A Process Data Acquisition Architecture for Distributed Industrial Networks

Florian Pethig and Oliver Niggemann

Fraunhofer IOSB-INA,
Application Center for Industrial Automation,
Fraunhofer-Institute of Optronics, System Technologies and Image Exploitation,
Lemgo, Germany
www.iosb-ina.fraunhofer.de
{florian.pethig,oliver.niggemann}@iosb-ina.fraunhofer.de

Abstract In this paper, a data acquisition solution for the monitoring of all sensor and actuator signals in a production plant is described. The solution handles the increasing degree of distribution of the automation system, the heterogeneity of the automation system, and the problem of a uniform and computer-interpretable access to the acquired data.

Introduction

Automation is a key factor in all production systems. Only an increasing level of automation can guarantee market-oriented product prices, can handle the increasing complexity of products, and can assure high product qualities and minimal resource consumptions.

So the monitoring and diagnosis of running production systems is one of the most important tasks of automation systems. E.g. the European Factories of the Future Roadmap defines the following goal for the field of automation: "Intelligent Measuring Systems for Zero-Defect Manufacturing. Development of fast and reconfigurable low/medium and high resolution measuring systems for accurate and time efficient measurements. Distributed Intelligent Measuring Systems reconfigurable both in space and time" [1].

In order to implement such a monitoring and diagnosis solution, as a first step, the current plant status must be known at each point in time. This paper describes such a data acquisition solution.

Currently, several problems impede such a data acquisition solution:

Non-local Error Causes: In most complex systems, effects of errors are not confined to the neighborhood of the error cause; instead, error effects are propagated throughout the whole system. E.g. a timing error in one plant module may cause buffer overflows at totally different parts of the plant. So any data acquisition solution must capture the status of the whole system, i.e. local data acquisition solutions are not sufficient.

Heterogeneous Automation Systems: Any system-wide data acquisition solution faces the heterogeneity of current automation systems: Different plant

modules use different control devices, different automation network protocols, and offer different information to the user. Any data acquisition solution must therefore cope with this heterogeneity.
Non-Synchronicity: Sensor, actuator, and controllers in automation system do not have a common time base. So current data acquisition solutions can often not identify the system status at a specific point in time.
Proprietary Access to the Data: The access to the acquired data is often implemented in a proprietary way. E.g. proprietary web interfaces are used. Furthermore, the interpretation of the data often requires additional information about the plant which prevents an automatic data analysis.

In the following, a solution is presented which addresses all these problems.

1 Architecture

The architecture shown in Figure 1 is intended for distributed industrial networks and adresses the problem of non-local error causes by placing distributed probes (called dataloggers) in all industrial ethernet segments. Here, dataloggers for different industrial network protocols allow for the data acquisition in heterogeneous automation systems. Additionally, the synchronization of these probes via IEEE 1588 [2] provides the required common time base. Details of these solutions for system-wide acquisition and synchronization can be found in section 2 of this paper.

The distributed probes send their acquired data to a central data server based on the OPC Unified Architecture (UA) [3] via an UDP/IP interface. OPC UA enables a standardized way for data access and information modeling, as a first step implemented by importing variable names from IEC 61131 engineering tools. The details of this solution are described in section 3.

OPC UA also offers possibilities for data access via WebServices (using SOAP and HTTP) and the platform independent development of clients [3]. This is used for mobile data access over the internet in section 4.

2 System-wide Acquisition and Synchronization

2.1 Heterogeneity

The heterogeneous nature of todays automation systems implies great challenges regarding process data acquisition: Data logging currently means acquiring process data at a central point, e.g. the programmable logic controller (PLC). Here often solutions such as OPC DA or TCP/IP function blocks are used. These solutions are not sufficient for demanding applications such as machine diagnosis, e.g. condition monitoring of distributed systems in a factory. This is why a system-wide acquisition has to be composed of distributed, synchronized probes, able to capture a variety of protocols.

Switched industrial ethernet system lead to one severe data acquisition problem: Since the purpose of a switch is to forward frames only to the appropriate

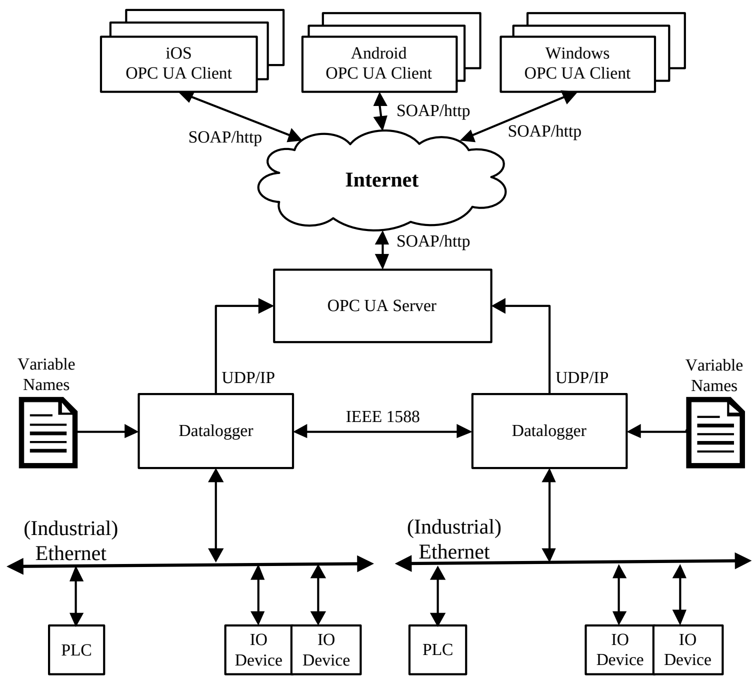

Figure 1. Architecture Overview.

port, it is not possible to capture all network traffic at only one point of the distributed system. Also because of the heterogeneous nature of todays automation systems, a modular architecture has to be used to log all process variables. This way data can be captured at different geographic locations, in different network segments, using different protocols and nevertheless all data can be aggregated in one location.

Figure 2 shows an example for a typical approach used in industrial manufacturing: A shop floor consists of different production cells, which again consist of different production modules (PM). Typically a highly heterogeneous structure of industrial network technologies, i.e. fieldbus and industrial Ethernet systems, exists in such environments. Machinery from different manufacturers uses different industrial networks (e.g. EtherCAT, PROFINET, Ethernet/IP, etc.) and often also the inter process communication between production cells is realized using different protocols (e.g. PROFINET, MODBUS/TCP, etc.).

Generally, every process variable of the shop floor is accessible when dataloggers are used inside of every production cell. But since no common time base is used between the different production cells this approach is not yet suitable for the analysis of distributed industrial production systems.

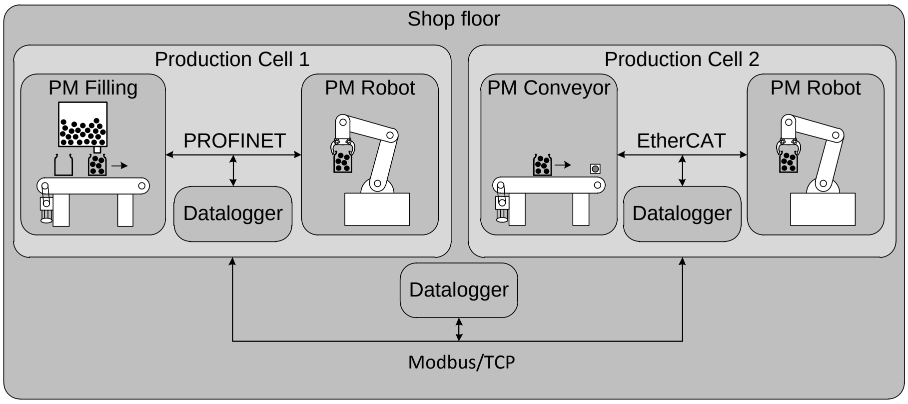

Figure 2. Shop floor communication example.

2.2 Synchronicity

As already mentioned in the introduction, to analyze a distributed system all captured data has to be synchronized. The synchronicity of all captured data is important e.g. for machine diagnosis applications such as anomaly detection or condition monitoring [4].

Regarding the cycle time, Real Time Ethernet (RTE) systems nowadays immerge into sub-millisecond ranges, which is the reason why past process data logging using e.g. OPC DA or the utilization of TCP/IP is not practicable anymore. Fast processes, e.g. the synchronization of axes in a motion application, require a synchronization accuracy < 1ms and typically a jitter $< 1\mu$s [5]. To achieve this accuracy reliably, the Precision Time Protocol (PTP) defined in IEEE 1588 is used here [6].

Figure 3 shows the architecture of a datalogger meeting these challenges by combining state-of-the-art real time ethernet hardware in a PC.

With regard to capturing traffic in industrial ethernet networks, a network tap should be used to enable passive capturing. This way, existing network configurations remain unchanged and the capture hardware does not affect the monitored network. Additionally, to achieve highly accurate and synchronized network traffic captures, a timestamping unit realized in hardware [2] which supports timestamping of Ethernet frames with an accuracy $< 1\mu$s [7] is used. A ready-to-use solution combining these requirements is the netAnalyzer PCI-Card from the company Hilscher. An Application Programming Interface (API) for the netAnalyzer allows the processing of captured frames and is used by the developed datalogger software component to gain access to the network traffic. To add the time synchronization via IEEE1588/PTP, a Software Stack for the Windows OS can be used to synchronize several datalogger PCs with the needed accuracy. The Software Stack used here was developed by the company Real-Time-Systems GmbH [8]. The software stack enables the synchronization

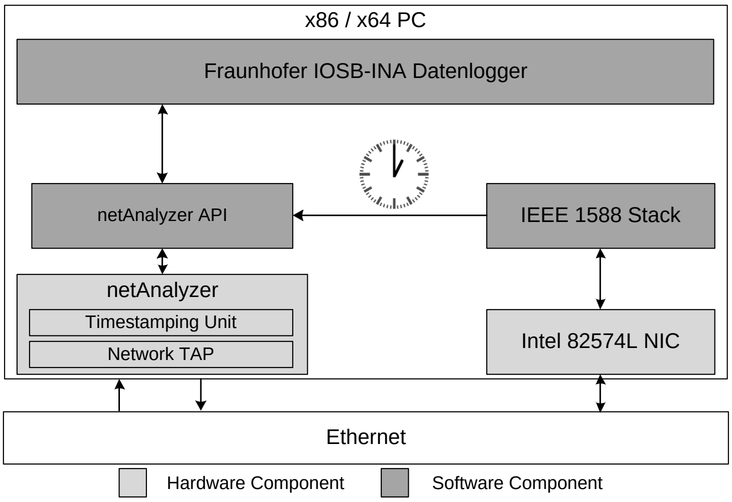

Figure 3. Datalogger architecture.

of distributed network interface cards (NICs) based on the Intel 82574L chipset and can also be accessed via an API.

By combining the described components to the architecture depicted in Figure 3, the datalogger is able to capture ethernet frames with a very high accuracy in synchronization.

But one questions remains: At which bit position can the process variables be found in the frames?

3 Semantics and Information Modeling

As described in section 2, state-of-the-art hardware components and methodologies (e.g. PTP) for capturing ethernet traffic and time synchronization allow for a very high accuracy and are suitable for even highly demanding applications. But capturing ethernet traffic is not equal to process data acquisition. To get process variables from ethernet frames, the variables' bit position in the appropriate frames must be known, i.e. both the protocol used and the actual network configuration have to be known. Furthermore, the variables' names should be extracted from the engineering tools.

As a first step, the names of process variables can be imported into the Fraunhofer IOSB-INA Datalogger software component. As a proof of concept this has been implemented for the engineering tool PC WORX from the company Phoenix Contact. E.g. the bus configuration of a PROFINET (PN) network and the process variables used in the IEC 61131-3 control program are imported into the datalogger software. Therefore the bus configuration has to be exported out of the engineering tool PC WORX and the resulting CSV-File including variables and their connection with PN-Devices and I/O modules is parsed.

Furthermore, bit positions (offsets) and data types are used to determine the position and length of specific variables in specific PN-Frames. PN itself handles IO data by exchanging IODataObjects between PN devices. For these objects, PN defines further offsets which are considered by analyzing the PN Configuration determined during the start-up phase of the PN system. These information combined can be used to monitor PN variables with the high accuracy described in section 2. An overview of the needed information to capture PN variables is shown in Figure 4.

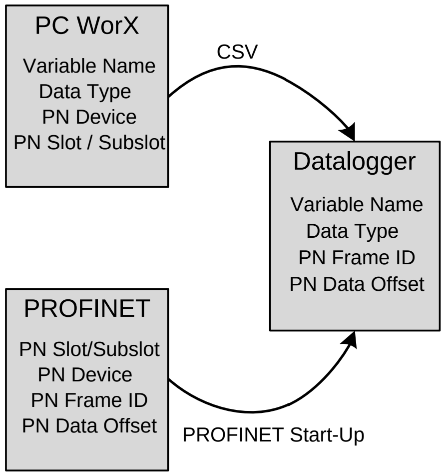

Figure 4. Import variable configuration.

Todays problem of proprietary data access, already mentioned in the introduction, can be avoided by using the OPC Unified Architecture [3].

In contrast to the older OPC DA standard OPC UA does not limit access only to Windows-based clients. OPC UA is based on WebServices communicating via SOAP/HTTP, which allows access to monitored values from various platforms and over the internet—since the firewall-friendly port 80 is used [3]. To secure connections, especially if SOAP deployed over TCP/IP and the internet is used, the UA also includes security mechanisms such as signing messages or complete sessions [3].

OPC UA also includes an explicit meta model which can be parsed at runtime. Using this reflection API, applications can identify and interprete data at runtime, e.g. for a signal verification, for diagnosis, or for system optimization tasks.

In the context of data access, future work will include a complete information model additionally describing location and/or characteristics of exchanged process variables. E.g. for a process variable "P-L1" it can be described that this variable is an energy measurement captured with a specific sensor, at a specific location, with a certain amount of active consumers and their specific characteristics. This way computer systems will automatically be able to interpret monitored values. I.e. they can then be used for anomaly detection and behavior models for production plants [9].

4 Data Access from Mobile Devices

As already stated in the introduction, proprietary data acquisition solutions often require matching, also proprietary, visualization and human interface applications, e.g. mobile applications. In connection with the platform-independent communication standard OPC UA, new solutions for the visualization of process data on mobile devices are possible. Such a solution provides the ability to access data at any place and at any time without the need for operating terminals at fixed locations in a factory. Scenarios where mobile data access would be needed include:

- **Maintenance:** Technicians can access all necessary data at a glance anytime from anywhere.
- **Location based visualization:** A factory manager can access real-time data for process cells/modules based on his current geographical location in the plant (e.g. using a Wifi positioning system or QR-Tags).
- **Enterprise Resource Planing:** The Management can base decisions on mobile data access.

Of course also certain requirements exist when thinking of mobile data access solutions:

- **Data security:** Access to plant data must only be granted to authorized personnel.
- **Data aggregation:** Data to be handled on mobile devices should be preprocessed (i.e. aggregated) to minimize data processing to be done on the device and to assure an appropriate look and feel of the mobile application.

Figure 5 and 6 show energy measurements from a plant accessed using Android and iOS devices.

5 Conclusion

In this paper, a methodology and a prototype for capturing sensor and actuator signals in distributed automation systems are described. One focus is the synchronization of different capturing devices via the IEEE 1588 protocol. This solution also comprises an OPC UA based standardized access to the data which can also be used to identify a signals semantics at runtime. As an application, a mobile visualization application is described.

Figure 5. Process Data Access from an Android device.

Figure 6. Process Data Access from an iPad.

References

1. A. hoc Industrial Advisory Group for MANUFUTURE-EU, "Factories of the future ppp strategic multi-annual roadmap," European Commission, Tech. Rep., 2010.
2. IEC/IEEE, "Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEC 61588 First edition 2004-09; IEEE 1588*, 2004.
3. "OPC Unified Architecture," 2009, DIN EN 62541.

4. A. Maier, A. Vodencarevic, and O. Niggemann, "Anomaly detection in production plants using timed automata," in *8th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2011.
5. M. Felser, "Real-time ethernet - industry prospective," *Proceedings of the IEEE*, vol. 93, 2005.
6. B. Neagoe, Cristea, "Ntp versus ptp in computer networks clock synchronization," *IEEE ISIE*, 2006.
7. Hilscher, "netAnalyzer," Website, 2011, www.de.hilscher.com.
8. Real-Time-Systems GmbH, "RTS IEEE 1588 Stack for Windows OS," Website, 2010, www.real-time-systems.com/ieee_1588.
9. A. Vodenčarević, H. Kleine Büning, O. Niggemann, and A. Maier, "Identifying behavior models for process plants," in *16th IEEE International Conference on Emerging Technologies and Factory Automation ETFA'2011*, Toulouse, France, September 2011.