this demo using miHMI framework

- **NXT SVG PAGES**: 1 hour (Visio e Photoshop)
- **NXT DRIVER**: 4 hours (C#)
- **CONFIGURATION**: 2 hours for testing

**TOTAL 8 hours!**
LET SEE THE DEMO...
Agenda

- Introduction
- Project objectives
- Architecture
- Live demo
- Questions and Answers