

# Open Source PC on the Factory Floor Toolkit

A Product Proposal White Paper  
5/1/02

Terry H. Ess  
T. H. E. Solution LLC  
Copyright 2002

## PROPOSAL

Provide an open source framework/toolkit for the development of “PC on the factory floor” applications. Support would be provided for all the major open environment platforms with good development support facilities. This includes:

1. Windows (NT/2000/XP/CE)
2. Windows with RTX
3. Linux
4. VxWorks
5. QNX

## WHY

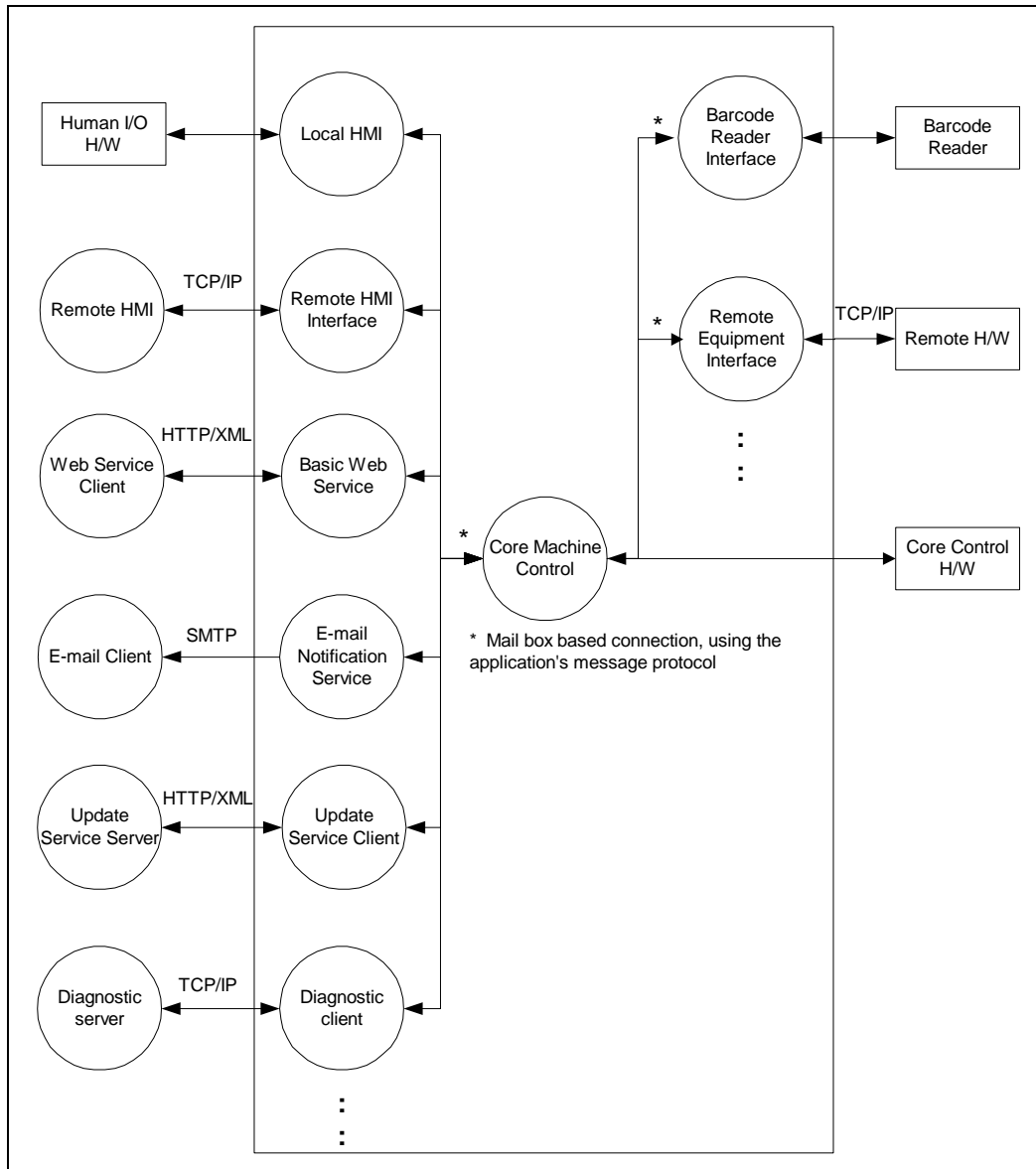
Using PCs for the control of factory floors has been a vision for some time. It has not really happened. Some of the more likely reasons are:

1. Conservative nature of manufacturing.
2. Significant transition costs.
3. Proprietary and therefore restricted nature of available development products and supported platforms.
4. The benefits of moving to PCs are not well understood or quantified.

The potential benefits of environments that allow the factory (or at least the key parts of it) to communicate with the rest of the company are now becoming evident. This will overcome the first item in the list if we can significantly impact items 2 and 3. The proposed open source development framework/toolkit provides development organizations (factory floor machine OEMs, system integrators) a way to do this.

## TECHNICAL DESCRIPTION

The framework’s architecture is summarized in Figure 1.



**Figure 1, Framework Architecture**

The key design points are:

1. Controls complexity and promotes robustness by supporting strong functional decomposition using process level separation.
2. All the parts are tied together using a central “post office” with mailboxes available to each process.
3. Common services/interfaces can be “canned” and provided as part of the framework but this in no way limits the services or interfaces that can be added.

The primary implementation goals are:

1. Easy movement between supported platforms.

2. Support use of appropriate programming language for specific purposes with C++ as the native base.
3. Provide robust exception handling in a distributed environment.
4. Provides mechanisms to support lights out operation scenarios.
5. Easy site/machine level configuration.
6. Robust, language independent function support set for HMIs.

The goals are achieved by:

1. Careful implementation of key components, e.g. the mailbox class, using conditional compiles and separation of platform dependent functions.
2. Post office/mailbox class that includes owner registration and dynamic reconfiguration.
3. Expandable application level, conversation based, messaging.
4. Includes services for automatic software updating, synchronous trouble support (diagnostics) and asynchronous trouble notification (E-mail).
5. Three varieties of HMI GUI (VB 6.0, C# .NET and Personal Java).
6. HMI dynamic library interface provides key basic services to GUI: application conversation access, dynamic language (human) selection, asynchronous communication handling, application logging and integrity checking. Implemented in appropriate formats for the platforms and GUI languages (DLL, SO, JNI).
7. Single machine level configuration file with strong integrity checks.

What makes the framework really useful is the toolkit that creates it. This is a series of Java based applications that support a specific development project. Initially this will include two primary tools:

1. A project generation tool. This will construct the appropriate project set (project directories, make files, source code etc.) for a specific project. It will generate the project space based on the GUI language, platform and desired service/interfaces.
2. A GUI form generation tool. This will generate a specific form with all the basic “plumbing” for the desired GUI language.
3. A core control code generation tool based on flow/state diagram input.

## **CURRENT STATUS**

A significant amount of the framework has been prototyped and checked:

1. A desktop Windows (with or without RTX) prototype using A VB 6.0 GUI with remote HMI and basic web service is completed and available on the web site (<http://www.the-solution-llc.com/release0-2.zip>).
2. A serial barcode reader interface for the Windows prototype is complete.
3. C# .NET and Personal Java based GUIs have been prototyped.
4. A Linux proof of principal prototype using a Java GUI is complete.
5. Both the VxWorks and QNX environments have been researched and the minor design changes required noted.

6. The initial prototype of a commercially viable update service using J2EE technology (Java, servlets, JSP, JDBC etc.) and open source tools (Tomcat and MySql) on the server side is completed.

At this point there should be little technical risk. What is needed is a partner that would help take the product to market. Anyone interested should contact: [terry.ess@the-solution-llc.com](mailto:terry.ess@the-solution-llc.com).