Framework for integration of virtual metrology and predictive maintenance

G. Roeder, A. Mattes, M. Pfeffer, M. Schellenberger, L. Pfitzner, Alexander Knapp, H. Muhlberger, A. Kyek, B. Lenz, M. Frisch, J. Bichlmeier, G. Leditzky, E. Lind, S. Zoia, G. Fazio

Framework for Integration of Virtual Metrology and Predictive Maintenance

Georg Roeder[1], Andreas Mattes[1], Markus Pfeffer[1], Martin Schellenberger[1], Lothar Pfitzner[1], Alexander Knapp[2], Heribert Mühlberger[2], Andreas Kyek[3], Benjamin Lenz[3], Markus Frisch[3], Josef Bichlmeier[4], Günter Leditzky[5], Erich Lind[5], Silvia Zoia[6], Giuseppe Fazio[6]

[1] Fraunhofer IISB, Erlangen, Germany, [2] University of Augsburg, Germany, [3] Infineon Technologies AG, Regensburg, Germany, [4] camLine GmbH, Petershausen, Germany, [5] austriamicrosystems AG, Unterpremstaetten, Austria, [6] Micron, Agrate, Italy

Abstract— **Within the ENIAC project "IMPROVE", new algorithms for virtual metrology and predictive maintenance are being developed to substantially enhance efficiency in European semiconductor manufacturing. The consortium comprises important IC manufacturers in Europe, solution providers, and research institutions. A major objective of the project is to make these new APC methods applicable in the existing fab systems of the IC manufacturers which widely differ in the automation infrastructure. A novel framework architecture for integration of the new control paradigms was researched and a software for implementation of the framework was developed. This paper describes the technical details and results of the framework development, implementation, and test.**

Keywords - Advanced Process Control, Virtual Metrology Predictive Maintenance, APC framework, SOA, Equipment Engineering System

I.   INTRODUCTION

In semiconductor manufacturing but also in related industries, e.g. in TFT manufacturing, the implementation of advanced process control (APC) systems has become essential to achieve technological requirements at continuously reduced process windows together with cost effective manufacturing. APC methods such as fault detection and classification (FDC), or run-to-run (RtR) control have been well implemented in IC fabrication and aligned with established quality control solutions such as statistical process control (SPC). To ensure high process stability, high production yield and to support APC methods, stand-alone and integrated metrology of critical parameters remain key enablers to reliably determine quality parameters on monitor or product wafers. Unit process monitoring and control should be performed in close time-frame after wafer processing possibly on a wafer-to-wafer basis or even in real time which may not be possible or affordable even with fast in-line measurements. This shortcoming is addressed by the concept of Virtual Metrology (VM) which is defined as the technology of *prediction of post process metrology variables* (either measurable or non-measurable) using process and wafer state information that could include upstream metrology and/or sensor data [1, 2]. Fig. 1 illustrates the typical place of execution of VM in a process flow: One or several process modules are embedded in a sequence of other process steps and monitored by pre-process and post-process metrology. Typically, equipment and process data is available through an FDC system which can then be provided to a VM module and supplemented with context information. The primary objective and key capability of the VM is to precisely predict a quality parameter of the wafer, typically on a wafer-to-wafer basis, e.g., by applying statistical models.
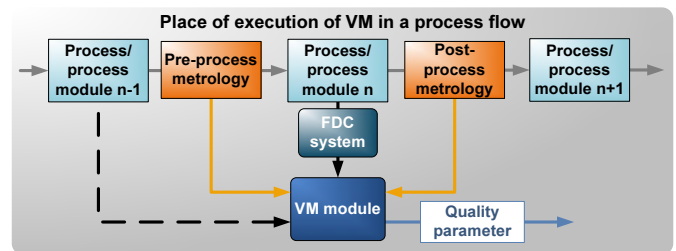

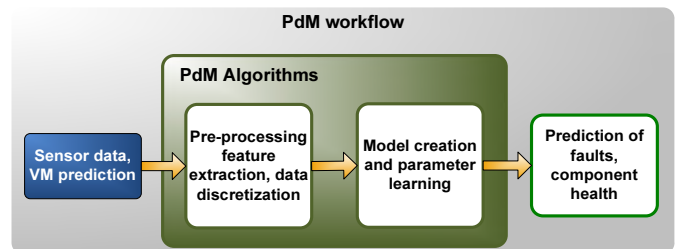Figure 1.   Place of execution of VM in a typical process flow.

Figure 2.   Predictive Maintenance (PdM) workflow.

Equipment maintenance is an additional relevant cost factor in semiconductor manufacturing. Currently maintenance is typically scheduled based on elapsed time or fixed unit count usage and maintenance frequency mainly depends on the process engineer's experience and on known wear out cycles of certain parts of the tool. The concept of predictive maintenance (PdM) aims at predicting upcoming equipment failures or events, their root causes and corresponding maintenance tasks in advance to improve uptime and availability and to reduce

operational cost and scrap. The concept of PdM is shown in Fig. 2. Data from equipment, metrology and sensors as well as VM predictions are provided to the PdM module. Statistical modeling is applied to predict upcoming faults and required maintenance actions based on the equipment health status as key number. Although being new APC methods, VM and PdM have received significant attention and a variety of publications on the development and application of VM and PdM algorithms are available. An overview is provided in [3-5] for VM and in [6-8] for PdM, however a comprehensive compilation is beyond the scope of this paper.

Another key requirement for VM and PdM is their capability for integration into an existing fab infrastructure. Several authors have addressed this issue from conceptual approaches to proposed implementation solutions. Qin et al. [9] elaborated a concept for a fab-wide RtR control system which included supervisory control strategies for fab-wide control. Khan et al. enhanced this concept by the application of VM including process examples and an assessment of issues for employment of VM on fab-level [10]. Cheng et. al. [11-13], Huang et al. [14, 15], and Su et al. [16-18] provide a thorough description of solutions for an automated VM framework including various stages of automation levels, framework components and their integration into the fab environment, embedded devices for the distributed information collection and transfer and the integration of the automated VM framework into manufacturing execution systems (MES). Pan proposes a centralized fab-wide architecture utilizing the MES and FDC system and including real time SPC for VM supervision [19].

An objective within the European ENIAC project "IMPROVE" is to develop and implement new algorithms for VM and PdM to substantially enhance efficiency in semiconductor manufacturing. A grand challenge within this task is that these new APC methods should be made applicable in the existing fab systems of the IC manufacturers which widely differ in the automation infrastructure. Hence, a novel framework architecture for integration of the new control paradigms was researched and a software for implementation of the framework was developed beyond the capabilities of existing solutions. This paper describes the technical details and results of the framework development, implementation, and test. The remainder of the paper is organized as follows. In Section II, the approach taken to develop a generic and reusable architecture for the implementation of VM and PdM together with other control entities is described. In Section III, the concepts for modeling the generic architecture and its mapping to the real-world fab infrastructures are elaborated. The software implementation of the framework is discussed in Section IV. In Section V, the implementation of the framework, the specific modifications, and the results of the framework tests are elaborated at the example implementations performed at three IC manufacturers' sites.

II. APPROACH FOR A GENERIC AND REUSABLE FRAMEWORK ARCHITECTURE

The challenge within IMPROVE to implement novel VM and PdM solutions for various fab infrastructures and operating at different automation levels required a comprehensive approach to ensure the reusability of the developed solutions and to avoid duplicated efforts and redundancies within specific implementations.

In an initial step, basing on the state-of-the-art as described in the literature, a survey amongst the fabs to assess the status of control infrastructures and the requirements for implementing new control paradigms was performed. This survey was complemented by collecting the state-of-the art in related industries, e.g. in the electronics and automotive industry. An additional key element during the specification was a collection of specific user requirements for development and implementation of VM, PdM, and a potential framework architecture. A catalogue with quantifiable criteria and well defined classification of user requirements was circulated amongst the IC manufacturers and feed-back was collected and consolidated. The classification of specific user requirements is depicted in Fig. 3.
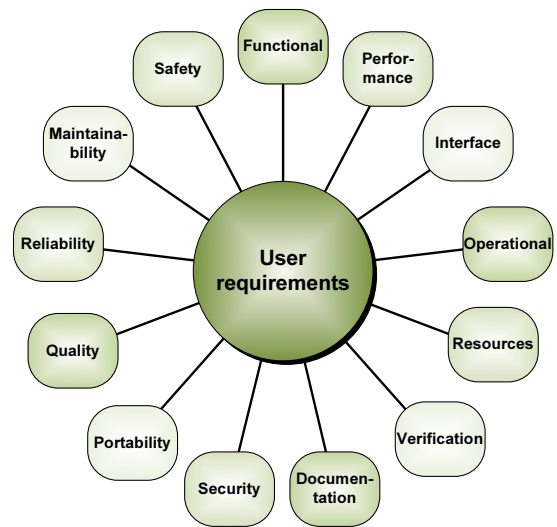

Figure 3. Classification of specific user requirements.

A major result of this analysis was that the requirements did not widely differ amongst the different users and specifications for a generic architecture could be consolidated and derived: The generic architecture should not rely on a particular IT infrastructure but be adaptable to different real-world ones; the data format should also be generic enough to capture arbitrary physical dimensions and annotations; the algorithms for the actual analyses should be not only implementable in different programming languages ranging from MATLAB [20] to C, but also be exchangeable statically by configuration as well as dynamically at run-time. In response to these requirements, in particular the demanded generality, the development approach abstracted from the specific IT infrastructures of the IC-manufacturers and introduced an "ideal" architecture on a conceptual level which could accommodate for the integration of VM and PdM modules into a common model. SEMATECH's Equipment Engineering System (EES) [21] was proposed as a starting point for the conceptual level and it was envisaged to define VM and PdM as Equipment Engineering (EE) applications (Fig. 4). This concept provided the advantage for the wide application of existing SEMI standards and, in fact, parts of the concept were included as inputs to an extension of the SEMI E133 standard [1].
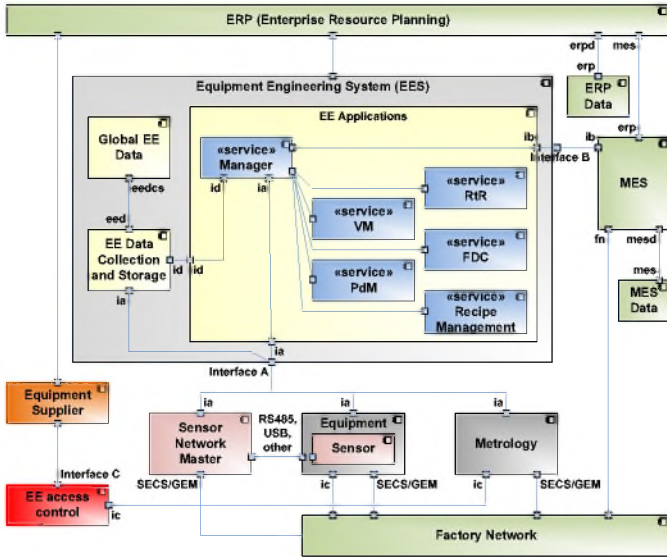
Figure 4. Definition of VM and PdM as EE applications on a conceptual level and described in UML.

However, it was necessary to turn the conceptual level into a concrete architecture that could be used as a blue-print for implementations and mappings to real-world infrastructures. For this purpose, the standard software modeling language UML ("Unified Modeling Language") [22] was chosen which offers precise and engineering-based means to express the static and dynamic structures of software systems.

In accordance with architectural best practices, the EES was modeled as a component system. Components make clear what the system boundaries are and which adapters are needed for fab-integration, in response to the user requirement for generality. The use of ports on and connectors between components provides an abstraction of the data to be exchanged but fixes what is provided and required by a component, in accordance with the user requirement on flexible data formats. Using hierarchical components it can be made clear where algorithms can be plugged in, answering the user requirement for flexible algorithms. However, in order to be able to exchange plug-ins also dynamically at run-time, the component modeling approach has to be complemented by an approach for reconfiguration. For this purpose a service-oriented methodology was chosen both for reifying static components and plug-ins (i.e., stubs for plug-ins). In fact, service-orientation blends well with component systems since the structure makes clear what is offered and provided by whom and who can access what. The paradigm of service-orientation induced to take the EE-applications, like VM or PdM, to be high-level services that have to be supported by low-level services for data gathering and merging. These low-level services could replace existing individual solutions for data collection limited to specific information and particular data bases, be it from the equipment or between the EES applications. Additionally, the high-level services can be aggregated by employing service composition techniques, like the usage of workflows. Fig. 4 shows the proposed architecture including basic VM and PdM modules modeled in UML. In a second stage, this ideal architecture was mapped to the existing infrastructures. For this purpose, the

infrastructures had also to be rendered partially as UML component models in order to identify the missing parts and the points of adaptation.

III. MODELING OF THE FRAMEWORK

The primary goal of the IMPROVE framework is to serve as a transparency layer between the different fab architectures and the EE application modules (VM, PdM) which also provides capabilities for the storage of fab data and prediction results. This is realized with a service-oriented approach that uses a data meta-model with general contexts in the form of key-value pairs. Workflows govern the sequence of processing. Fig. 5 shows the main components of the IMPROVE architecture described in UML. The components on the left side are fab specific and connected to the framework with three defined interfaces. The Data Gathering Adapter can push fab data to the framework (in the data meta-model format) using the Data Push Port of the Data Aggregation and Workflow Notification Service (DAWN) or trigger workflows using the Workflow Initiation and Notification Request Port. The Data Access and Manipulation Adapter on the other hand provide a data pull and data manipulation interface to the fab using the same data meta-model. Integration of the IMPROVE framework into a fab only requires the implementation of these two adapter components.
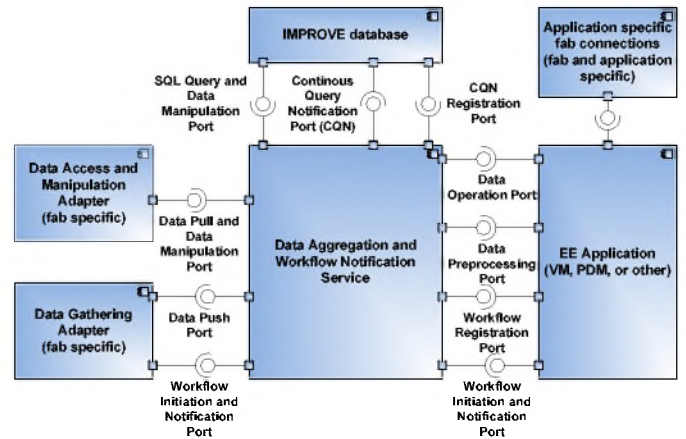

Figure 5. Overview of the main components of the IMPROVE framework and their connections to the fab framework.

EE Applications (VM, PdM) use workflows which are registered at startup using the Workflow Registration Port and are then triggered by DAWN via the Workflow Initiation and Notification Port. These workflows can access and change information in the IMPROVE framework by using the Data Operation Port of DAWN. Data operations are formulated in the IMPROVE data meta-model. DAWN handles these requests transparently by either translating them into Structured Query Language (SQL) queries on the database or forwarding them to the fab specific Data Access and Manipulation Adapter. In addition, EE applications can access a library of data preprocessing methods via the Data Preprocessing Port of DAWN. Workflows can be triggered by other workflows or by events like the elapse of time (for periodic tasks) or changes of data (for reactive tasks). Detecting changes to data in the IMPROVE data base may involve arbitrary queries. To handle

this efficiently a Continuous Query Notification (CQN) mechanism, available in several standard database systems like Oracle or SQLServer, can be used through the CQN ports.

IV. Software implementation of the framework

The software realization of the framework was accomplished using a service-oriented Java infrastructure (Fig. 6). At the core is a JBOSS application server which serves as a container for the modules and provides the communication backbone. The actual VM or PdM algorithms are plug-ins that can be written in MATLAB, R [23] or other languages.
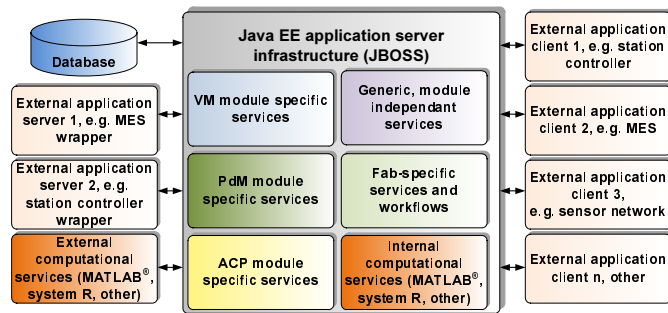

Figure 6. Software implementation of the framework and available services.

The main components of the framework are:

- JBOSS 7 Application Server Version [24]
- JBPM 5 Engine to execute production workflows [24]
- Oracle Database 11 [25]
- Listeners
- Clients
- Software modules realized as Enterprise Java Beans
- WEB-Service technology as the communication system

Production workflows tend to be more and more complex and JBPM allows to graphically organize these flows. JBPM is able to edit the flows and also provides an engine to execute workflows, according to input data, events and decisions along the execution path. The JBPM engine is integrated in the framework and API's allow to initiate such workflows. External programs (foreign applications) may register themselves as so called "listeners" to the framework to receive notifications upon occurring events. For example a maintenance management system like SAP [26] should be notified in case the IMPROVE PdM module generates a maintenance request. Client applications are active and connect to the IMPROVE modules and call their services. For example a station controller can initiate a dialogue to send the production data to the IMPROVE system and at the end of a production unit (a lot or a single wafer) it can initiate some VM or PdM calculation(s) and workflows.

V. Implementation and test of the framework in fab environments

The prototype of the IMPROVE framework was tested at three IC manufacturers' sites, each applying different configu-

rations of the system. At Infineon Technologies, testing was performed on a PdM module. Here, at first a MATLAB-based PdM algorithm, described as EE application in section III, needs to be registered via the data operation port with the DAWN component and via the CQN registration port with the database. A fab specific data gathering adapter was implemented to push new data from the internal FDC database via the data push port into the DAWN component and then via the SQL query and data manipulation port into IMPROVE database. Any data push triggers this PdM algorithm using first the continuous query notification port and then the data operation port. Fig. 7 displays how data are loaded into a value table within the memory of the framework.
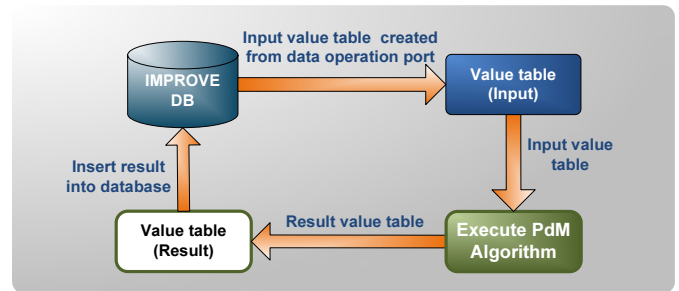

Figure 7. Data are transferred as value table between the IMPROVE database and the EE application (PdM algorithm at Infineon Technologies).

This value table is then converted into a MATLAB matrix and passed to the MATLAB-based PdM algorithm. After the computation of the PdM algorithm the calculated output is again converted from a MATLAB matrix into a value table in the memory. Finally, the outcome is passed back via the data operation port to the DAWN component and then stored back into the database again via the SQL query and data manipulation port. Now the calculated result can be assessed to trigger further actions. An example for these actions could be e-mail notifications to engineering staff, SAP maintenance requests or production equipment stops.

At Micron, it was necessary to extend the existing fab infrastructure in order to implement PdM and VM algorithms and the framework and implementation solutions developed in IMPROVE met this expectations. In particular, the following solutions were provided by implementation of the framework which enabled the application of the emerging APC methodologies like PdM and VM:

- Data bases were unified by merging data coming from different sources, e.g. from FDC and SPC data bases.

- Advanced algorithms can be executed which was not supported on traditional APC systems.

A Microsoft (MS) SQL Server [27] was chosen as database platform and a specific data loader was developed to merge data coming from two different FDC systems and two metrology measurements. For the Micron architecture it is planned to have a separate platform called "development platform" for the activities of coding, maintenance, algorithm update and test, validation of code, and deployment (Fig. 8).
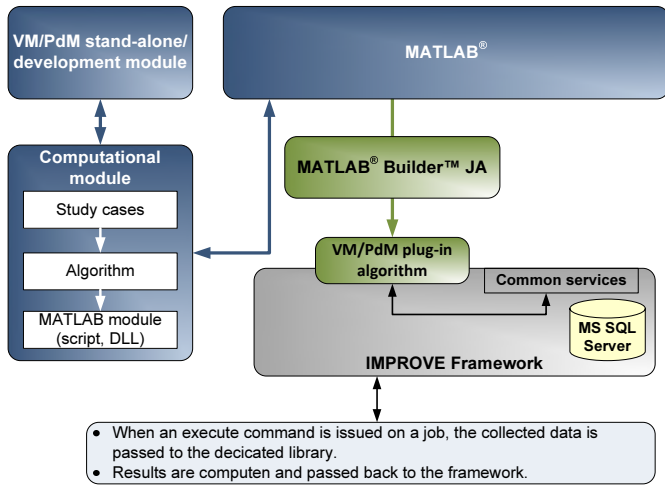
Figure 8. Framework implementation at Micron, Agrate, Italy.

In this context, the platform is used to compile the algorithms as Java package. With this specific platform, the integration of the algorithm passes through three phases:

- Compilation of the VM/PdM modules to Java using the MATLAB environment,
- Testing of the Java package for the dedicated Java application,
- Integration of the tested VM/PdM modules into the framework.

The installation of the demo/prototype framework has been completed and results from ongoing test and validation studies are available. The framework was tested on VM and PdM algorithms in dry-etch applications. These are the

- prediction of a required electrostatic chuck change (PdM),
- prediction of CD in a dry etch process (VM).

Performance tests of these applications were conducted on different VM/PdM algorithms comparing the codes in MATLAB and Java language. The results are summarized in Table 1 and the excellent performance of the algorithms and framework response vs. the targeted values is clearly visible.

TABLE I. Test results for VM and PdM applications at Micron, Agrate, Italy, applying the IMPROVE framework.

Applica-tion	Algo-rithm	Response time			
		Target	MATLAB	Java	Frame-work
PdM	Gamma Filter	minutes	< 7 sec	< 10 sec	< 39 sec
VM	Lasso	(periodi-cally)	< 1 sec	< 2 sec	< 1 sec
VM	Predic-tion	minutes	< 1 sec	< 2 sec	< 1 sec

The use case at austriamicrosystems' 200 mm wafer fab is to trigger a maintenance action to replace the source of a medium current implanter. The source is a consumable with a lifetime of one to two weeks depending on the process mix. To

perform maintenance at the latest possible time but before the source breaks can significantly reduce operation costs and tool downtime. A statistical model which describes the source degradation in terms of "remaining lifetime" was already developed within the IMPROVE project in close cooperation with Graz University of Technology. It is available as a plug-in programmed within the open source statistics package R.

Using adaptors the real time FDC data from the implanter can be pushed into the framework and to the PdM module where the projected remaining lifetime is calculated. This result is then handed over to the framework and transferred to the existing production SPC system where the data are visualized and used to trigger further actions. In our case an e-mail alarm automatically informed the responsible engineer 48 hours ahead of the projected breakdown and maintenance could be planned accordingly.

The whole configuration was successfully tested in the 24/7 production environment over two months achieving overall delay rates in the order of 25 seconds. This response time is completely satisfactory for PdM applications. Anyway, most of the delay time is generated by the provisional adaptors used to connect to the fab infrastructure. The response time of the framework including the PdM module is around three seconds (Fig. 9).
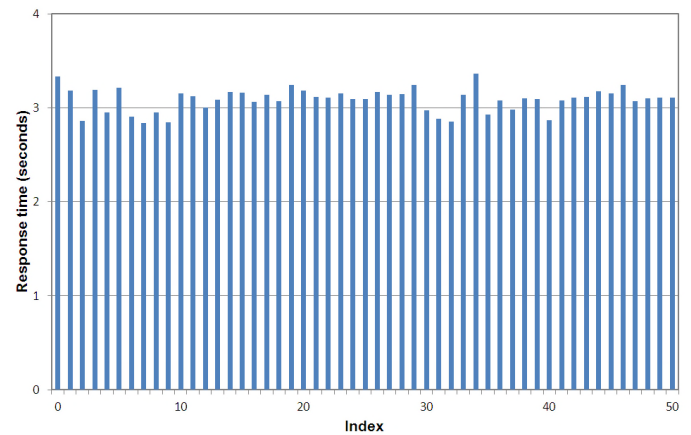

Figure 9. Response time as tested in production environment at austriamicrosystems.

VI. CONCLUSIONS

Within the ENIAC project "IMPROVE", a new framework for the implementation of VM and PdM as new control entities into existing fab-environments was developed, implemented and successfully tested. The development of the framework was targeted at the realization of a generic and reusable framework solution which is compliant to a wide range of standards and is compatible with different programming languages for algorithm realization. On a conceptual level, VM and PdM were defined as EE applications and the EES was modeled as component system in UML. To enable a highly modular structure and to be able to achieve configurability at run-time, a service-oriented methodology was selected for reifying of static

components and plug-ins. With this approach, the conceptual architecture can be mapped to the real-world fab infrastructures where the IMPROVE framework acts as a transparency layer between the different fab architectures and the EE application modules (VM, PdM). The software realization of the framework was accomplished applying a service-oriented Java infrastructure which provides capabilities for the storage of fab data and prediction results but also interfacing to other data sources and fab-applications. The software was implemented and tested on VM and PdM applications at three IC manufacturers' sites. The generic structure of the framework enabled specific implementations and extensions to the framework. Good performance of the framework system was obtained achieving response times below from 1 s to 60 s depending on the specific application and fab environment.

REFERENCES

[1] "Specification for Automated Process Control Systems Interface," SEMI Standard E133-0212 San Jose, CA: Semiconductor Equipment and Materials International (SEMI), 2012.

[2] A. A. Khan, J. R. Moyne, and D. M. Tilbury, "Virtual metrology and feedback control for semiconductor manufacturing processes using recursive partial least squares," Journal of Process Control, vol. 18, pp. 961-974, 2008.

[3] P. Kang, D. Kim, H.-j. Lee, S. Doh, and S. Cho, "Virtual metrology for run-to-run control in semiconductor manufacturing," Expert Systems with Applications, vol. 38, pp. 2508-2522, March 2011 2011.

[4] S. Pampuri, A. Schirru, G. Fazio, and G. De Nicolao, "Multilevel Lasso applied to virtual metrology in semiconductor manufacturing," 2011 IEEE International Conference on Automation Science and Engineering, 01 2011.

[5] H. Purwins, A. Nagi, B. Barak, U. Hockele, A. Kyek, B. Lenz, G. Pfeifer, and K. Weinzierl, "Regression methods for prediction of PECVD Silicon Nitride layer thickness," in Automation Science and Engineering (CASE), 2011 IEEE Conference on, 2011, pp. 387-392.

[6] ISMI Consensus Preventive and Predictive Maintenance Vision Guideline: Version 1.1, ISMI Technology Transfer # 06114819C-ENG.

[7] ISMI Predictive and Preventive Maintenance Equipment Implementation Guidelines, ISMI Technology Transfer #08064934A-ENG.

[8] G. Vachtsevanos, F. L. Lewis, M. Roemer und A. Hess, Intelligent Fault Diagnosis and Prognosis for Engineering Systems, New York: Wiley & Sons, 2006.

[9] S. J. Qin, G. Cherry, R. Good, J. Wang, and C. A. Harrison, "Semiconductor manufacturing process control and monitoring: A fab-wide framework," Journal of Process Control, vol. 16, pp. 179-191, 2006.

[10] A. A. Khan, J. R. Moyne, and D. M. Tilbury, "An Approach for Factory-Wide Control Utilizing Virtual Metrology," Semiconductor Manufacturing, IEEE Transactions on, vol. 20, pp. 364-375, 2007.

[11] F.-T. Cheng, Cheng, H.-C. Huang, and C.-A. Kao, "Developing an Automatic Virtual Metrology System," Automation Science and Engineering, IEEE Transactions on, vol. 9, pp. 181-188, 2012.

[12] F.-T. Cheng, J. Y.-C. Chang, C.-A. Kao, Y.-L. Chen, and J.-L. Peng, "Configuring AVM as a MES component," in Advanced Semiconductor Manufacturing Conference (ASMC), 2010 IEEE/SEMI, 2010, pp. 226-231.

[13] F.-T. Cheng, G.-W. Huang, C.-H. Chen, and M.-H. Hung, "A generic embedded device for retrieving and transmitting information of various customized applications," in Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, 2004, pp. 978-983 Vol.1.

[14] Y.-T. Huang, H.-C. Huang, F.-T. Cheng, T.-S. Liao, and F.-C. Chang, "Automatic virtual metrology system design and implementation," in Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on, 2008, pp. 223-229.

[15] H.-C. Huang, Y. Chuan, F.-T. Cheng, and J.-M. Jian, "Development of a Generic Virtual Metrology Framework," in Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on, 2007, pp. 282-287.

[16] Y.-C. Su, W.-H. Tsai, F.-T. Cheng, and W.-M. Wu, "Development of a dual-stage virtual metrology architecture for TFT-LCD manufacturing," in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 2008, pp. 3630-3635.

[17] Y.-C. Su, F.-T. Cheng, M.-H. Hung, and H.-C. Huang, "Intelligent prognostics system design and implementation," Semiconductor Manufacturing, IEEE Transactions on, vol. 19, pp. 195-207, 2006.

[18] Y.-C. Su, F.-T. Cheng, M.-H. Hung, Y.-C. Lin, and R.-C. Lin, "Design and implementation of an intelligent prognostics system," in Automation Science and Engineering, 2005. IEEE International Conference on, 2005, pp. 273-278.

[19] J. C.-H. Pan and D. H. E. Tai, "Implementing virtual metrology for in-line quality control in semiconductor manufacturing," International Journal of Systems Science, vol. 40, pp. 461 - 470, 2009.

[20] MATLAB, the language of technical computing, The MathWorks, Inc., http://www.mathworks.com/.

[21] "Equipment Engineering Capabilities (EEC) Guidelines (Phase 2.5)," International SEMATECH & JEITA/Selete, 2002. http://ismi.sematech.org/emanufacturing/docs/eecguidebook.pdf.

[22] Object Management Group. Unified Modeling Language v2.4.1, Superstructure. OMG, 2011, http://www.omg.org/spec/UML/2.4.1.

[23] The R Project for Statistical Computing, http://www.r-project.org/.

[24] JBoss Application Server 7, JBPM 5 Engine, http://www.jboss.org.

[25] Oracle Database 11, http://www.oracle.com/.

[26] SAP Business Management Software Solutions, Applications and Services, http://www.sap.de.

[27] Microsoft-SQL Server, http://www.microsoft.com.