2009

IT Infrastructures in Manufacturing: Insights from Seven Case Studies

Oliver Günther
Humboldt-Universität zu Berlin, guenther@wiwi.hu-berlin.de

Lenka Ivantysynova
Humboldt-University, Berlin, lenka@wiwi.hu-berlin.de

Jochen Rode
SAP Research, jochen.rode@sap.com

Holger Ziekow
Humboldt-University, Berlin, ziekow@icsi.berkeley.edu

IT Infrastructures in Manufacturing: Insights from Seven Case Studies

Oliver Günther
Humboldt-Universität zu Berlin
guenther@wiwi.hu-berlin.de

Lenka Ivantysynova
Humboldt-Universität zu Berlin
lenka@wiwi.hu-berlin.de

Jochen Rode
SAP Research CEC Dresden
jochen.rode@sap.com

Holger Ziekow
ICSI / Humboldt-Universität zu Berlin
ziekow@icsi.berkeley.edu

ABSTRACT

IT solutions in manufacturing support the execution as well as the monitoring of production operations. Fast reaction to exceptions, detailed documentation of operations, and the detection of inefficiencies in the production are among the benefits of a tight IT integration of shop-floor processes. Several dedicated software solutions and standards exist for the manufacturing domain. However, each manufacturer must tailor the IT to the special requirements of its processes and infrastructure. We found that real-world installations show considerable variations. In this paper we present the results of seven case studies on IT infrastructures in manufacturing. For each case we portray the employed architecture and the main factor that influenced the design. From this analysis we derive reoccurring patterns on the structure of IT solutions in manufacturing and relate them to existing standards. Our results provide system architects with guidance for picking the right architectural choices in different manufacturing environments.

Keywords

Production Data Acquisition, Manufacturing Execution Systems, Architecture

INTRODUCTION

IT solutions in manufacturing support the execution as well as the monitoring of production operations. Fast reaction to exceptions, detailed documentation of operations, and the detection of inefficiencies in the production are among the benefits of a tight IT integration of shop-floor processes. Manufacturers face the challenge of tailoring the IT to the needs of their particular processes and infrastructure. This paper targets the architectural design of IT solutions in manufacturing. We present findings from case studies at seven manufacturers. On this basis, we identify reoccurring patterns in IT requirements as well as in corresponding IT architectures. Finally, we derive guidance that supports system designers in architectural choices for the manufacturing domain.

Typically the flow of information in manufacturing IT solutions can be described as follows: sensors on machines (counters, temperature, pressure, flow…) generate raw data that are collected through Supervisory Control And Data Acquisition (SCADA) systems or Distributed Control Systems (DCS). The tasks of the SCADA/DCS are monitoring and controlling the processes and providing human machine interfaces (HMI). Frequently, large temporal databases – so-called historians – store raw data for later analysis (tracking and tracing, callback management, etc.). In many cases, raw data are passed on to a system for production data collection (PDC). PDC functionality is often part of a Manufacturing Execution System (MES) which controls the overall process flow, including technical and logistical aspects (Vollmann 2005). MES typically have an operational horizon ranging from seconds to a few days. Aggregated data (production information, component consumptions, yield/scrap rates, etc.) are periodically communicated to Enterprise Resource Planning (ERP) systems, which manage the long-term production planning and logistics.

Software systems in manufacturing commonly show a hierarchical structure. Figure 1 provides a simplified overview. The depicted pyramid shape reflects the data volumes and granularity on each level in the system hierarchy. Top down the production plans are consecutively refined and mapped to production resources. The lowest level captures production data and passes them on to higher system levels. Each level processes data and aggregates them for reporting to the next higher system level.

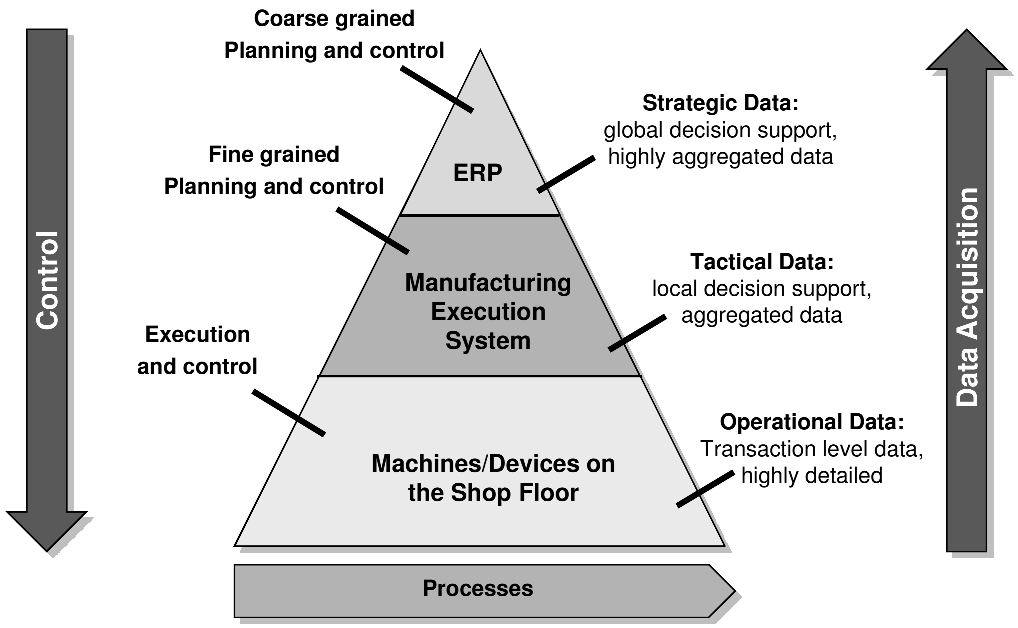

Figure 1: Simplified structure of software systems in manufacturing

Several reference models and standards have been defined for the flow of information in manufacturing control systems. The Purdue reference model (PRM) divides the manufacturing domain into six levels of computing functionality, which are based on the hierarchy of the manufacturing enterprise (Williams 1992).

The second – and most prominent - standard is the ISA-S95 family (Brandl ed. 2005). This is a standard family for plant-to-business (P2B) integration with a special focus on manufacturing. The ISA-S95 standard builds on the PRM and defines a multi-level functional reference model for manufacturing systems. This standard defines a terminology for structuring MES and illustrates how the functionality within a manufacturing system is distributed across four different levels.

Besides the ISA-S95 Plant-to-Business integration (P2B) standard, there are two other P2B integration standards: RosettaNet (RosettaNet 2008), and OAGIS (OAGIS 2007). In contrast to these standards, ISA-S95 focuses solely on the integration of ERP and MES. Both RosettaNet and OAGIS go far beyond P2B integration in an attempt to model every class of B2B (Business-to-Business) transaction.

Software standards for manufacturing specify functional building blocks but do not state on which hardware platforms they should run. However, our studies show that the physical deployment is a critical architectural decision. Like software systems, the IT hardware at manufacturers is usually hierarchically structured. For discussion in this paper we distinguish four hardware tiers: the device tier, the edge tier, the middle tier, and the back-end tier (see Figure 2). We define the different hardware tiers and their relation to ISA-S95 levels as follows:

The *device tier* comprises devices on the plant floor, such as robots, machines, RFID readers, etc. Software running on those devices is, for instance, for user interfaces on the machines, PLC software, or distributed control systems. This can be mapped to ISA-S95 level 1 and 2.

The *edge tier* comprises computers on the plant floor. Often these are PCs that host OPC servers (OPC Foundation 2007) and diverse clients like MES or PDC client. The ISA-S95 standard provides no separate level for these functionalities but includes them in level 3.

The *middle tier* comprises servers and PCs that physically reside in the plant. These machines typically host MES or PDC systems and related software components. The software components can be, for example, programs for visualization of the shop floor, logging programs, or utility maintenance software. Such functionality belongs to level 3 in the ISA-S95 standard.

The *back-end tier* comprises servers that run systems with company-wide scope. Typically these are the ERP system as well as data warehouse and business intelligence solutions. Servers in the back-end tier often run in remote data centers and are accessed via a wide area network. The back-end tier hosts software that matches to level 4 of the ISA-S95 standard.

Note that the above sketched association of ISA-S95 and hardware tiers is idealized. Our case studies show significant deviations in the software deployment of the investigated companies. These deviations reflect the adoption of the IT system to the specific environment of each manufacturer. We discuss the architectural choices for each case study in Section 2. Section 3 presents lessons learned from the studies and Section 4 concludes the paper.

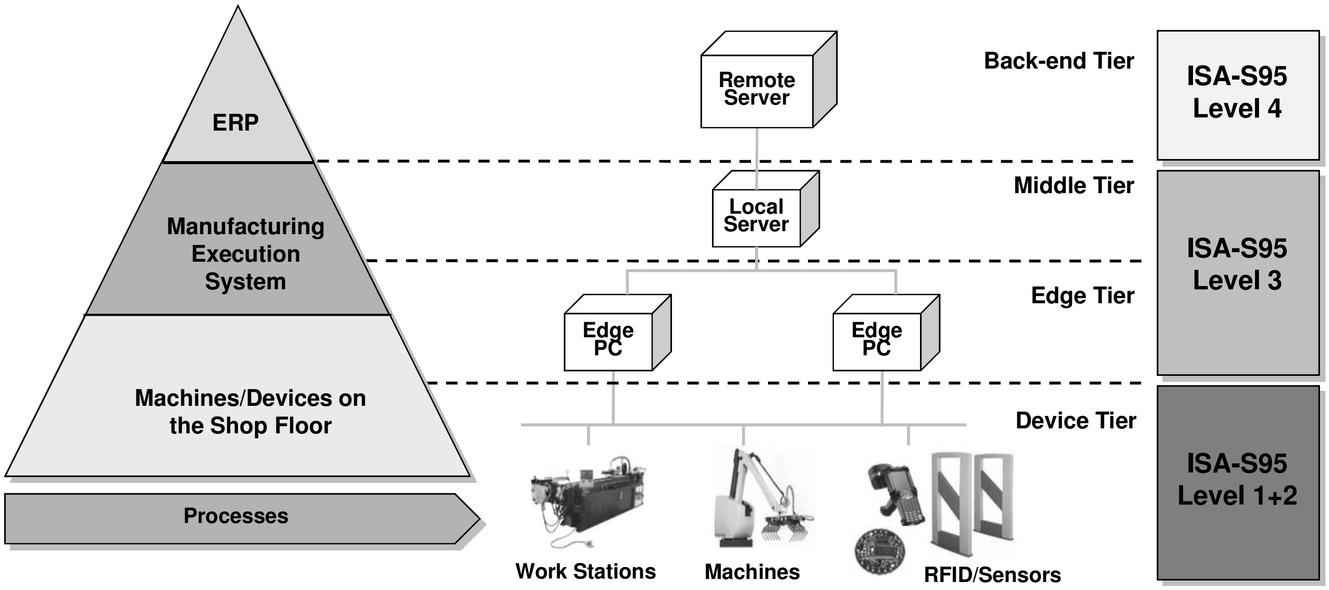

Figure 2: Common hardware tiers in manufacturing

CASE STUDIES

In order to examine the current use of IT, we conducted seven case studies at manufacturers from diverse industries. As is true for any case-based analysis, we cannot claim that our insights are representative. However, we found repeating structures in requirements and corresponding architectural solutions within our sample. We are therefore confident that our insights can provide helpful guidance for the architectural design of IT solutions in a broad range of manufacturing plants.

We conducted the case studies in the period from August 2007 to August 2008. The participating companies are from the following industries (names are not revealed due to non-disclosure agreements):

- batch production: manufacturer of milk products (short MIP),
- discrete production: manufacturer of engine cooling modules (COO),
- discrete production: manufacturer of refractories (REF),
- discrete production: manufacturer of engines (ENG),
- batch production: manufacturer of chemicals (CHE),
- discrete production: manufacturer of power plants (POW),
- discrete production: manufacturer of tires (TIR).

All seven companies are headquartered in Germany. Their sizes range from several hundred to over 100,000 employees worldwide. Four out of the seven companies are listed in the DAX or MDAX. In structured interviews, we questioned the IT staff of each manufacturer about system requirements. We focused on support for *Openness/Adaptiveness*, *Lightweight design*, *Reliability*, *Fast response times*, *Security*, *Event driven communication*, and *Scalability*. *Openness/Adaptiveness* refers to system support for adapting the configuration and for extending the functionality. *Lightweight design* refers to a small system footprint and a simple structure. *Reliability* refers to guaranteed availability of the system. *Fast response times* denotes high reactivity of the system. *Security* refers to protection against malicious attacks. *Event driven communication* refers to support for push-based interaction, and *Scalability* refers to the system's ability to handle increasing workload. In open, interviews we discussed IT architecture at each plant and how the requirements were addressed.

Case MIP

In the following, we discuss the IT infrastructure at the manufacturer of milk products (short MIP). The investigated departments are highly automated. Manual intervention is limited to configuring machine settings, loading/unloading the machines, and taking samples for quality checks. We interviewed the IT staff at the plant about requirements that are particularly relevant for their IT system. The following properties of the system were perceived as most important:

- Openness/Adaptiveness
- Reliability
- Fast response times
- Scalability
- Support for event driven communication

Error! Reference source not found. visualizes the system deployment across the four hardware tiers. MIP is currently using the SAP R/3 ERP system. Communication from the ERP system to lower system layers is enabled via SAP XI. Data exchanged between the ERP system and SAP XI is realized using IDocs. The communication between SAP XI and the PDC system is done by means of XML documents which are transmitted via HTTP. A part of the PDC system solution is an Order Monitor. This component is used to manage production tasks at specific process steps. From the Order Monitor the production tasks are communicated to the PLC of the respective Machine. For communicating machine events to the ERP system, the PDC system creates notifications that are passed through a Notification Monitor to the SAP XI. The SAP XI then passes the data on to the ERP via RFC calls. In addition, some reporting is done manually using MS Excel and Word.

On the device tier, MIP mainly uses Siemens S5 and S7 as PLC on the machines. The PLCs are connected with PCs that host OPC servers. These servers in turn are linked to the PDC system. Data from machine sensors, machine settings, and production tasks as well as programs for machine control are communicated along this connection.

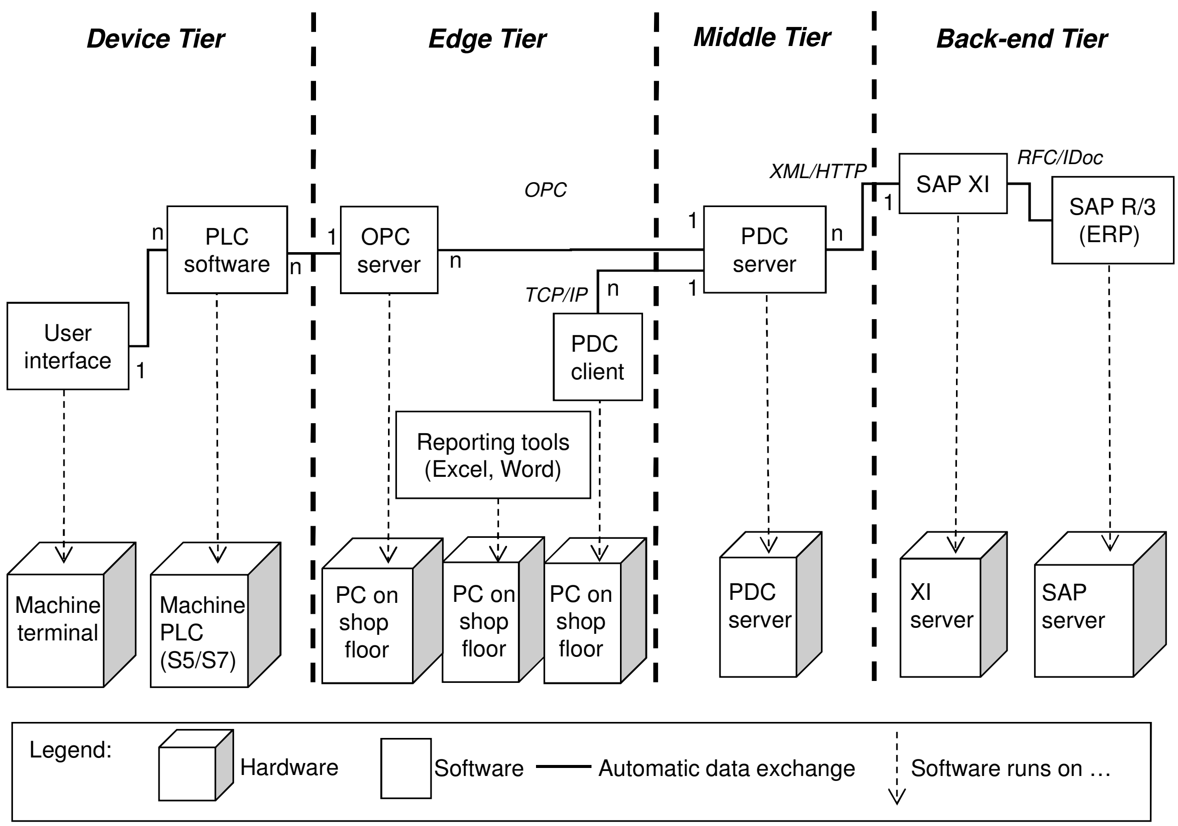

Figure 3. Deployment diagram of the software and hardware at MIP.

Note that MIP deploys functionality on all four hardware tiers in a hierarchical structure. Software on each level processes data as it traverses the system hierarchy. With this architectural choice MIP achieves high performance and scalability. This is further supported by using event driven communication for reporting alerts and status changes. MIP's demand for system reliability is not directly supported by architecture. The PDC poses a single point of failure. However, redundant clusters for hosting the PDC overcome this problem.

Case COO

COO is a manufacturer of engine coolers with several plants worldwide. The IT solutions at COO's plants are not unified. COO aims to change this situation by employing a PDC system in each plant. The foremost required properties of the system are listed below:

- Openness/Adaptiveness
- Support for event driven communication
- Lightweight and simple

Note that the current system design is denominated by the need for a light weight and simple solution. Today every plant of COO handles data acquisition differently. However, some similarities can be identified throughout the isolated solutions. We outline these similarities in **Error! Reference source not found.**, which gives an overview of the IT system.

COO does not employ a dedicated system for production data acquisition. Each plant captures, logs, and backs up data from the device tier locally. However, the top management needs aggregated reports from the plant floor to support strategic decisions. Currently COO employs two solutions for reporting. Workers at some plants manually create and email reports to the headquarters. Other plants use a self implemented data base tool for entering reports in a central MS Access database. Employees at the headquarters use the mailed reports and the data from the MS Access database for creating MS PowerPoint presentations for the top management.

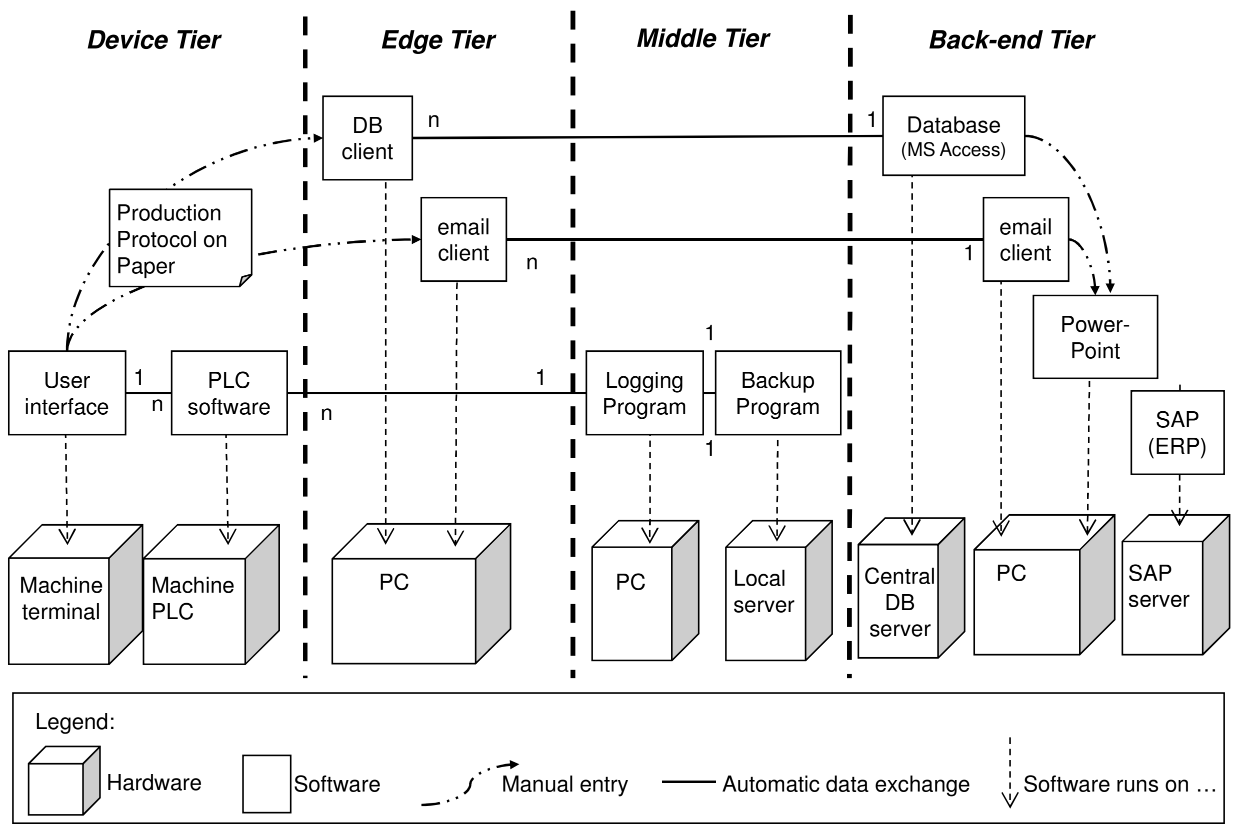

Figure 4. Deployment diagram of the software and hardware at COO.

COO designed the current solution primary with regards to simplicity. However, the lack of direct integration of shop-floor systems with the ERP back-end accounts for inaccuracies in the data management and poses limits to analysis of shop-floor operations. These are the primary reasons why COO pursues the introduction of a dedicated PDC system.

Case REF

REF is a globally operating company producing refractories (i.e. bricks for ovens) for diverse industrial sectors. The most relevant requirements for the IT solution are:

- Openness/Adaptiveness
- Lightweight and simple

REF requires foremost a simple, *lightweight* IT solution. This is because only limited resources are available in the IT department. The current IT architecture at RFF along the flow of production data is depicted in **Error! Reference source not found.**.

REF does not employ a dedicated system for production data collection. Instead, data are recoded manually on paper. At the end of each shift, data from these paper documents are manually entered into several purpose build applications. These are (1) application for recording quality checks, (2) a software for managing the utilization of kilns for processing refectories, (3) an application for tracking the utilization of cast form, and (4) the salary control system FoxPro®. In addition, data are entered into a self-implemented application in MS Excel. REF uses this Excel solution for fine-grained planning and control of production tasks. Integration with the ERP system is realized manually via SAP clients.

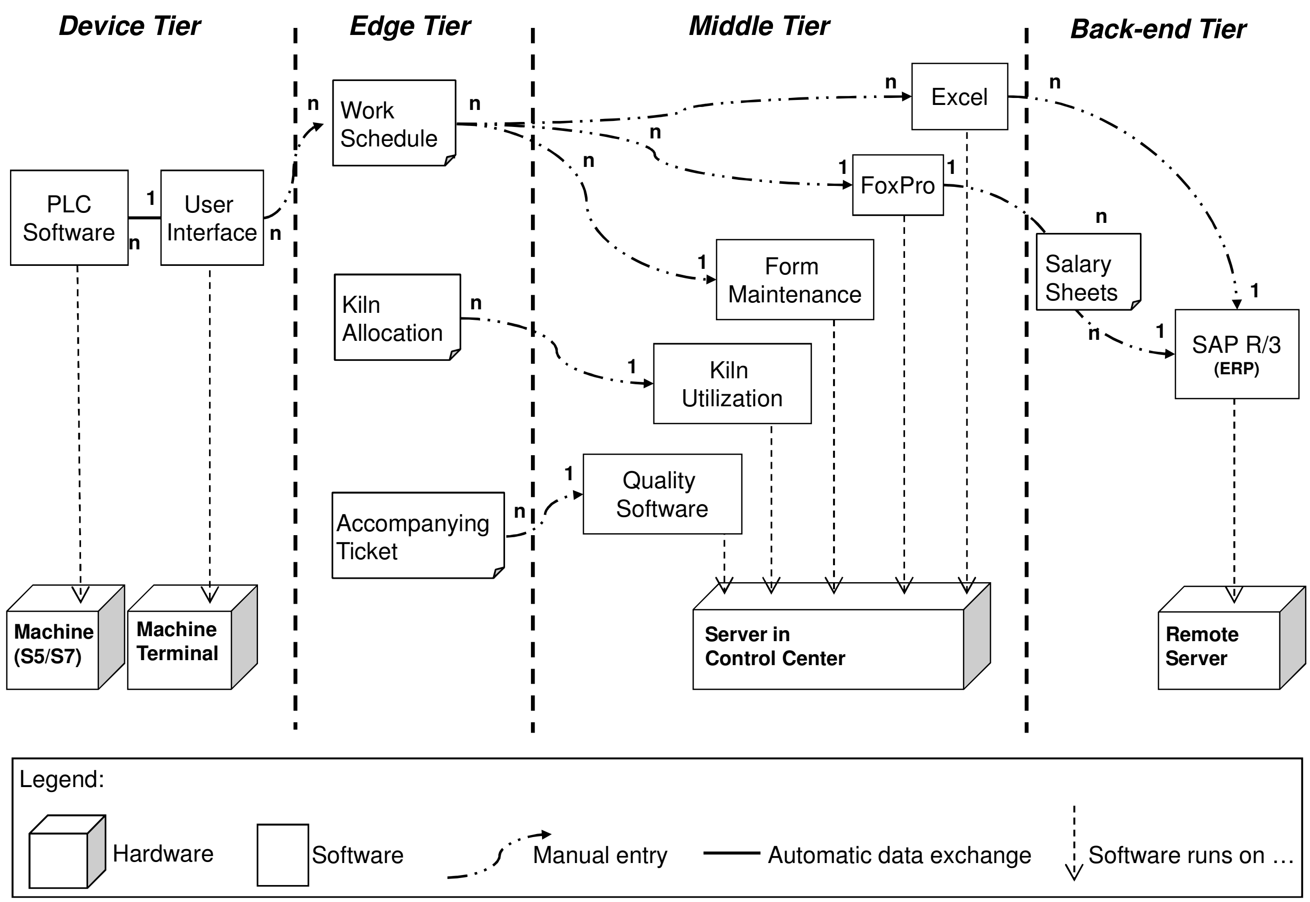

Figure 5. Deployment diagram of the software and hardware at REF.

Special about the architecture is that no software is deployed on the edge tier and the middle tier hosts only simple software solutions. REF uses no dedicated MES or PDC system. Instead, a range of very simple – often self-implemented – tools are in place for managing and documenting production tasks. This solution accounts for the limited resources available in IT.

Case ENG

ENG is a manufacturer of engines. The investigated plant is highly automated. ENG realized completely paperless data management on the plant floor and also avoids manual configuration of machines. In addition, ENG uses an automated transport system for material movements on the plant floor. The most relevant system requirements in the environment are:

- Openness/Adaptiveness
- Reliability
- Fast response times
- Scalability
- Support for event driven communication

We found nine software solutions (or classes of solutions) at ENG that are relevant for production activities and the related data management. These are an ERP system (SAP R/3), an integration software (SAP XI), a facility control system (FCS), a visualization system (VIS), a control system for automated transports (ATS), operator clients for plant-floor workers, station control systems for production facilities (SCS), programmable logic controller (PLC) software for machine control, and CNC programs for robot control. **Error! Reference source not found.** shows the deployment of these systems at ENG along with logical communication links.

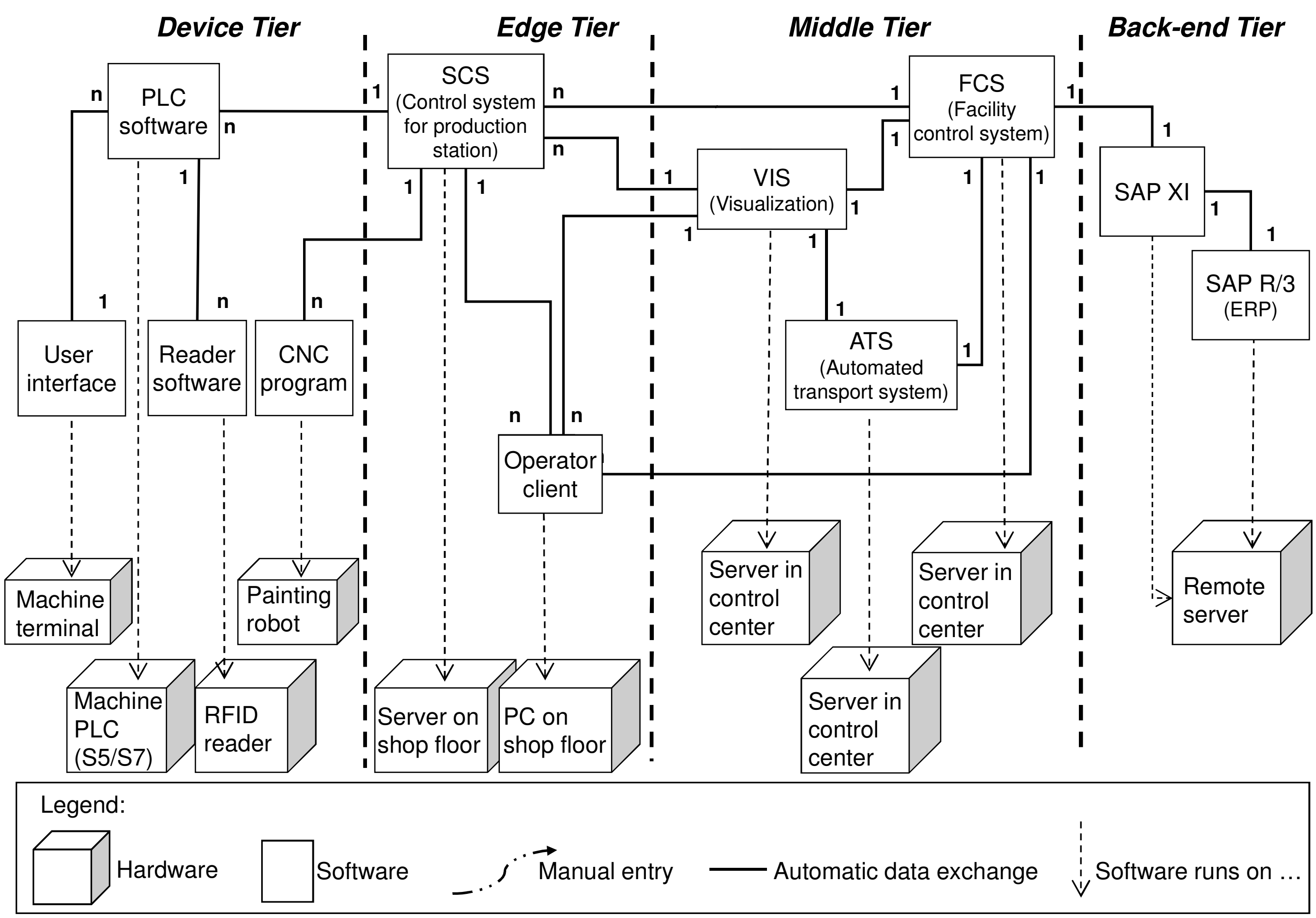

Figure 6. Deployment diagram of the software and hardware at ENG.

Several aspects are remarkable about the IT architecture at ENG. One is that each hardware tier at ENG hosts a lot processing logic. This design ensures *scalability* because much of processing is done decentralized on lower system layers. ENG further reduces the system load by using *event driven* communication rather than regular polling of data.

Also noticeable is the high degree of autarky of the system components. ENG uses dedicated control servers for each production station. These servers cache production tasks for several hours in advance and can operate independently. In addition, operator clients cache upcoming tasks and can respond to operators' requests solely using local data.

With the autarky of system components, ENG achieves *reliability* by architecture. Due to this design ENG can – at least partly – continue production operations for several hours even if parts of the system fail. Using local caches in each layer further does not only support the autarky but also enables fast system responses.

Finally, it is interesting to note that ENG implemented significant parts of the system itself or in tight cooperation with software vendors. This gives ENG a very high degree of control over its infrastructure and ensures *openness* to changes.

Case CHE

CHE is a globally operating company producing various chemicals. The plant is highly automated and a well functioning IT system is crucial for the production. During the production process, CHE must control a large number of actuators (such as valves) and evaluate sensor data in real time. According to the IT staff, the most important requirements in this environment are:

- Openness/Adaptiveness
- Reliability
- Fast response times
- Scalability
- Support for event driven communication

Error! Reference source not found. shows the architecture of the IT solution at CHE. The device layer at CHE controls hardware such as boilers, tanks, and pipes that are used in the production. The PLCs steer actuators like agitators or valves and capture sensor data like temperature or pressure values from the machines. The PLCs receive control information from the edge layer and pass back event data about sensor measures.

The edge layer at CHE comprises WinCC clients and Servers. The WinCC servers act as hubs for the communication with devices on the device layer, the WinCC clients, and system components in higher layers. The WinCC clients provide interfaces for monitoring and controlling operations on the plant floor. In addition, they integrate interfaces of the Simatic Batch system in the back-end layer.

The middle tier host most functionality. Among others this includes a historian (OSI PI), and a solution for managing alerts (Matrikon Process Guard), a Laboratory information management system (LIMS), and self implemented tools to support data analysis (Batch to SQL, Analyze, Wrapper GUI). The core systems in the middle tier are Simatic Batch and Simatic IT.

The central component for controlling the production is Simatic Batch. This component receives the recipes for scheduled production processes from the ERP and controls the execution of the production steps accordingly. Synchronization with the ERP in the Back-end tier is done via Simatic IT.

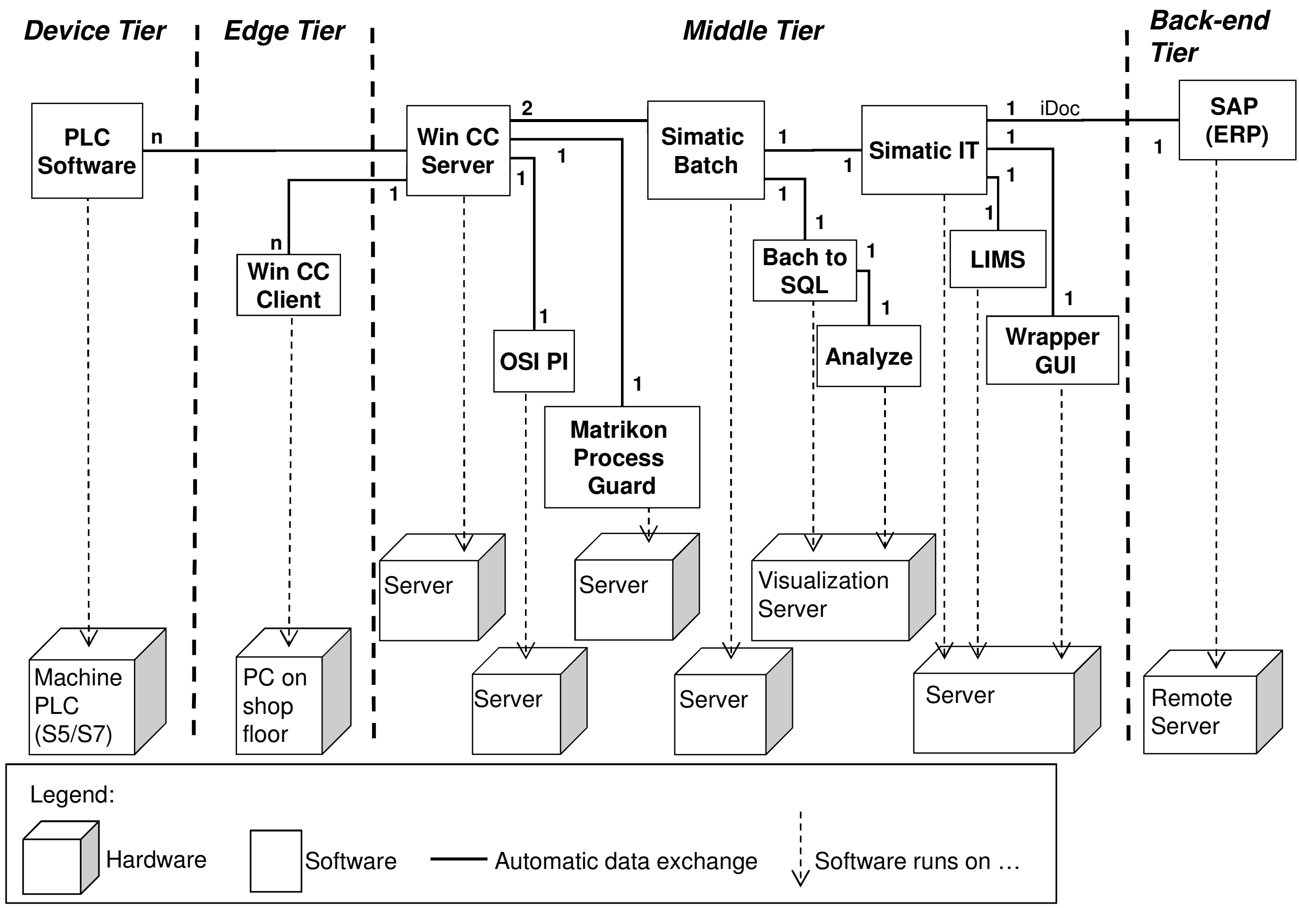

Figure 7. Deployment diagram of the software and hardware at CHE.

The infrastructure of CHE is an example of architectures with centralized control. That is, a small number of powerful computers are used for production data management. This architecture is necessary because a large number of actuators and sensors in the plant must be controlled at once and hence from a single point. *Reliability* is ensured by running crucial components on redundant clusters. CHE addresses the demand for *fast response* time and *scalability*, by decentralizing processing where possible. That is, CHE uses two instances of WinCC servers to distribute load. Using *event driven communication* for alerts and status updates reduces network load. The demand for adaptiveness has led to a number of extensions to the system. Self-implemented solutions like "Batch to SQL," "Analyze," and "Wrapper GUI" provide specialized functionality for CHE.

Case POW

POW produces power plants that include numerous pipelines in diverse sizes and dimensions. In our case study we investigated a plant that produces the pipes which are later assembled in plants. POW's foremost requirements for its IT infrastructure are:

- Openness/Adaptiveness
- Lightweight and simple

Error! Reference source not found. shows the physical deployment of involved software systems along with logical communication links. The device layer comprises machines and PLC devices. The only software that runs in the device layer is the PLC software. The company uses no terminals for machine control. That is, workers configure the machines manually. The reason for this thin device layer lies in the nature of the produced products. The company does not use many machines during the welding and bending process. Most work needs to be done manually.

The edge layer includes logging programs that log data from the PLCs. Some customers desire reports about the production process. In such a case POW extracts the desired data from the logging program and feeds them into reports. The data extraction is done on a case-by-case basis with manual queries. Besides the logging program and the reports, the edge layer also comprises ERP clients and BHR clients. The middle layer comprises the CAD system. The CAD system is, for example, used for design and construction of new power plants. Furthermore, we find a self-developed MES (BHR) in this layer.

The back-end layer includes the ERP system, a business intelligence system, and a fileserver. The fileserver is used for all backups of all logged data worldwide.

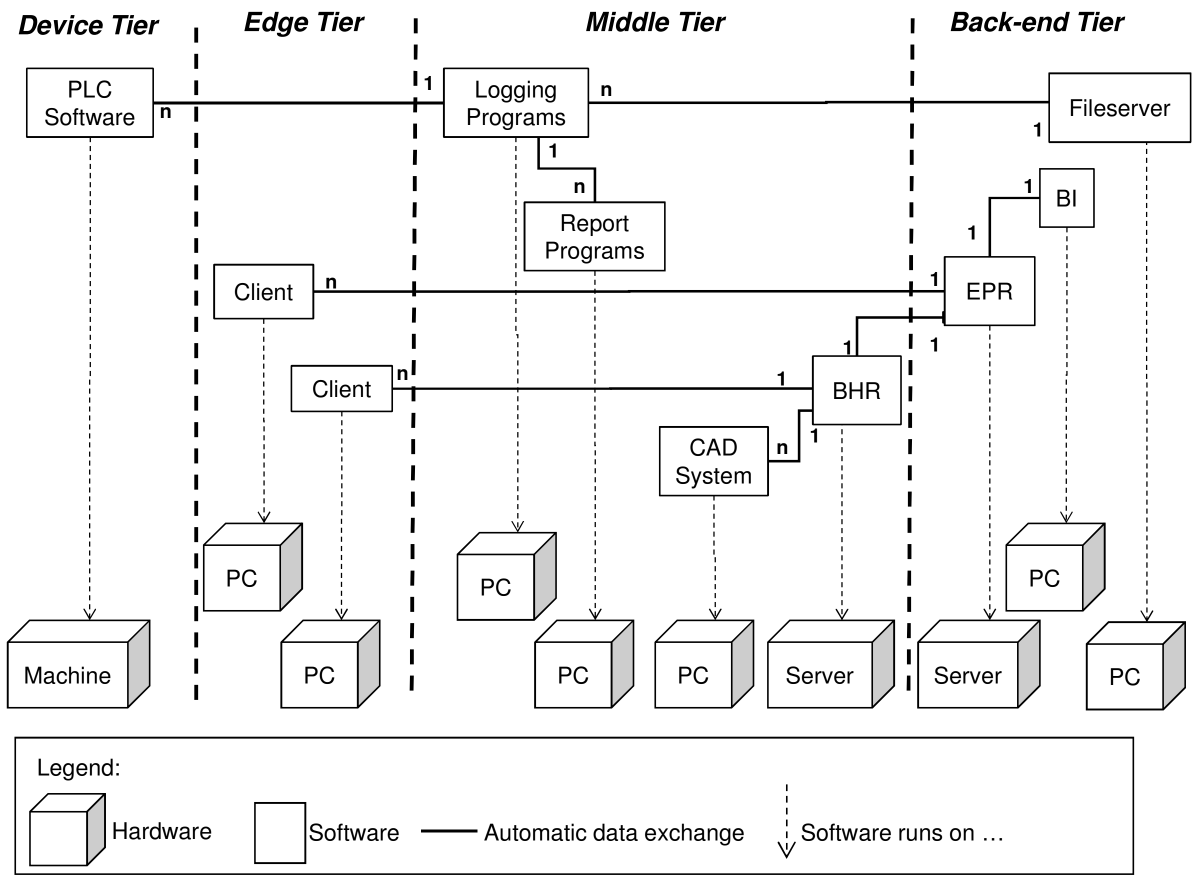

Figure 8. Deployment diagram of the software and hardware at POW.

Remarkable about the IT infrastructure is the thin edge layer. This design is possible because shop-floor operations at POW require only little or no IT support. The currently implemented solution meets POW's desire to run a *lightweight* solution that is *open* for the implementation of changes.

Case TIR

TIR is a manufacturer of tires for cars and trucks. The production is automated to a large degree but includes steps where shop-floor operators need frequent system interaction. According to interviews with IT staff, the most important system requirements are:

- Openness/Adaptiveness
- Reliability
- Fast response times
- Scalability
- Support for event driven communication

The IT infrastructure at TIR and its components are depicted in **Error! Reference source not found.**. The device layer includes all machines and their corresponding PLC devices on the shop floor. Software in the device layer includes the PLC software and the user interfaces of the machine terminals.

The edge layer at TIR comprises only the MES clients. Via these clients, plant floor workers receive descriptions for the production tasks. They also use the MES clients for manual data entry to report about the production.

The back-end layer comprises the MES, ERP, a separate data warehouse, detailed planning software, time recording application, and a separate specification database. The specification database and the data warehouse lie each on a central server hosting the data of all plants worldwide.

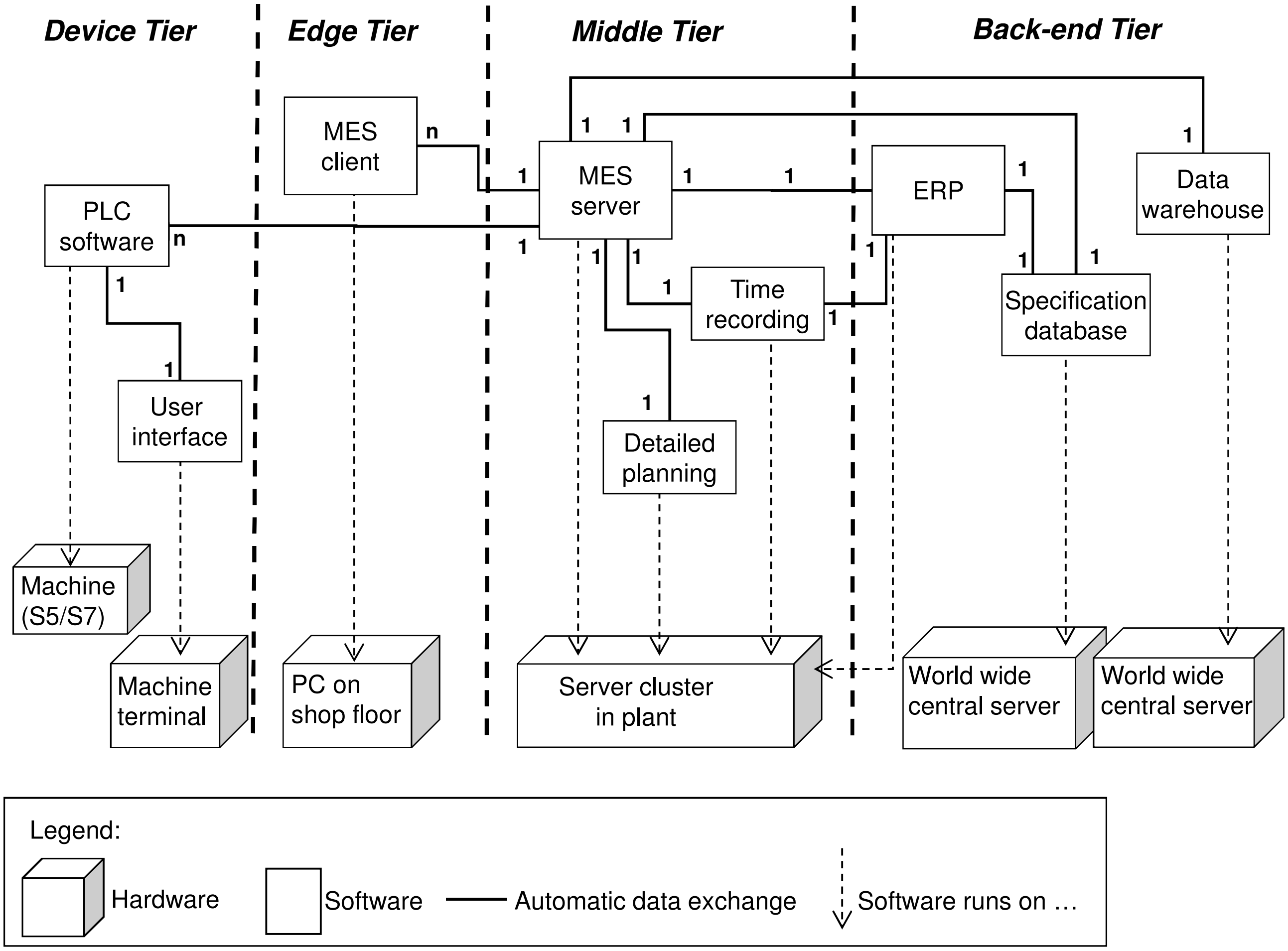

Figure 9. Deployment diagram of the software and hardware at TIR.

The infrastructure of TIR is *open* to changes and the IT staff makes frequently use of the option to add new functionality. The system architecture is an example for solutions with centralized control. The majority of the systems lay in the back-end layer. This design originally resulted in slow system responses during peak load. TIR addresses this issue using buffers and caches on the lower system layers. TIR thereby meets the requirements of *fast system responses* and *scalability*. Using *event driven communication* further helps TIR to reduce network load. *Reliability* is ensured by running crucial system components on redundant clusters.

LESSONS LEARNED

This section generalizes the individual findings of the case studies, focusing on different approaches for monitoring and controlling production. We first review the requirements found in each case study and match them against the distribution of software logic throughout the different hardware tiers. Here we point out common sets of requirements and corresponding architectural choices. Following that, we delve in more detail into the architectural choices for production control. Here we identify two common patterns and discuss their advantages and disadvantages respectively.

Requirements and Corresponding Distribution of Processing Logic

In Table 1 we summarize the companies' requirements on MES or PDC systems. Regarding the frequency of requirements, we found the following: *Adaptability* of the system was most important in the analyzed group of companies. Four out of seven companies mentioned this as an important criterion. In contrast, the requirement of a *lightweight* solution is not seen as relevant in these cases. We found this requirement explicitly in those cases where a PDC system is not crucial for maintaining the production processes. The case studies further show that *reliability*, *scalability*, and *fast response times* of the system are often important in combination. Regarding security, we found that it is not major concern. Event-driven communication has appeared as an important paradigm. In five cases this was explicitly demanded.

Despite the heterogeneity in requirements, we found two clusters of similarities. These are highlighted in Table 1. One cluster comprises companies with very demanding requirements for the IT. This cluster is characterized by the combined demand for reliability, performance, and scalability (see MIL, ENG, CHE, TIR). The other cluster of requirements is characterized by the absence of requirements for reliability, performance, and scalability. Instead, in all these cases the demand for a lightweight system was prevalent (see COO, REF, POW).

The analysis of IT architectures revealed huge case-specific differences. However, we found two clusters with similarities in the distribution logic. The right side of Table 1 provides an overview of which hardware tier hosts much processing logic in which case. Horizontally we list the four hardware tiers as introduced in section 2. In the table we put an "X" where much processing logic was deployed on the respective tier, an "(X)" where some logic runs in the tier, and a blank cell where no or almost no logic is deployed.

Table 1 reveals that the clusters in requirements show correspondence to the distribution of logic. Cases that demanded a *lightweight* system only have deployment of logic in the back-end tier and the device tier in common. The middle tier and the edge tier host only little or no logic. Cases with requirement for *reliability*, *fast response times*, and *scalability* have much logic deployed on all hardware tiers. An interesting exception is the case of TIR. This manufacturer initially realized little logic on the edge layer. However, the initial system design caused performance problems. TIR reacted to this problem by moving tasks to the lower system layers and thereby moving toward the typical logic distribution for this cluster of requirements.

Table 1: Requirements and corresponding distribution of processing logic

Case	Requirements							Tiers with "much" processing logic			
	Adapt. open	**Lightweight**	**Reliable**	**Fast**	**Secure**	**Event Driven**	**Scalable**	**Device Tier**	**Edge Tier**	**Middle Tier**	**Backend Tier**
MIL	X		X	X		X	X	X	X	X	X
ENG	X		X	X		X	X	X	X	X	X
CHE	X		X	X		X	X	X	X	X	X
TIR	X		X	X		X	X	X	(X)	X	X
COO	X	X				X		X	(X)		X
REF		X						X		X	X
POW	X	X						X	(X)		X

Central versus Local Production Control

We observed two distinct approaches to production control: centralized and decentralized production control. It depends on the production environment which option is favorable. Figure 10 depicts production control with a central component. Here the control is accomplished in three steps. The first step is generating a production plan in the ERP system. This can be mapped to ISA-S95 level 4 activities. The production plan is then coordinated in the second step. This includes sending control commands and job instructions to the hardware on the shop floor. These activities fit to ISA-S95 level 3. The instructions are performed on the shop floor and events reported back to level 3.

This approach has little functionality on level 3. It is suitable if an integrated view of all actuators is required. Examples for such IT infrastructures are the case studies MIL and CHE. It is also suitable for lightweight infrastructures like REF.

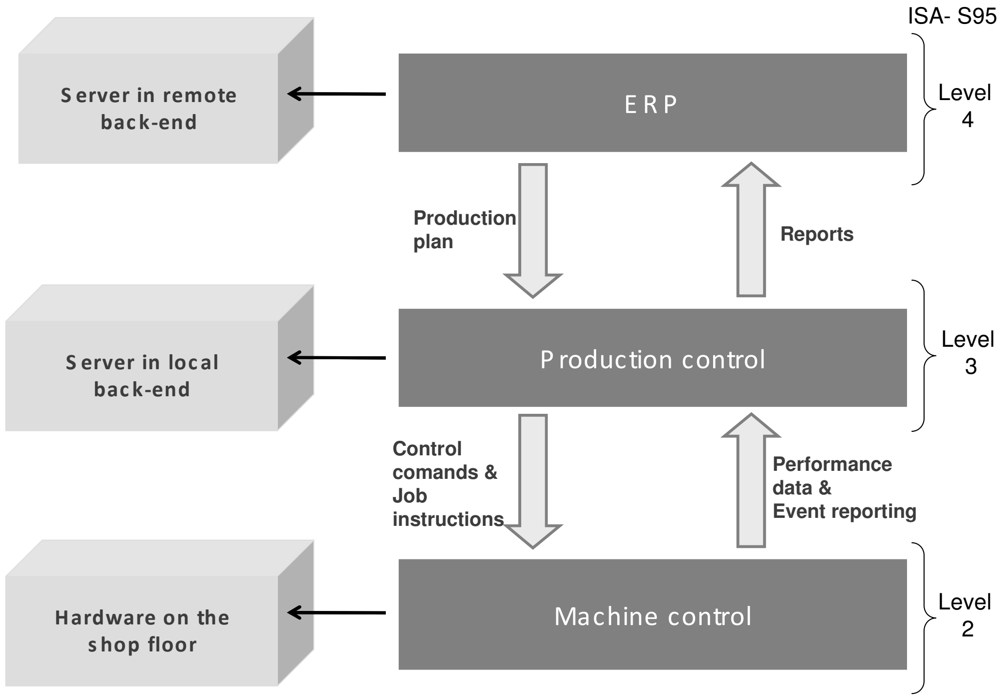

Figure 10. Central production control.

The second approach is decentralized – typically hierarchical – production control, see Figure 11. In its structure this architecture is related to the Presentation-Abstraction-Control pattern for hierarchically organized software agents (Laurence and Coutaz 1991). To some extent it realizes the paradigm of Distributed Autonomous Systems for manufacturing (Iftikhar 2004). Here, production control is conducted in four steps. The first step is also generation of a production plan in the ERP system. We can also map this to the ISA-S95 level 4. In the second step a detailed production plan is generated from the production plan. This is then split up and sent to the appropriate systems (mostly basic PCs) on the shop floor. They are responsible for the production control of specific processes and the corresponding machines.

This approach has substantial functionalities in level 3. Information is kept redundant in the system. This enables a high level of scalability. It also supports productions with well encapsulated tasks. An example is the case ENG. The advantage is higher autarky of all system components, which reduces the problem of bottlenecks. Through this decentralization, the IT systems can react faster to changes on the shop floor. Nevertheless, one has to keep the disadvantage in mind that maintenance is more complex.

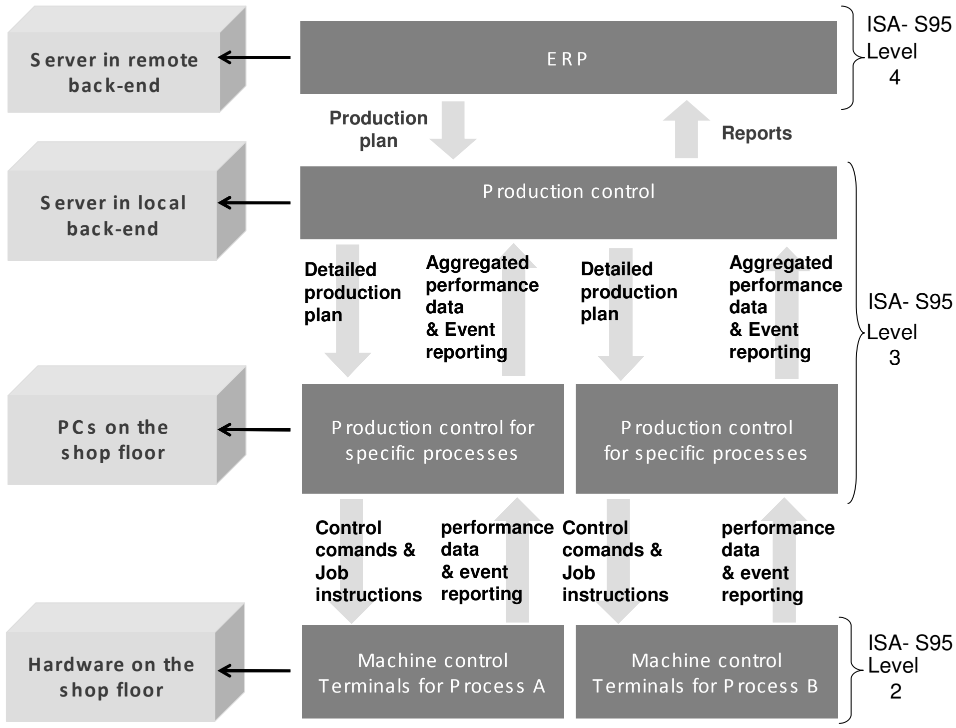

Figure 11. Decentralized production control.

CONCLUSION

The seven case studies provided us with insights on issues and trends in the architectural design of IT solutions in manufacturing. Our studies show that there is no "one size fits all" IT architecture. All infrastructures are tightly customized to the specific needs of the company. However, we identified clusters with similarities within the solutions. This is reflected in common deployments of logic for certain sets of requirements (see 4.1) and reoccurring architectures for production control (see 4.2). In addition to the above discussion, we now summarize six key insights that should help software vendors and system architects in designing future manufacturing solutions:

1. **Manufacturers want control of the IT system**: Adaptability of the system was an explicit demand in six out of seven cases. The reason is that production environments vary a lot and manufactures want to tailor their IT solution to their specific needs. Also, the IT staff must adapt the system when production processes change or demands for new data processing tasks arise. Software vendors should take this into account by providing open interfaces and support for extensions.

2. **Security is of low relevance:** Security was not perceived as important aspect of the IT in any of the seven cases. Manufacturers care about security but do not consider it part of manufacturing execution and data acquisition. They rather add a security layer to shield their system from external attacks.

3. **Processing on the edge is required**: In four out of seven studies manufacturers deployed considerable parts of their IT functionality across all four hardware tiers. This design was chosen to meet requirements of *reliability*, *scalability*, and to ensure *fast response times*. The layered design along with caching on each layer makes system components less dependent on each other. This is an important property for manufacturers with a high dependence on the IT. To support these manufacturers, software vendors should enable the distribution of functionality across hardware tiers and provide clear interfaces for that.

4. **Event driven communication is desired:** Five out of the seven investigated companies considered event driven communication relevant for their current or future operations. Manufacturing is a domain where systems must often react to events rather than simply process transactions (e.g.: a production step is completed or a threshold is exceeded). Software vendors should therefore directly support this communication paradigm.

5. **Buffers are needed on every system layer:** Manufacturers buffer routing data and production at several layers in their architecture. This supports the autarky of layers and allows fast system reactions based on locally cached data. Particularly communication with the back-end usually runs through buffers to make the production independent from availability of the ERP and WAN. Two manufacturers needed particularly fast response times for operator clients on the plant floor (TIR, ENG). These manufacturers made extensive use of caches in the edge tier. Software vendors should support such solutions by providing persistent storage solutions for components on different hardware tiers.
6. **Systems should have a lightweight footprint:** For three out of seven investigated manufacturers the foremost requirement was that the IT system is lightweight and simple. These manufacturers need only little IT support and want to avoid the cost of managing a complex system. Software vendors should meet the demands of such manufactures by providing a system with a small footprint that can be extended as desired.

The results in this paper provide insights into current issues in real world IT infrastructures in manufacturing. Software vendors and system architects should keep the six points above in mind to tailor future solutions more tightly to demands in the manufacturing domain.

REFERENCES

1. Brandl D. ed. (2005). ANSI/ISA}-95.00.01 Enterprise-Control System Integration Part 1-3: Models and Terminology, ISA Research Triangle Park, North Carolina, USA.
2. Iftikhar Ahmed (2004), Towards the Next Level of Abstractio - The Autonomous Decentralized Systems - An OVerview, Studies in Informatics and Control, Vol 13, No. 4. December 2004
3. Vollmann, T., (2005). Manufacturing Planning and Control Systems for Supply Chain Management. New York: McGraw-Hill.
4. Laurence Nigay and Joëlle Coutaz (1991), Building user interfaces : Organizing software agents. In: Proc. ESPRIT'91 conference, Bruxelles,
5. OAGIS (2007). Open Applications Group Integration Specification (OAGIS), http://www.openapplications.org.
6. OPC Foundation (2007) OPC. http://www.opcfoundation.org. Accessed 17 Feb 2009
7. RosettaNet (2008). RosettaNet, http://www.rosettanet.org/RosettaNet/.
8. Williams T. ed. (1992). The Purdue Enterprise Reference Architecture -- A Technical Guide for CIM Planning and Implementation, ISA Research Triangle Park, North Carolina, USA.