On Automating the Network Management in Industrial Automation Systems

Thomas E. Koch, Esther Gelle,
ABB Switzerland Inc., Corporate Research
Segelhof, CH 5405 Baden-Dättwil, Switzerland,
{thomas.koch, esther.gelle}@ch.abb.com

1. Introduction

The installation and administration of large heterogeneous IT infrastructures, for enterprises as well industrial automation systems, are becoming more and more complex and time consuming [1][2]. The growing number of interconnections between networks, the development of new intelligent IT devices, and increasingly sophisticated computer hardware and software, require in-depth knowledge of IT protocols, interfaces, and standards to manage such infrastructures. This and the fast technology cycles make it virtually impossible to manage IT infrastructures centrally. Industrial automation systems present an additional challenge, in that these control and supervise mission critical production sites, which must be up and running 24/7. Nevertheless, it is common practice to manually install and maintain industrial networks and the process control software running on them, which can be both expensive and error prone. In order to address these challenges, we believe that in the long term such systems must behave autonomously. In this paper we want to sketch some issues of such self-managing industrial automation systems.

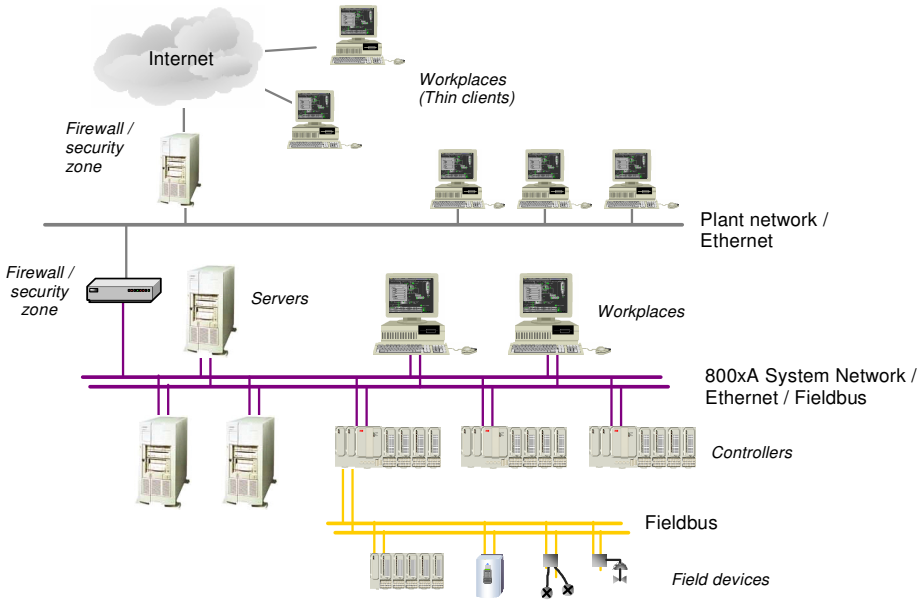

Figure 1: Example of an Industrial Automation Network (with courtesy of ABB Inc.): It consists of many heterogeneous devices like (process) controllers, motors, small machines, routers, switches, servers and client PCs, connected by Ethernet and TCP/IP.

2. Industrial Automation Systems

State of the art: Currently ABB provides with its new automation system 800xA an operator platform for typical automation applications that control and supervise for example a cement or steel production plant [3]. A typical industrial automation network consists of several layers: process, field, group control and process control level (Figure 1). The operator workplace is connected to the control network and shows the operator the current status of the process online receiving a continuous stream of data from the controllers using OPC [4]. It is critical for continuous and reliable operation that not only the technical process is supervised but also the control network itself.

In the 800xA system, the application "PC, Network and Software Monitoring" (PNSM) provides the operator with an overview of the status of the control network and the devices connected to it. It enables the monitoring of IT assets, e.g. computer nodes, routers, printers etc. An IT asset comprises all IT items to be measured such as hard disk usage, network load or number of connections [5]. The IT item as the basic piece of information is retrieved from Windows Management Instrumentation (WMI) via OPC making use of the fact that 800xA runs on Windows [6]. WMI provides an interface for network management applications in Windows and also interfaces to SNMP agents [7].

The 800xA "Network and Device Scanning" tool (NDS) [5] improves the manual and error-prone configuration process. NDS scans the network using ICMP and SNMP, compiles MIB files and provides mapping information from SNMP OIDs to WMI paths (Figure 2). The main benefits of this automation of the former manual configuration steps include reduction time consuming and complex engineering efforts, improvement of the quality of configuration data, and faster integration of new Assets into the Windows operating system repository and PNSM library.

Benefits: The integration of network management of the automation systems' IT infrastructure into a process control system like ABB's 800xA pays off after a very short time [8]. The advantages of this integration through PNSM, OPC and WMI are evident; i) One supervision system for the whole automation system instead of two. Network failures are shown in PCS. ii) No need for extra IT specialists for network management at run-time.

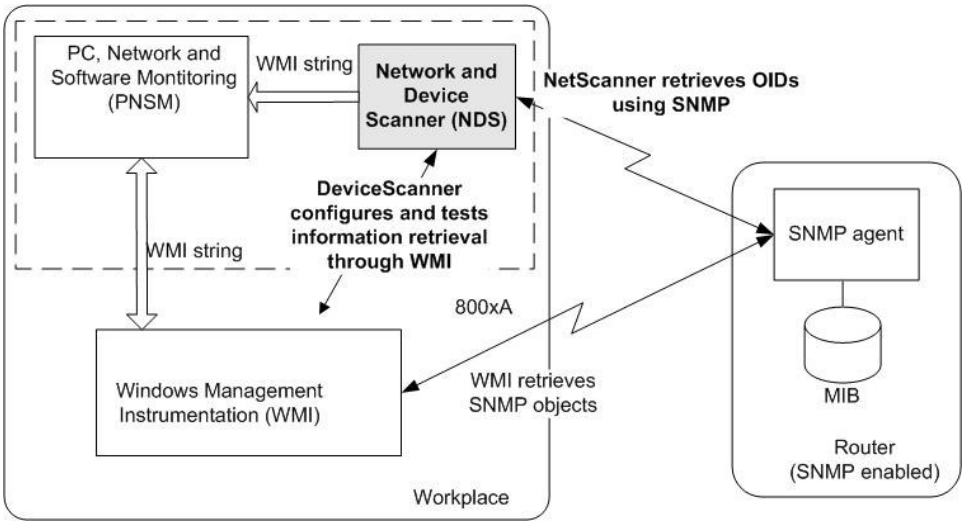

Figure 2: Current architecture of PNSM and NDS.

Challenges: The new challenges in industrial automation networks are manifold. The Ethernet on plant level finds more and more usage, replaces or merges proprietary networks and all the different bus standards (FIELDBUS, PROFINET, etc.). Additionally it connects to the enterprise networks and thus to the Internet. On the lower Fieldbus/Ethernet level this setup implies problems of real time delivery versus broadcast and multicast messages, scheduled clock synchronization and other "unwanted" TCP/IP traffic. Connecting the plant networks with enterprise networks arise more security issues, even with firewalls. Today, the monitored control devices itself have limited resources like CPU power and memory. This complex integration of plant floor to enterprise applications is forced by customers and competition. Thus, new architectures are needed to meet these challenges.

3. Autonomous Industrial Network Management

Vision: In an approach towards the realization of visions like autonomic or organic computing [1][2][9], a new device in the network would automatically register and activate update mechanisms in PNSM. In a first step, this includes scheduled network scans and analysis of differing information, new devices are scanned and configured automatically.

Concerning the autonomic and organic computing visions, IT networks and its IT assets will be self-aware. These components will have these "self"-characteristics. From our perspective presented in this paper, autonomic IT assets such as computers, routers, switches and controllers, for example ABB's AC800M device controller, could acknowledge, install and configure them selves. Additionally they may inform neighbor components in the network of their existence, e.g. by publishing their offered services. In order to achieve these goals, the components need to talk the same language. Data structures, protocols and services must be openly standardized and implemented by vendors.

More specifically, in our application we expect each hardware device and each software component in the architecture to behave like an autonomic component as shown in Figure 3. An autonomic component will publish the services it requires and provide its own contact information and services. Its policies describe optimization criteria respected by the component in its actions throughout the life-cycle. The state of the art in Software Architecture research is that middleware increasingly includes compositional adaptation mechanisms which enable software to modify its structure and behavior dynamically in response to changes in its execution environment [10]. In its ultimate form the component will learn through interaction with other components as proposed by current research in agent technology and use adaptive mechanisms to evolve in its behavior [2][11]. We want to apply such mechanisms and behavior to the industrial automation system and propose the following architecture.

Proposed architecture implementing autonomic behavior: Let's assume the following use case. A component added to the network, for example a router, automatically registers itself to the directory service. Another service, NDS, discovers the router and informs PNSM about the new component. If PNSM already has a configuration available for that type of component, it is instantiated and made available for monitoring. If not, PNSM informs NDS, which starts retrieving further information from the router using SNMP. If it cannot resolve all OIDs, it either contacts already known web services to retrieve further MIBs or it searches for new web services that can provide MIBs. Once a suitable MIB has been located, NDS installs and tests it. PNSM on the other hand continues its monitoring of all configured and instantiated devices. If it detects values that do not respect a specified threshold it informs the diagnostic service, which starts collecting monitoring information from other components

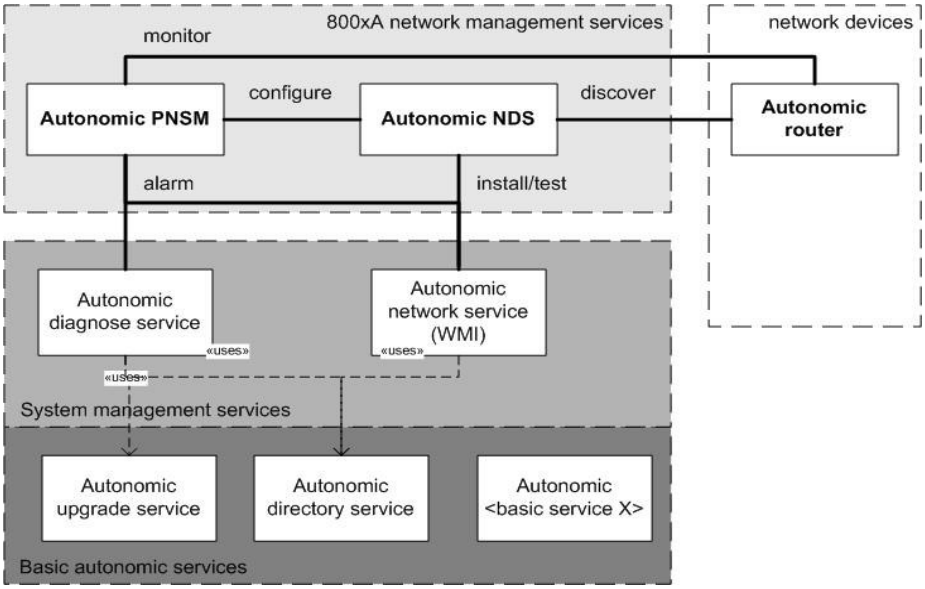

Figure 3: This figure shows different levels of autonomic services. All services register themselves at the directory service. PNSM and NDS services manage and monitor network devices such as a router using diagnosis and network services.

and from log files, and analyzes system-wide data. Each component monitors the availability of new upgrades. If a new upgrade is available the upgrade service downloads and tests the new version against the component's specifications. If a problem occurs, the component is reverted back to its old version.

In order to implement this type of architecture, the components will have to be categorized into classes of middleware components providing compositional adaptation. Basic services provided by each autonomic component will be part of the first middleware layer (Figure 3) whilst more specialized services such as a network or a diagnose service will be part of the next layer of system management. The NDS and PNSM component should be seen in an application-specific context such as network management. Let's look at the services discover, configure, and alarm of Figure 3 in more detail distinguishing the current implementation in 800xA from future more autonomous behavior we envision.

Alarm: Today we use the simple network management protocol (SNMP) to monitor devices on the network. SNMP OIDs are retrieved periodically in a fixed time interval through Windows Management Instrumentation (WMI) and delivered to the network management system PNSM. In a more autonomic setting the device would monitor itself and send an alarm to PNSM when a given threshold for the particular information item is reached. In architectural terms this would result in pushing the intelligence down to the device level. A more modest improvement would be to change the retrieval interval based on historical values of an information item. On small changes the interval would grow whilst a sudden change in value would result in shrinking the interval again. Such an adaptation algorithm would improve the additional bandwidth required by the monitoring function. Architecturally, a distributed algorithm, e.g. using control-theory concepts [12], could be used for implementation.

Discover: Currently a new device in the network is discovered by NDS scanning the entire network explicitly. A bunch of information items are sent in form of OIDs to NDS and proposed to the user for selection. A more autonomic behavior would be that a new device installed in the network sends registering information to a directory service (Figure 3). The

directory service would then register the device and its services and also publish its services to NDS and PNSM and perhaps to neighbor devices.

Configure: Currently, NDS presents a configuration string for each information item found, once the information could be identified through a MIB which needs to be compiled into WMI. This then allows the user to configure the specific information in PNSM by setting up a new device type for example. In the future, the NDS component could for example maintain rules on which Web services to contact for further MIBs and how to choose MIBs relevant for the device to be monitored. Based on higher-level policies typical information would be considered for each device type to be monitored. Based on a kind of type template a new device type could be defined in the library of PNSM which a set of standard information items and the corresponding configuration strings could be extracted from NDS.

The issues of how to identify suitable MIBS automatically or how to identify which information items should be selected form monitoring will be tackled in further research in this area. As new devices are added to the system, such policies might have to be updated and adapted to the new environment. The issue of decision making and self-learning becomes a major research area if the vision of autonomic computing is to be realized in all its consequences [10].

4. Conclusion

In this paper we discussed the complex problems of configuration and execution of network management of industrial automation systems, especially for monitoring purposes. We showed how we solve this nowadays with our process control system 800xA (including PNMS and NDS tools). We pointed out the advantages of combining process control and network management in the domain of industrial automation technology. Furthermore we suggested an architecture of self-managed autonomic components for network management of industrial automation systems.

References

[1] Horn, P., 2001. Autonomic Computing: IBM's Perspective on the state of Information Technology.

[2] Kephart, J. O., Chess, D.M., 2003. The Vision of Autonomic Computing. IEEE Computer. 41-50.

[3] Industrial IT System 800xA, System Architecture. 3BUS092080R0001. ABB Automation Technologies. http://www.abb.com - Products & Services – ABB Product Guide – 800xA.

[4] OPC. OPC Foundation, http://www.opcfoundation.org.

[5] Gelle, E., Koch, T.E., Sager, P., 2005. IT Asset Management of Industrial Automation Systems, Proceedings of 12th IEEE International Conference on ECBS, 123-128

[6] Policht, M., 2001. WMI Essentials for Automating Windows Management. Sams.

[7] Stallings, W., 1996, SNMP, SNMPv2 and RMON – Practical Network Management. Addison-Wesley, 2nd Ed..

[8] Seufert, F., 2003. Netzmangement der Zukunft, megalink, 27-29 (In German).

[9] Müller-Schloer, C., von der Malsburg, C., Würtz, R.P. 2004, Organic Computing. Informatik Spektrum, 332-336.

[10] McKinley, P.K., Sadjadi, S.M., Kasten, E.P., Cheng, B.H.C., 2004, Composing Adaptive Software. IEEE Computer. 56-64.

[11] Preiss, O., Naedele, M., 2002. Architectural Support for Reuse: A Case Study in Industrial Automation. In Building Reliable Component-Based Software Systems. Eds: Crnkovic, I., Larsson, M., Artech House Publishers.

[12] Diao, Y., Hellerstein, J.L., Parekh, S., 2005. Self-Manging Systems: A Control Theory Foundation, Engineering of Autonomic Systems (EASe), Proceedings of 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems ECBS, 441-448