SmartFactory: the challenges of open and low-cost ICT in the small manufacturing industry

Peter Bosscha[a†], Riaan Coetzee[a], Piet Terblanche[a], Albert Gazendam[b] and Sherrin Isaac[b]

We address the neglected field of appropriate information and communication technologies for small companies in the manufacturing industry, through the development of an architecture that integrates various open and low-cost technologies. The SmartFactory architecture reported here consists of open source software, networking and computing equipment that complies with open standards and web-based interfaces made possible by open standards.

Introduction

In small, micro- and medium-sized companies (SMMEs) within the South African manufacturing environment, there is a need for seamlessly integrated suites of low-cost, plug-and-play sensors and monitoring systems in order to monitor process execution and to provide real-time feedback to process controllers and actuators.

Larger manufacturing companies use sophisticated systems to address these and other contributing causes of poor product quality and inefficiencies. A range of integrated systems from PROFIBUS,[1] at the control area network (CAN) level, and SAP[2] at the enterprise resource planning (ERP) and customer relationship management (CRM) levels, are employed with varying degrees of success in complex plants and large corporations. The high costs associated with such systems, however, put them beyond the reach of small manufacturers.

Smaller companies (with fewer than 100 employees) within the manufacturing industry often find it difficult to compete with their larger counterparts. Although less complex and expensive automation and reporting alternatives exist, these tend to cater for the medium-sized manufacturing companies and are not readily affordable for a small manufacturing business. This situation is perpetuated by a perceived lack of market opportunity for solutions to this problem. Thus, absence of economies of scale and the resultant high costs associated with erecting and sustaining a manufacturing facility, makes survival and growth difficult for smaller manufacturing companies.

In a small manufacturing company that cannot afford advanced management systems, there can be various causes for poor product quality and inefficiency, of which the following are typical examples:

- delays in problem identification and intervention;
- poor job scheduling; and
- poor inventory management of both raw materials and manufactured products.

Existing solutions can be divided into advanced and basic systems. A typical advanced solution consists of a full supervisory control and data acquisition (SCADA) system, high-powered programmable logic controllers (PLCs), and interfaces into a company's enterprise resource planning system. The costs of these systems typically run into millions of rands, with the PLC's costs running into the tens of thousands of rands.

The basic systems are often home-grown solutions involving non-networked basic PLCs,[3,4] costing a few thousand rands, to perform limited control functions. These systems cannot be easily upgraded, and typically do not provide networking nor the possibility of advanced control. As a result, in these smaller installations, where a data collection system is installed it is often simply paper based.

The problem that we aim to address is that of limited availability and use of open and low-cost information and communication technologies (ICT) for automation in small manufacturing companies. The solution addresses the above-mentioned causes of poor product quality and inefficiency, is affordable and allows open access for other parties to work on and improve.

For the purposes of this paper, 'open ICT' is defined as:

- open source software, such as Linux, FreeBSD,[5] Mobile Mesh,[6] PostgreSQL,[7] Apache web server,[8] and Compiere ERP+CRM;[9]
- communications and computing equipment that comply with open standards such as 802.11b,[10] Internet Protocol (IP),[11] miniPCI[12] and the IBM compatible x86 personal computer (PC) architecture;[13] and
- web-based interfaces made possible by open standards such as HTTP[14] and HTML.[15]

Solution

There are open source software and other open and low-cost building blocks that can be integrated in an architecture that addresses the requirements of a small manufacturing company. This article presents an architecture that integrates various open and low-cost building blocks into an architecture that we call 'SmartFactory'. From an implementation perspective, the SmartFactory architecture contains two main components. Each of these components enables different functions that address the three influential causes of poor product quality and inefficiency identified above:

- A Control Area Network for factory floor monitoring and factory floor reporting.
- A Central Server providing web-based SCADA and ERP for factory floor history, job scheduling and inventory management.

A review by Neumann[16] and the design goals of the new PROFINET[17] standard confirm that few sophisticated commercial CAN systems provide wireless networking, IP addressability, web-based interfaces and easy configuration — all aspects addressed by the SmartFactory architecture.

Architecture and technologies

Figure 1 shows the system architecture with the various open and low-cost technologies that it integrates. Open source software was chosen for its low cost and is deemed reliable enough for our purpose.[18, 19] Suitable candidates were not available in the public domain for a number of integrative building blocks, which necessitated their development as indicated. Further details on the open and low-cost technologies employed in the architecture are shown in the figure.

A. 802.11b

The established open wireless networking standard 802.11b, otherwise known as Wi-Fi, is suitable for deployment on the factory floor in small manufacturing companies. Experiments[20] conducted by the CSIR have confirmed the robustness of this and similar wireless technologies in various harsh environments, including an electrical sub-station, where significant performance degradation did not occur.

Dire warnings in industry as to interference problems with

[a]Manufacturing Science and Technology, CSIR, P.O. Box 395, Pretoria 0001, South Africa.
[b]Meraka Institute, CSIR, Pretoria.
†Author for correspondence. E-mail: pbosscha@csir.co.za

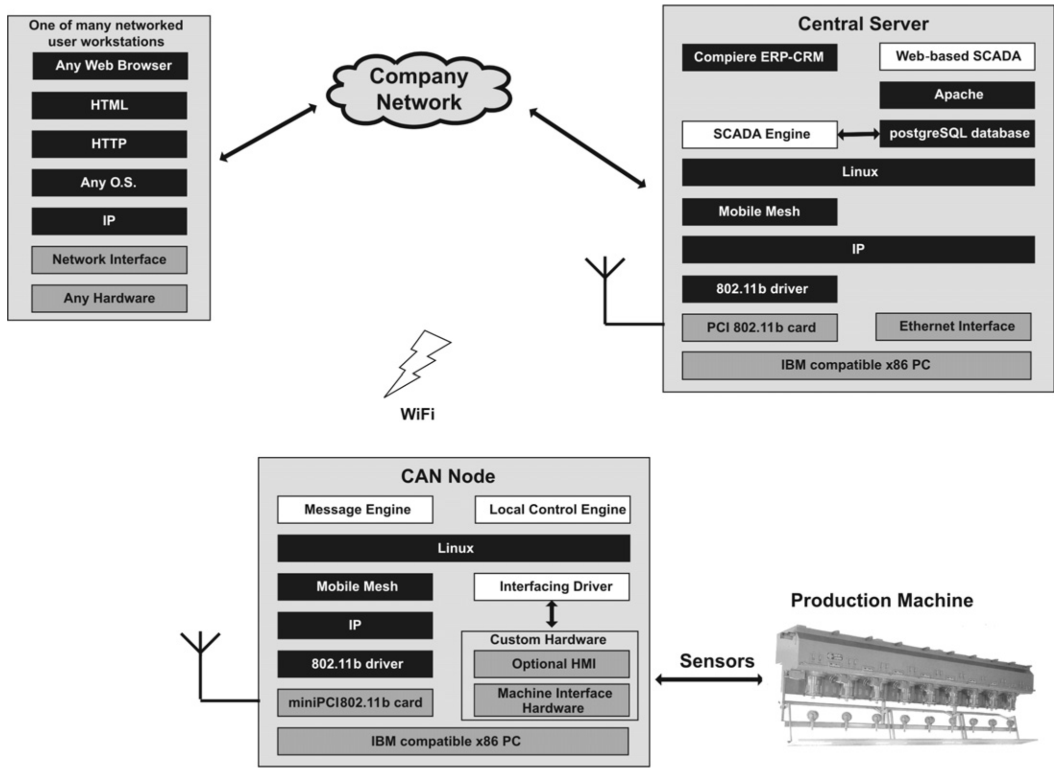

Fig. 1. The SmartFactory architecture consisting of a CAN and central server, each containing various open and low-cost components. Dark grey represents hardware items; black represents standard software items; white, represents software items created for this project.

multiple wireless standards have proved to be partially correct: multiple systems can co-exist but bandwidths will be affected, although not to a large degree.[21] The reporting rate on a CAN node will never be faster than 1 second, and indeed is usually slower, with actual data content in the region of a few hundred bytes. This means that even when the wireless system is impaired due to interference problems, our bandwidth requirements will still be much less than would be available.

The use of 802.11b as networking technology allows for the easy implementation of IP stacks on the nodes of the CAN, thus making them directly addressable and much more flexible in terms of information flow possibilities. Although other 802.11 and unrelated wireless networking technologies could potentially also satisfy most requirements of the SmartFactory architecture, 802.11b's relatively low-cost and superior drivers for various open source operating systems make it the most suitable wireless networking technology for application in small manufacturing companies.

B. Mobile Mesh

The 802.11b standard allows for the nodes of the CAN to be operated in client mode to enable connectivity throughout the CAN, assuming that the central server operates in access point mode. For such a configuration to work, it is necessary that all client nodes are within range of the central server. This might be a problem on a large factory floor, since the range of an 802.11b wireless connection is typically limited to around 100 m when using standard antennas.

Mobile Mesh is an open source tool that enables connectivity throughout the CAN, even when some client nodes are out of range of the central server. CAN nodes function in ad hoc mode as provided for by the 802.11b standard, while Mobile Mesh dynamically updates their routing tables. This permits an out-of-range connection between a client node and the central server to pass through an in-range client node.

It is important to note that the network in this application is intended mainly for data collection purposes. Where control is implemented this will always be local control, making network transport a non time-critical issue. When data transport is more critical (for instance in set-point control), care must be taken to ensure that sufficient redundancy is available in the network paths. In this way nodes that fail can be routed around.

C. FreeBSD and Linux

There are various open source operating systems suitable for deployment on the CAN nodes and central server. Application execution and IP networking are enabled at the FreeBSD equipped CAN nodes, while the Linux-based central server contains the PostgreSQL database performing the historian function, an Apache web server that provides the SCADA interface and the Compiere ERP+CRM server.

FreeBSD and Linux were selected because of the availability of various open source networking and database packages, stable 802.11b wireless networking drivers and the Mobile Mesh tool that was ported to FreeBSD by the CSIR. The superior reliability of open source operating systems such as FreeBSD and Linux makes them very suitable for the SmartFactory architecture, which demands open and low-cost building blocks.

D. Embedded PC

The use of embedded PCs as the CAN nodes is a departure from the conventional CAN architecture that contains PLCs of various intelligence levels throughout the network. In environments where hard, real-time performance is not required, an embedded PC equipped with the necessary interfacing electronics and decision-making logic implemented in software can provide an affordable alternative.

Further benefits include easy application development on an x86 PC platform running FreeBSD, a wide range of hardware suppliers because of the open x86 PC standard and increased flexibility due to the benefits of distributed intelligence. In short, apart from the lack of hard, real-time performance, an embedded PC as a CAN node provides all the benefits of sophisticated PLCs without the high cost and proprietary entanglements.

E. Interfacing electronics

Conventional PLCs have interfaces that embedded PCs lack. These interfaces are used for the sensors and actuators located on the machines of a production facility. The SmartFactory architecture requires the use of interfacing electronics that enable both analogue and digital inputs and outputs. A human–machine interface (HMI) that forms part of the interfacing electronics contributes to the factory floor, reporting functionality identified earlier as being important to the SmartFactory architecture.

PLC vendors often attempt to lock their clients into proprietary solutions. They do this by providing unique programming tools at significant cost, so that a break from the chosen architecture comes at a high price. A client has the option either to purchase the software or to employ an outside contractor at high hourly cost to do the programming. In addition, conventional PLC hardware costs are significantly higher and have less flexibility than that provided by a design built around an embedded PC with external electronic interfacing.

F. Commodity PC

The central server has relatively high performance requirements as a result of the PostgreSQL database, Apache web server and Compiere ERP+CRM that it hosts. An embedded PC is thus not suitable as central server, which is why a commodity PC was selected as hardware platform for the central server.

We acknowledge that the central server is a single point of failure and that the use of low-cost commodity-off-the-shelf (COTS) components increases the risk of total system failure. Should this be a problem, the architecture can easily provide for a mirrored central server that would provide seamless operation in the event of failure of the primary central server. In such an event, the open nature of the x86 PC standard allows for cost-effective maintenance and support from various service and component suppliers.

G. PostgreSQL

The historian functionality is important in order to provide the ability to access data captured from various sensors through a production facility and to perform trends analysis on these data accumulated over long periods of time. The free and open source database PostgreSQL was selected for deployment on the central server owing to its licence being more accommodating than that of the other established open source database, MySQL.

H. Web-based SCADA

The use of supervisory control and data acquisition is well-known in the manufacturing industry. Such systems typically provide detailed information on the state of variables throughout a production facility and often allows for control functionality through its graphical user interface. The SCADA system designed for SmartFactory provides similar functionality through an easy to use web-based interface, which has the benefit of enabling access to the system using any popular web browser from any PC on the company's local area network (LAN). Different levels of user access ensures that information security and plant safety cannot be compromised.

I. Compiere ERP+CRM

The motivation for using an open source ERP system is to remove the high cost associated with traditional ERP packages. For instance, one study[22] has shown that 57% of SAP reference customers had not achieved a positive return on investment. The research found the average cost of a three-year SAP deployment to be about $10 million. There is also ERP software available from Syspro, catering for SMEs at about $50 000 to $100 000.[23] Open source ERP systems do not require purchase costs or licence costs, but may incur setup and support costs (if applicable).

Compiere ERP+CRM is currently the most prominent open source enterprise and resource planning (ERP) software package available. It covers all areas from customer management to supply chain and accounting. The Compiere ERP+CRM project started in January 1999, was registered on SourceForge on 8 June 2001 and, as of 19 January 2005, has had 55 developers. During this period, the project has continuously been among the top 10 projects in terms of development activity according to SourceForge statistics, and the various versions of the software have been downloaded over 800 000 times.[7] In one on-line quick poll of ERP installations conducted, Compiere ERP+CRM's deployment popularity was rated third highest (after SAP and Oracle).[24]

There are add-on, open source projects to Compiere ERP+CRM that would potentially address specific requirements of the manufacturing industry. Two examples registered on SourceForge are Compiere MFG & SCM[25] and Compiere MPCS.[26] These examples are currently quite immature in capability and it is difficult to predict the speed at which development work will proceed.

Results and discussion

Key characteristics of the architecture's enabling components, and motivations for why they are important, are as follows:

- Wireless networking. Owing to the intrusiveness and high cost of cabling as well as the need to accommodate the movement of machines on the factory floor, wireless networking reduces the cost of installation and maintenance in addition to enabling a wider application area.
- Ease of use. Reduces the learning-curve and facilitates installation, maintenance and support either by the small manufacturing companies themselves or by easily trained external consultants.
- Low-cost. Open standards allow for integration with other systems as well as maintenance and expansion using components from various suppliers. This is a further benefit, since the currently available offerings often force lock-in with a specific supplier.

The developed CAN node and attendant HMI interface consists of an embedded PC and accompanying sensor interface electronics. The entire set of hardware is integrated into an industrial housing, which can then be installed at a suitable position in the factory environment at low cost.

Currently, a node provides for eight 16-bit analogue inputs, eight digital inputs, four high-speed counter inputs and eight high-current outputs (relays). The embedded PC caters for the wireless interface, memory and FreeBSD operating system, with attendant ease of higher level scripting and programming. The HMI is a serially interfaced, script programmable unit with a user interface consisting of a graphics display and six key inputs; the interface typically makes use of a menu style interface.

At the current level of SmartFactory implementation, only monitoring and limited control options are available. In future, the intention is to grow this into a fully fledged open source PLC with open source programming tools, as already started by projects such as the MatPLC,[27] Classic Ladder[28] and the Comedi[29] hardware interface libraries.

The SmartFactory concept is still evolving. This first version certainly does not meet all the envisaged goals of ease of use, installation and maintenance, but it does assist in achieving a clearer understanding of what the real-life problems and requirements are. This is evaluated by testing the system against an actual factory environment — in this case an automotive supplier manufacturing materials used for brake linings.

The system was successfully tested in our laboratory environment, but showed practical problems once we attempted installation in the field. Unstable mains supply problems required the addition of an uninterruptible power supply (UPS) for the measuring node, and the factory's specialized sensing requirements caused extensive delays. Because of timing constraints, the system was never fully tested in the factory.

The implementation process of the system has provided us with a number of lessons. On the machine level, there are never enough inputs/outputs (I/O). Thus the systems must be designed in such a way that the I/O requirements in terms of sensors and actuators can be met in a flexible and easily implementable manner. Although the system has the modularity of multiple nodes, a single node should be easily expandable in terms of I/O. Currently, we favour the implementation of finer-grained sensing, such as that provided for in ZigBee[30] systems, although research has to be performed on the co-existence of various networking standards and radio frequencies.

FreeBSD is perhaps not the best solution for a node. With its emphasis on stablility, support for the newer hardware, as for instance the latest USB/UPS solutions, proved to be a problem. A more experimental and advanced solution like Linux would probably be more appropriate. For most industrial environments at this stage, the Microsoft Windows™ operating system is the de facto standard. This means that a Linux-operated server presents a maintenance problem to the factory.

Perhaps the weakest present link in the system is the SCADA level. It is difficult to deliver an easily configured system that is user-friendly and still powerful enough to give a comprehensive overview of the factory. There is a fundamental conflict between trying to cater for every need and keeping things simple for the user. The currently accepted solution for the factory insisting on a higher level of complexity, is commercially available packages.

A remote link into the system is indispensable; offsite support is preferable to an on-site visit lasting a few hours. Not all problems can be solved in this way, but it makes for easier support when addressing the more mundane issues.

Summary and conclusions

The possibility of integrating open and low-cost building blocks in an architecture that addresses the needs of a small manufacturing company has been partially addressed. Our experiences have shown that PC-based implementation at the node side adds unnecessary cost and complexity to the solution. We have, therefore, decided to pursue an implementation based on a simpler and more robust open, embedded solution.

Aside from the technical issues, there are other matters to deal with regarding solution roll-out and industry acceptance. The intention is to attract some SMEs that are in a position to perform these tasks rather than involve the CSIR directly. Some preliminary steps have been taken in this direction, through the AMTS[31] PRIME project.

1. Anon. PROFIBUS International. Online: http://www.profibus.com/ (accessed 15 February 2005).
2. Anon. Systems Applications and Products (SAP). Online: http://www.sap.com/ (accessed 15 February 2005).
3. Anon. Easy Systems, Klockner Moeller. Online: http://www.klockner moeller.com/easy_systems/easy-systems-overview.htm (accessed 15 April 2006).
4. Anon. Logo, Siemens Automation. Online: http://www.automation.siemens.com/logo/index_76.html (accessed 15 April 2006).
5. Anon. The FreeBSD Project. Online: http://www.freebsd.org/ (accessed 5 February 2005)
6. Anon. Mobile Mesh, MITRE Corporation. Online: http://www.mitre.org/work/tech_transfer/mobilemesh/ (accessed 15 February 2005).
7. Anon. PostgreSQL Global Development Group. Online: http://www.postgresql.org/ (accessed 15 February 2005).
8. Anon. The Apache Software Foundation. Online: http://www.apache.org/ (accessed 15 February 2005).
9. Anon. Compiere ERP+CRM. Online: http://www.compiere.org/ (accessed 15 February 2005).
10. IEEE Standards Association (1999). ISO/IEC 802-11:1999: Information technology Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.
11. IETF Network Working Group (1995). RFC 1812: Requirements for IP Version 4 routers. Online: http://www.ietf.org/rfc/rfc1812.txt (access 20 February 2005).
12. PCI-SIG (1999). Mini PCI Specification 1.0. Online: http://www.pcisig.com/ (access 15 January 2005).
13. IBM (1981). Personal computer announced by IBM. Online: http://www-1.ibm.com/ibm/history/documents/pdf/pcpress.pdf (access 20 February 2005).
14. W3C Network Working Group (1999). RFC 2616: Hypertext Transfer Protocol — HTTP/1.1. Online: http://www.w3.org/Protocols/rfc2616/rfc2616.html (accessed 25 April 2005).
15. World Wide Web Consortium (1999). HTML 4.01 Specification. Online: http://www.w3.org/TR/1999/REC-html401-19991224/ (accessed 15 February 2005).
16. Neumann P. (2002). Merging fieldbus and telecommunication systems in the industrial automation domain. In *Proc. IEEE AFRICON 2002*, George, South Africa, vol. 1. Sixth IEEE AFRICON Conference, George.
17. Pöschmann A. and Neumann P. (2004). Architecture and model of Profinet IO. In *Proc. IEEE AFRICON 2004*, Gaborone, Botswana, pp. 1213–1218. Seventh IEEE AFRICON Conference, Gaborone.
18. Working group on Libre Software (2000). Free Software/Open Source: Information Society Opportunities for Europe?, section: Perceived disadvantages of open source models. Online: http://eu.conecta.it/paper/paper.html (accessed 15 April 2006).
19. Anon. Arguments about Open Source. Online: http://www.netc.org/openoptions/pros_cons/comparing.html (access 15 April 2006).
20. Johnson D., Roux K. and Cambridge G. (2003). Evaluation of Bluetooth Technologies. CSIR Report RES/SC/02/19067, Pretoria.
21. Golmie N., Van Dyck R.E. and Soltanian A. (n.d.). *Interference of Bluetooth and IEEE 802.11: Simulation Modeling and Performance Evaluation.* National Institute of Standards and Technology Gaithersburg, MD.
22. Vance A. (2003). SAP costs too much — Customers. *The Register.* Online: http://www.theregister.co.uk/2003/03/31/sap_costs_too_much_customers/ (accessed 22 October 2004).
23. Anon. Homer Computer Services. ERP Software Selection Guide Website. Online: http://www.homercomputer.com.au/erp_guide.htm (accessed 23 October 2004).
24. Anon. MySQL AB. Which ERP applications are currently in use or will be used at your company? Online: http://dev.mysql.com/tech-resources/quickpolls/erp-applications.html (accessed 2 October 2004).
25. Anon. Compiere MFG & SCM. Online: http://www.compiere-mfgscm.sourceforge.net/index.html (accessed 2 December 2004).
26. Anon. Compiere MPCS. Online: http://sourceforge.net/projects/cmpcs/ (accessed 2 December 2004).
27. Anon. MatPLC. Online: http://mat.sourceforge.net/ (accessed 16 January 2006).
28. Anon. Classic Ladder. Online: http://membres.lycos.fr/mavati/classicladder/ (accessed 16 January 2006).
29. Anon. Comedi. Online: http://www.comedi.org/ (accessed 6 January 2006).
30. Anon. ZIGBEE Alliance. Online: http://www.zigbee.org (accessed 16 January 2006).
31. Anon. The Advanced Manufacturing Technology Strategy (AMTS). Online: http://www.amts.co.za (accessed 16 January 2006).

Glossary

CAN	Control area network
CRM	Customer relationship management
CSIR	Council for Scientific and Industrial Research
ERP	Enterprise resource planning
HMI	Human–machine interface. The user interface of a mechanical system, a vehicle or an industrial installation is often referred to as the HMI
ICT	Information and communication technologies
IP	Internet protocol
LAN	Local area network
PC	Personal computer
PLC	Programmable logic controller; an instrument used to measure and control machinery.
SCADA	Supervisory control and data acquisition; typically used to show overview of factory plants and to provide high level control.
SME	Small or medium enterprise
SourceForge	A public repository on the Internet, containing numerous open source projects.
UPS	Uninterruptible power supply; a device which maintains a continuous supply of electric power to connected equipment by supplying power from a separate source when utility power is not available.
USB	Universal serial bus; a worldwide computing serial interface standard.
x86	x86 or 80x86 is the generic name of a microprocessor architecture first developed and manufactured by Intel. The x86 architecture has dominated the desktop computer, portable computer, and small server markets since the 1980s' IBM PC, which ran primarily versions of Microsoft Windows™ and Unix variant operating systems.