

Human Machine Interface

Annu Somra*

Rajasthan College of Engineering for Women, Jaipur, Rajasthan, India

Abstract

An industrial control user interface provides a flexible, extensible architecture for the provision of real-time process data to a state-of-the-art user interface, using MSHTML as the underlying rendering engine. The architecture of the interface provides a user interface that is designed to harness key industry standard technologies in an open, adaptable, architecture that facilitates the technological convergence of disparate HMI products. The preferred embodiments offer open display page architecture via the separation of the provision of data and server specific user interactions from the implementation of individual page elements. The only real assumption made about a page element is that it is part of the HTML display page and accessible via the Document Object Model.

Keywords: User interface; Document object model

Description

Field of invention

The present invention relates to a human machine interface (HMI).

The invention has been developed primarily for Industrial, Security, Life Safety, Access Control, Heating, Ventilation and Air Conditioning applications such as the monitoring of data generated by industrial processes and equipment, and will be described hereinafter with reference to that application. However, it will be appreciated that the invention is not limited to use in that particular field (Figure 1).

Background of the invention

Remote monitoring of data is frequently used in industrial settings. The desire for implementing such monitoring ranges from the need for centralized monitoring of relatively complex industrial systems to the convenience of allowing a small number of operators to monitor multiple data streams via a single workstation. In many cases, it is also desirable that the HMI allow operators to control at least some of the industrial processes and machines in relation to which the data is being collected.

Historically, collection and display of industrial data on operator workstations has been dealt with on a proprietary basis, with application-specific software linking remote data sources with dedicated terminals. Whilst such systems have provided satisfactory performance, they are relatively time consuming to design, and relatively inflexible once implemented.

Another problem is that individually implemented software solutions are often limited in the types of data sources with which they can interface.

One of the most profound developments of the last five years has been the staggering growth of the internet. This growth has changed public perceptions about the availability of data, about what to expect in a user interface, and how open and interoperable computer systems should be. Standalone computer systems are no longer acceptable to the market - the increasingly networked world is setting the pace and scope of change.

In addition, this growth has led to a huge shift in development focus for practically every software organization. The web has become the ubiquitous data delivery mechanism, and as software organizations focus their energies on harnessing its potential, the technologies which

underpin its success are being pushed forward at ever-increasing rates. Where once the web was not mature enough to support mission-critical functions, such as major financial transactions or process control, it is clear that this is no longer the case (Figure 2).

It is this growth which is driving investment into user interface technology, as the major stakeholders seek to establish the internet as a viable business platform. As a result, current technological trends have meant the underlying technologies are becoming increasingly more appropriate to the field of industrial process control.

The development of what may be loosely termed "internet technologies" has been extremely rapid, and has a very short history:

- December 1993: Only 200 known http servers existed.
- December 1994: First W3C meeting. Over 200 members, charter to promote interoperability.
- October 1996: OLE Controls 96 Specification published. Promotes lightweight, windowless controls suitable for the web environment, and intended to add increasing sophistication to browser capabilities.
- December 1996: Cascading Style Sheet specification. Introduced 2D positioning, size, colour, font attributes.
- July 1997: HTML 4.0 specification. Included scripts, objects, framesets, internationalisation. Browser technology advanced appropriately.
- September 1997: Dynamic HTML, Document Object Model. Full access to the HTML object model, providing web pages with unprecedented power and flexibility.
- September 1998: Internet Explorer 5.0 Beta 2. Includes Vector Markup Language for vector graphics capabilities.

***Corresponding author:** Annu Somra, Rajasthan College of Engineering for Women, Jaipur, Rajasthan, India, Tel: 01412251249; E-mail: annusomra@gmail.com

Received September 09, 2016; **Accepted** December 22, 2016; **Published** December 29, 2016

Citation: Somra A (2016) Human Machine Interface. J Electr Electron Syst 5: 208. doi:10.4172/2332-0796.1000208

Copyright: © 2016 Somra A. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

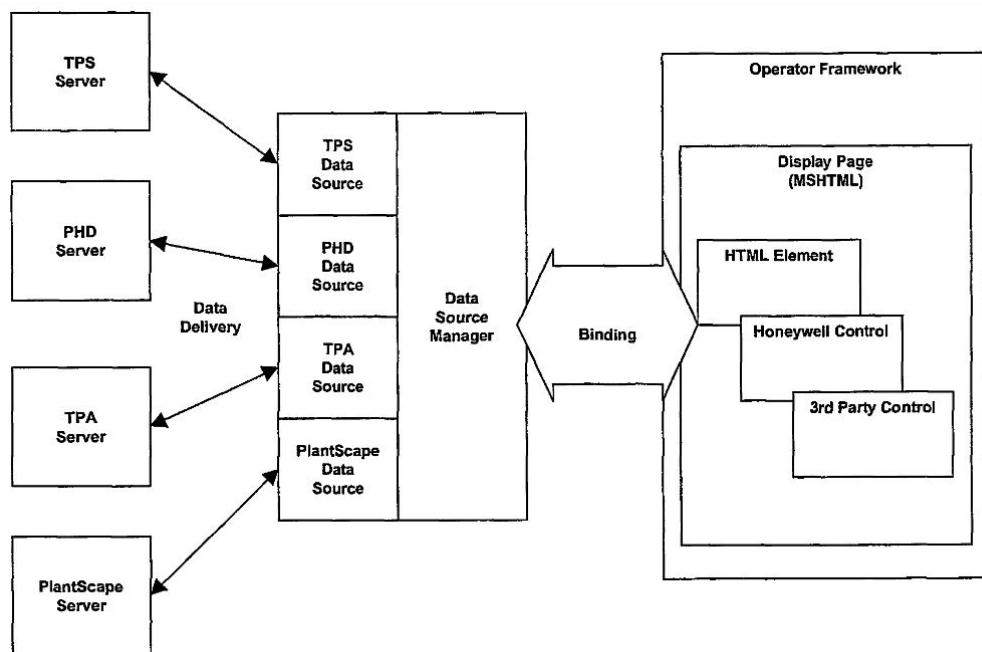


Figure 1: Schematic overview of the architecture according to the invention.

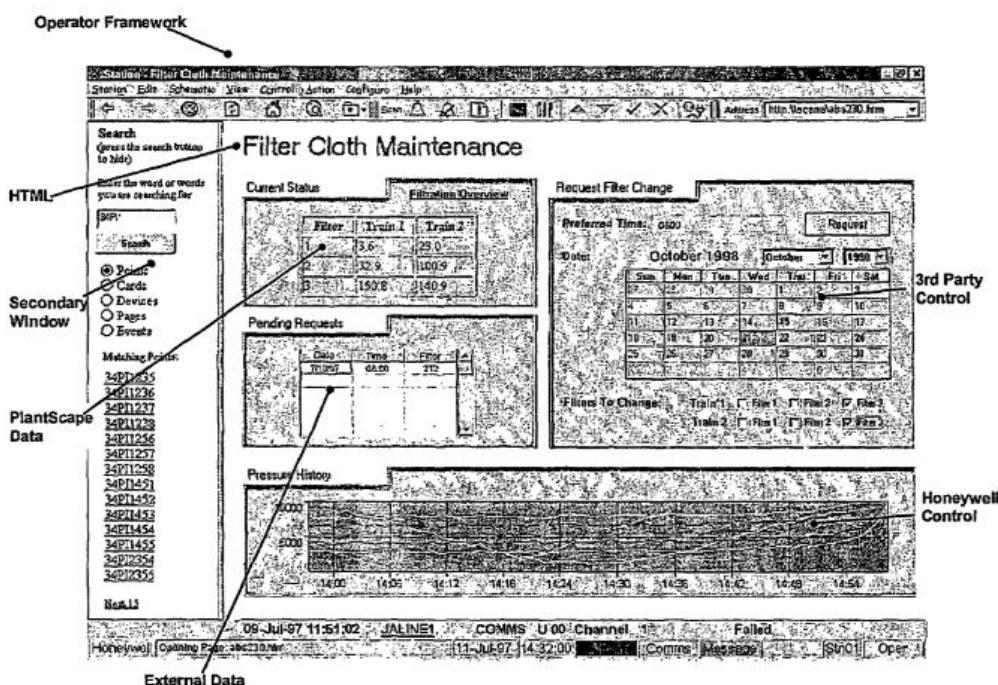


Figure 2: Schematic representation of an operator runtime environment of one embodiment of the invention.

- late 1999: Office 2000 to use HTML/XML as document format for all suite products. Microsoft look to include sufficient functionality in HTML for it to be suitable for use by Visual Basic forms engine.

In parallel with these developments has been an increasing awareness that HTML's origins as a language intended to store presentation information means its ability to store and represent data is

severely limited. This limitation has led to the rise of XML, or extensible Markup Language, intended to work in concert with HTML to provide a means of storing both data and presentation information.

It was these trends that led the applicant to consider the suitability of such technologies for industrial control systems.

The development requirements of an industrial control system indicate clearly that known web browsers are not suitable for the

operator environment of an industrial control system. They may be acceptable for casual use, but the operator environment has specific requirements, such as restricted navigation, support for industrial keyboards, alarm and status indication, and security.

The partial solution to a number of these issues may be provided using a software package sold and marketed by Microsoft®. In particular, Microsoft® supply a rendering engine used in Internet Explorer (known as MSHTML), for use by third-party developers wishing to integrate HTML rendering capabilities into their applications. While this software package can represent a useful tool for system designers, its generic nature ensures that it will not automatically integrate or interact with other applications. Further although the invention in some aspects will be described in relation to this software package, it will be appreciated that other browser techniques may be used.

Disclosure of the invention

It is an object of the present invention to overcome or ameliorate at least one of the disadvantages of the prior art, or to provide a useful alternative.

According to a first aspect of the invention there is provided a Human Machine Interface (HMI) including:

- a display page including a plurality of display page elements;
- a data source manager including a plurality of data source components; and
- a binding engine which transfers data and commands between the data source components and the display page elements.

Preferably, the display page further includes a plurality of infrastructure components. More preferably, the data source components are server system specific data source components. Even more preferably, each display page has its own corresponding binding engine.

Preferably also, the individual data source components are shared by more than one binding engine (Figure 3).

In a preferred form, the display page acts as a container for the page elements that make up the display page and provides the primary user interface thread of execution. More preferably, the page elements include any one or more element that can be included in an HTML file. More preferably, the page elements include one or more of:

- ActiveX controls;
- VML graphics elements;
- HTML elements;
- HTML scriptlets; and
- Java Applets.

Preferably, the display page is constructed using a mixture of standard HTML and XML tags. More preferably, the HTML describes presentation aspects of the display page—that is the page elements—while the XML tags describe what data is required for the page and how that data is to be applied to the page. Even more preferably, the infrastructure components assist with the parsing of the XML content of the display page, the delivery of data to the display page and the initiation of server commands from display page elements.

In a preferred form, at run time, the display page appears as an instance of a standard Document Object Model (DOM). More preferably, the DOM is the standard for the industry to which the HMI is applied.

Preferably, the DOM provides the destination for data provided by the binding engine. More preferably, the DOM further provides the basis for the display page scripting environment.

Preferably also, the display pages are capable of being encapsulated and re-used as encapsulated displays. More preferably, the

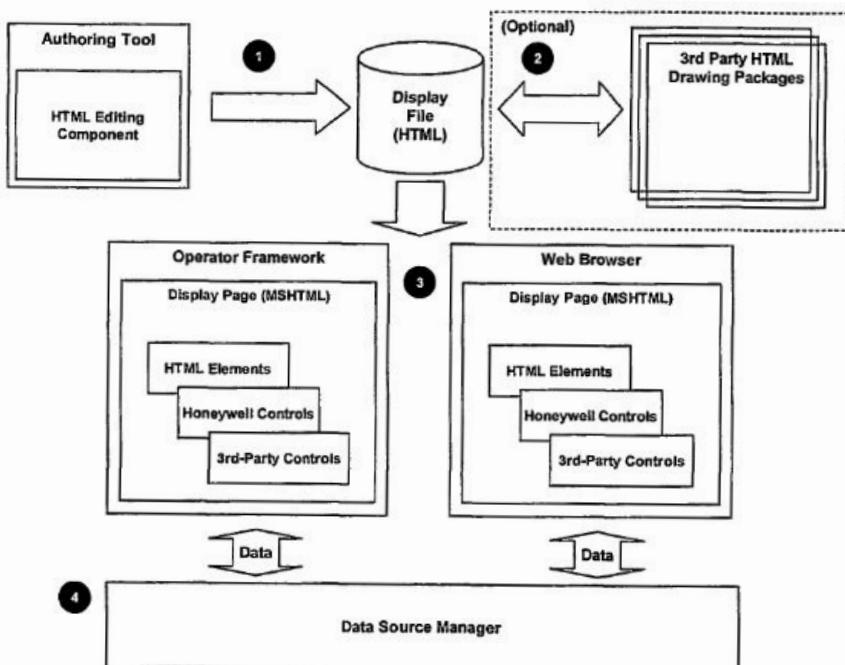


Figure 3: Schematic overview of the display authoring that is utilised in preferred embodiments of the invention.

encapsulation includes the parameterisation of any data references in the display page and the addition of properties, methods and events that allow the encapsulated display to act as a fully fledged component. Even more preferably, the encapsulated displays are embeddable. More preferably still, the encapsulated displays are linked into containing display pages.

Preferably, the display page is HTML based.

In a preferred form, the data source manager manages a plurality of server system specific data source components that encapsulate the details of delivering data to and from particular server systems. More preferably, each data source component presents data from a server system in the form of a hierarchical object model. Even more preferably, the data source manager pulls the separate data source component object models together into a unified data source object model (DSOM) that is used as the source of data for the binding engine [1-5].

Preferably also, the data source components are informed of the data requirements for a particular display page by means of a data source definition that is stored as part of an HTML/XML display page file.

In a preferred form, the server systems include one or more of a variety of different server system, a small subset of which includes the server systems provided by Honeywell Limited and that are known as:

- Plantscape;
- Enterprise Buildings Integrator;
- TPS;
- TPA;
- QCS;
- Uniformance;
- OPC; and
- HCI.

Conclusion

Preferably, the data binding engine takes the data provided by the data source object model and applies it to the display page. More preferably, the data binding engine takes the data provided by the data source object model and applies it to the display page in a way defined by binding definitions contained in the HTML/L display page. Even more preferably, each display element that requires data has an associated binding definition that defines what data is required for the element and how it is to be applied to the element.

Preferably also, the data binding engine is able to bind data to any property of the DOM. More preferably, the DOM includes the body element and any container elements that are used to organise other elements on the display page. More preferably, the binding engine uses transformations to perform the transfer of data from the data source object model to the display page. Even more preferably, the transformations transfer the data directly from the data source object

model to the display page. It is also preferred that the transformations transforms the data as they transfer the data.

Discussion

In a preferred form, the transformations include user written “OnDataChange” scripts and data driven page element “dynamics” such as rotation, path animation and break point animation. More preferably, the transformations are binary components. Even more preferably, the transformations are written using an XML syntax and script code.

In a preferred form, the binding engine executes in an apartment managed by the data source manager and transfers data from the data source object model to the display page in a very efficient manner.

The preferred embodiments of the invention provide an operator framework built around MSHTML to provide the functionality specific to the needs of an industrial control system. Additionally, it has been found that these embodiments, using MSHTML as the underlying rendering engine, provide a flexible, extensible architecture for the provision of real-time process data to a state-of-the-art user interface. That is, the preferred embodiments make use of synergies that are possible from the interaction between the operator framework and the rendering engine. Again, while the embodiments make use of the MSHTML browser software, other browser techniques would be equally applicable.

References

1. Guillermin P (1996) Fuzzy logic Applied to Motor Control. IEEE Transactions on Industrial Application 32: 51-56.
2. Jain SK, Agrawal P, Gupta H (2002) Fuzzy logic controlled shunt active power filter for power quality improvement. IEE Proceedings of the Electric Power Applications 149: 317-328.
3. Norman M, Samsul B, Mohd N, Jasionita J, Omar SB (2004) A Fuzzy logic Controller for an Indirect vector Controlled Three Phase Induction Motor. Proceedings Analog And Digital Techniques In Electrical Engineering, TENCON 2004, Chiang Mai, Thailand 4: 1-4.
4. Afonso JL, Fonseca J, Martins JS, Couto CA (1997) Fuzzy Logic Techniques Applied to the Control of a Three-Phase Induction Motor. Proceedings of the UK Mechatronics Forum International Conference, Portugal. pp. 142-146.
5. Chiewchitboon P, Tipsuwanpom P, Soonthomphisaj N, Piyarat W (2003) Speed Control of Three-phase Induction Motor Online Tuning by Genetic Algorithm. Fifth International Conference on Power Electronics and Drive Systems, PEDS 1: 184-188.