Foreword

The pace of change in industrial control and automation technology is accelerating. Demands on machines and systems concerning flexible retrofit, production speed, and fail safety are increasing, as are cost pressures. Software has more and more become the essential factor with products, systems, and complete plants.

At the same time, changes in the field of automation brought by the use of the PC as an automation component, by the Internet, and by the tendency to more open standards, can be clearly seen – to the benefit of the user and the manufacturer.

The PC is used today for visualization, data acquisition, process control, and the solution of further tasks in automation. It complements or replaces the traditional PLC and the operator terminal. The reasons for this are the continual decrease in price of the mass-produced PC, the permanent multiplication of the computing capacity of the CPU, the availability of even more efficient and comfortable software components, and the ease of integration with Office products.

Efficiency and cost savings are achieved through the reuse of software components and the flexible compilation of such components into distributed automation solutions.

Horizontal integration of the automation solutions through communication between the distributed components also plays an important role. Immense additional savings are achieved by means of vertical integration by optimizing the process of product planning, development, manufacturing and sales. This optimization is realized through a consistent data flow, permanent data consistency, and the availability of data on the field level, control level, and office level.

The use of standardized interfaces by several manufacturers is a prerequisite for the flexible compilation and integration of software components.

OPC has been generally accepted since many years as the most popular industrial standard among users and also among developers. Most of the HMI (Human Machine Interface), SCADA (Supervisory Control and Data Acquisition), and DCS (Distributed Control System) manufacturers in the field of PC-based automation technology, as well as the manufacturers of PC based Control and MES (Manufacturing Excecution Systems) applications are offering OPC client and/or OPC server interfaces with their products. The same is true for suppliers of devices and interface cards. In the last years OPC servers widely replaced DDE (Dynamic Data Exchange) servers and product-specific drivers in this field.

Today, OPC is the standard interface for access to Windows-based applications in automation technology. Over 20,000 OPC products by several thousands manufacturers are in use worldwide in a vast variety of industries. The number of installations has reached many millions and covers a wide range of application areas. The OPC interface is today not only used for simply transmitting process data or individual parameters: Entire ERP documents, parameter sets, control sequences, video signals or drive programs are transferred via OPC. Within only a few years, OPC has conquered the industry. Whether in the production industry, in process technology or in building automation, OPC is the standard! This fact is substantiated by a study with which the OPC Foundation commissioned the ARC market research company in 2003. The result of the ARC study states that already in 2003 84% of all participants in the study are very familiar with OPC technology. The study adds that in 78% of the production systems and MES applications, 75% of the HMI/SCADA systems, 68% of the process supervisory control systems and PLCs, and 53% of the ERP systems, OPC is the preferred method for data exchange and connection to the process. The market research company expected this trend to be continued at an even greater pace over the next five years, with OPC extending to subsystems, small automation components, and even down to the sensor level.

This forecast from 2003 was confirmed by a large-scale market survey conducted by Softing AG in cooperation with the OPC Foundation in 2008. The questionnaire was sent to over 70,000 addresses around the world and contained questions about the importance of OPC for business, the scope of use of OPC products in company-specific projects, the strengths and weaknesses of OPC, and much more.

Over 3,500 people answered the 15 questions of the market survey. 20 percent of them were end users, another 20 percent system integrators, and the remaining 60 percent were software or hardware vendors. Those who participated in the survey were almost evenly spread over the oil and gas, power generation, chemical, factory automation, building automation and automotive industries and many more.

60 percent of the participants used OPC frequently or always, only 2 percent did not know OPC. Two thirds used OPC for visualization, followed by controlling and data archiving. About half of the participants used OPC for internal data exchange, i.e. for communication between applications of the same vendor. 43 percent, on the other hand, used OPC for accessing external data in applications from other vendors.

When asked what technology would be used in data exchange in the future, 1,100 of the 3,500 participants answered they would mainly use OPC. Only about 320 said that mainly proprietary drivers or other technologies would be used for data exchange.

While nearly four fifths regarded OPC as a useful technology, half of the participants thought that OPC could be improved. Only five percent considered OPC insufficient or faulty.

45 percent regarded the OPC Unified Architecture either as useful or as better than Classic[1] OPC or as the ideal technology, whereas 42 percent were not yet familiar with OPC UA. 60 percent believed that the importance of OPC for their future business would increase, while only six percent thought that OPC would be less important in future.

Where and how is OPC used today?

OPC specifications define an interface between clients and servers as well as servers and servers for different fields of application – access to real time data, monitoring of events, access to historical data, and others. Just as any modern PC can send a print task to any printer, thanks to the integration of printer drivers, software applications can have access to devices of different manufacturers without having to deal with the distinct device specifications. OPC clients and servers can be combined and linked like building blocks by the use of OPC technology.

At present, OPC clients and servers are mainly available on PC systems with Windows 2000/XP/Vista and x86 processors.

Apart from a few exceptions, the OPC products on the market today are based exclusively on the Distributed Component Object Model (DCOM) – the Microsoft technology for implementing distributed systems. Many Windows CE versions support DCOM, so that there are many different solutions (operator panels, handhelds, etc.) with OPC Client or Server interfaces.

With the increasing acceptance and popularity of Web Services in IT and industry, the OPC Foundation started developing the XML-DA Specification aimed at introducing completely new concepts to OPC. The objective was to maintain investments in existing OPC developments while opening up whole new application possibilities for OPC in the future. The specification defines data exchange via Web Services, using the eXtended Markup Language (XML) as the description language. This approach of the OPC Foundation is based on proven international standards, W3C (World Wide Web Consortium) standards that allow platform neutral application and exchanging data over the Internet across firewalls. The possibility to map not only pure process data, but also alarms, historical data and commands to Web Services led to a completely new approach, the OPC Unified Architecture Specification. It defines the management of process data, alarms, historical data and programs in a single, unified address space. In addition to

1 With the introduction of the new technology generation OPC Unified Architecture the term “Classic OPC“ has been created to distinguish the sum of the “old” DCOM based OPC specifications including XML-DA from the new OPC UA specification.

these new properties – platform independence and unified data management – OPC UA introduces numerous other properties, including scalability, security regarding unauthorized access, precautions against data loss, and the support of complex data structures and information models from other organizations. With OPC UA, the OPC Foundation pursues the vision of global interoperability, i.e. of a standardized data exchange between software applications independently of the manufacturer they come from, the programming language they were developed in, the operating system they run on, or the place they are located. OPC UA introduces a new generation of technology, which has the potential to extend far beyond today's range of OPC applications and become established in areas such as embedded systems, IT, etc.

Why is OPC so successful?

When creating OPC specifications, the main focus has been and still is the practically feasible and not the theoretically desirable. The main center of attention is the 60% of functionality required by 80% of users. The approach of the OPC Foundation has always been to avoid unnecessarily detailed discussions and political disputes and to create practical facts within a very short time. The Foundation therefore succeeds again and again to build additional requirements into the specifications, to rapidly develop new OPC specifications, and to successfully implement them in practice.

The book on hand is addressed to all those interested in OPC and who want to know more about it – users of OPC products, decision-makers, product managers, project managers, and software developers. This completely revised and extended fourth edition provides for "beginners" and experienced OPC users alike the opportunity to become familiar with the current status of the OPC specifications including the new OPC Unified Architecture (UA) and various collaborations of the OPC Foundation with different other standardizations organizations.

The reader should have a general understanding of the requirements of modern automation technology. Basic knowledge of programming with object-oriented languages would be an advantage. Technical terms from the fields of automation technology and software development, which may be less common, are explained the first time they appear in the book.

In chapter 1 of this book, the history of OPC is described. Technological development – from specific drivers limited to one product, via the first standardization efforts in the form of DDE, up to the OPC technology – is described. The advantages and disadvantages to the development and use of product specific drivers, DDE servers, and OPC servers are examined. A general overview of the basic DCOM, .NET and Web Service technology and the principal functions of OPC clients and OPC servers are also included in this chapter. The chapter finishes with an insight into the structure and operation of the OPC Foundation and an overview on several collaborations of the OPC Foundation with other organizations.

The first part of chapter 2 explains the Classic DCOM based OPC specifications Data Access, Alarms and Events, Historical Data Access and Commands with their "companion" specifications Common Definitions and Interfaces, Batch, Security and Complex Data and the Web Service based specifications OPC XML-DA and OPC DX. For every specification, the field of application and the model this specification is based on are presented. Afterwards, the contents and use of the specification are described using specific scenarios.

The second part introduces the OPC Unified Architecture multi-part specification providing an overview on the 13 specification parts, the most important technical characteristics and the main features of OPC UA. The OPC UA introduction will be completed with a description of some OPC UA "companion" standards which are the result of collaborations of several standardization organizations with the OPC Foundation. In a third part, chapter two explains the Compliance Test for Classic and OPC UA products.

Chapter 3 gives answers to questions arising from design and implementation of OPC components. Also this chapter distinguishes between Classic OPC and OPC UA. In a first part the general aspects of the use of DCOM and implementation with DCOM is explained. The second section describes how to create OPC UA components for Windows, Linux and VxWorks. It distinguishes between the OPC UA Stack based implementation in C/C++, .NET and Java and the implementation using toolkits. The third section discusses the combination of Classic OPC, XML-DA and OPC UA in a hybrid client or server component.

The end users of the OPC technology are the main focus of chapter 4. It describes the use of OPC in automation products of different manufacturers. Key aspects of using Classic OPC products, such as configuring DCOM settings, OPC Tunneling or direct server-server and client-client communication are discussed. An article from Beckhoff describes how OPC UA can be embedded directly in a PLC. A tool for designing and modeling a UA AddressSpace is introduced in the article from CAS. SAP illustrates the significance of the OPC Unified Architecture for their future product concepts. Chapter 4 finishes with performance analyses for OPC and OPC UA.

Chapter 5 contains a summary and an outlook on the future of OPC.