Building A Web Service Technology-Based Factory IT-Landscape Using WebObjects

Matthias Meier, Philipp Dreiss, Ralf Muckenhirn
Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), Stuttgart, Germany

IPA
Fraunhofer Institut
Produktionstechnik und
Automatisierung

Introduction

Factory automation – the current situation

The IT architecture that is operated in many factories today has been growing over the years. As the requirements for tracking, data acquisition, production control, etc. became more demanding, more and more applications were added. Thereby, the complexity of the IT landscape increased significantly, especially due to the large number of more or less documented dependencies between several applications. Maintenance and management of these environments becomes more and more difficult.
At the same time, the importance of software on the shop floor has been increasing. Production lines cannot be operated anymore if the software is not available. Software issues can easily delay the ramp phase of a new production line or be a limiting factor for further development of existing factories.
On the other hand, the market situation changes frequently and requires companies to adapt to changes very quickly. An agile IT landscape is therefore a prerequisite for success.

Fig. 1 gives an insight into a 300mm semiconductor manufacturing line. Although the semiconductor industry was the starting point for our work, the results are by far not limited to this area and can be transferred to other industrial sectors.

(Source: Infineon Technologies AG)

Figure 1: Semiconductor Manufacturing Line ("Wafer Fab")

Factory automation – the current IT landscape

Looking into semiconductor production lines four basic software layers can be distinguished (Fig. 2). The bottom most layer is the equipment software layer (~300-600 equipments). It has to deal with hard real-time conditions within an equipment, processes sensor data and controls actuators. On top of it is the equipment integration / automation layer. It realizes an adapter to upper layers and implements equipment or equipment type specific scenarios (~80 types). A factory integration layer connects the equipment integration instances and applications of the factory automation layer which control and monitor the overall production facility (~several hundred applications). On top of this stack resides the company's ERP system (not shown in Fig. 2).

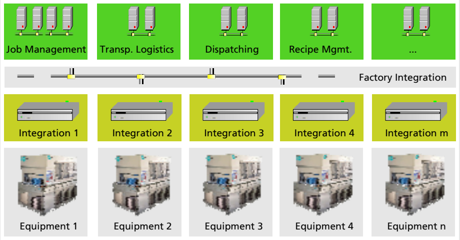

Figure 2: Current production IT landscape (green: factory automation layer)

The Approach

Current issues and goals

The production IT landscape in today's factories is often highly heterogeneous. A large number of different applications, running on different operating systems, using a variety of databases and communicating through a number of protocols need to be operated, maintained and extended in order to fulfill the business needs. This causes significant development, integration and maintenance costs. Furthermore, such systems are extremely sensitive to changes and need specifically skilled developers and operators to make them run stable.

The goal of our work is to overcome many of these issues. Starting on the factory automation layer with the Manufacturing Execution System (MES), our focus is to:

- Simplify the overall architecture by focusing on a smaller number of concepts and technologies.
- Make use of concepts and technologies that are widely used on the Internet.
- Implement concepts that are applicable in a broad range of industries.
- Realize an integrated concept for the complete factory IT landscape.
- Avoid high license costs.

Analysis

As the need for automation is particularly high in semiconductor industry, similar approaches focused on this area of application have been started before some years ago [1, 2]. These approaches were heavily based on CORBA and OO frameworks. Neither CORBA nor the new architecture proposals were really accepted throughout the industry.
In the meantime the boundary conditions have changed. Web technologies, such as Web Services, have been broadly accepted across all industries – many commercial applications offer Web Service interfaces, for example. Open source as well as commercial implementations and toolkits are available on the market and provide a basis to build products upon. Furthermore, a significant amount of experience using these technologies has been collected over the last years and can be taken into account for the development of new systems. For this reason we think it's the right time for new proposal to overcome the issues described above.

In the first analysis phase we identified a minimal set of services we need to provide to cover the basic requirements for the MES system (Fig. 3). It consists of a collection of services to monitor and control the factory, to realize the desired level of traceability and to collect management information, for example numbers that can be used for the calculation of the overall equipment efficiency, etc.

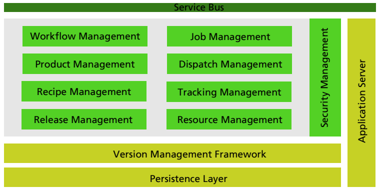

Figure 3: A collection of basic MES services

The services shown in Fig. 3 are exposed to potential requestors using a simple service bus. The service bus provides the capability to advertise, discover and request services. Service clients within the MES and outside the MES use the service bus to request the services they need.
In contrast to many existing solutions we follow the approach to separate the MES system into distinct subsystems that communicate through their service interfaces only. There are no additional dependencies introduced by shared data sources.

The Prototype

The server side – WebObjects-based

WebObjects [3] lays the foundation for the server side of our prototype. In addition to the out-of-the-box functionality, for example the EOF persistence layer, we identified a set of additional basic aspects we needed to realize our applications:

- A version management framework for entities
- A release management framework for entities
- A security framework for authentication / authorization
- A service discovery framework

These aspects have been implemented within the WebObjects environment.
All required MES services that have been identified in our analysis phase were implemented as independent WebObjects applications. Each application uses it's own EOModel and has a clearly defined scope which is reflected by its Web Service interfaces. Any external communication of these applications is based on Web Service requests.

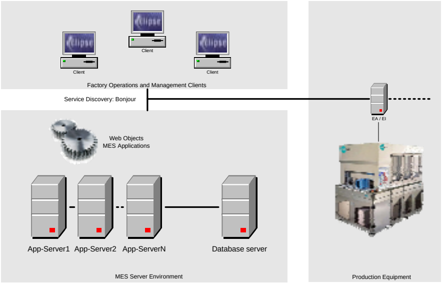

Figure 4: Prototype Components

The client side – Eclipse-based

The GUIs of our MES system were implemented based on the Eclipse framework [4]. Thereby we achieve a common look and feel for all the GUIs for the different tasks. Users are presented different perspectives depending on their current role (Figure 5 shows the product specification perspective for an artificial assembly example). The plug-in concept ensures easy extensibility for the future.

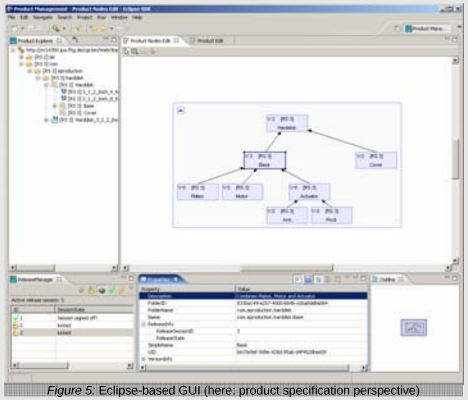

Figure 5: Eclipse-based GUI (here: product specification perspective)

Standardization Potentials

Standardization activities within the semiconductor industry are facilitated by the SEMI organization [1]. These activities prepare the ground for open markets and for a reduction of manufacturing costs. Major standardization efforts in the field of automation software have been spent on the equipment integration / automation layer as suitable interfaces between equipment and equipment integration layer are a prerequisite for realizing sophisticated automation scenarios. With the new interfaces A (equipment interfaces) [5], B (interfaces between equipment engineering systems) and interface C (diagnostics and maintenance interfaces) Web technologies have found their way into the standards.

Based on the results of our studies we propose to drive these steps further to the higher layers of the automation software stack. An architecture as described before using widely accepted Web Service technologies to connect software components that have a clearly defined scope opens the doors for further standardization activities. We expect significant benefits if these standards were defined and implemented by the industry as the integration costs were lowered and the market would be opened for specialized solution providers.

Conclusion

The production IT landscape which is implemented in many factories today causes high integration and maintenance costs, is difficult to extend and lacks the agility required by fast changing markets.

Within our project we were able to show that a Web Service technology-based factory IT landscape, especially on the level of the manufacturing execution system, has the potential to improve the situation significantly. Concepts and frameworks that have proven their value on the Internet, such as WebObjects, for example, can be reused in the area of factory automation. Using them as a basis for the development of the automation solution speeds up the development, saves costs and paves the way for new promising standardization activities.

Although the approach we propose was explained using examples from the semiconductor area, it can be easily transferred to other industries. In our current research project we focus on small and medium sized enterprises producing large volumes of miniature/micro-products (MASMICRO). From our point of view they can benefit from the experience of the semiconductor industry.

For the next months we plan to further extend our system according to production needs, to setup a demonstration environment, to evaluate high load and exception scenarios and to deploy the system to production.

References

Literature

[1] Semiconductor Equipment and Materials Institute – SEMI
http://www.semi.org

[2] SEMI E81: Provisional Specification for CIM Framework Domain Architecture, 1999

[3] Apple WebObjects 5.3 http://www.apple.com/webobjects

[4] Eclipse.org http://www.eclipse.org

[5] SEMI E120, E125, E132, E134 Standards, 2006/2005

Project sponsor

The work presented here has been created in the context of the MASMICRO project. This is an FP 6 integrated research project sponsored by the EU.
Please visit http://www.masmicro.net for more information.