14

CIM Framework architecture and application models

J. S. Hawker
Motorola Semiconductor Products Sector
2150 Woodward Street
Austin, Texas 78744, USA
phone: 1.512.373.3160
e-mail: RP5454@email.sps.mot.com

Abstract

The SEMATECH CIM (computer integrated manufacturing) Framework defines an industry standard framework for semiconductor manufacturing information and execution systems (MIES). The CIM Framework architecture defines an object-oriented software component architecture building on CORBA and CORBA services specifications from the Object Management Group. Using this architecture, the CIM Framework Specification defines an application mode that specifies the interfaces and behavior of functional software components for MIES. There are component definitions for machine control, material movement, material management, process specification management, advanced process control, factory operations, dispatching, and labor management. Standard interface and behavior specifications for these MIES functions enable manufacturers to assemble and evolve manufacturing systems using components from different suppliers, in-house development or legacy systems. In early 1998, SEMI Standards, the semiconductor industry standards organization, will ballot the CIM Framework for adoption as provisional standards. This paper provides an overview of the CIM Framework architecture and illustrates the form and content of the application models in the CIM Framework.

Keywords

Computer Integrated Manufacturing, CIM, Manufacturing Information Systems, Manufacturing Execution Systems, Component, CORBA.

J. J. Mills et al. (eds.), *Information Infrastructure Systems for Manufacturing II*

1 INTRODUCTION

Flexible manufacturing information systems are essential to manufacturing agility and competitiveness (Holland *et.al.*, 1996; Hawker *et.al.*, 1996a). Manufacturers must be able to modify, upgrade and enhance their manufacturing information and execution systems (MIES) with new functions and technologies that meet their changing business needs (SIA, 1997). However, most MIES available and in use today are monolithic systems where it is very expensive, in time and money, to integrate new functions or to upgrade technologies (Aardal, 1994). There are significant, often unacceptable, time and cost barriers to building new systems and enhancing current systems, resulting in the risk of not having the MIES that enable competitive manufacturing.

The SEMATECH CIM Framework addresses the needs for semiconductor manufacturers to, over time, assemble MIES software components from multiple suppliers into an integrated MIES system that meets their changing business needs.

SEMATECH CIM Framework

SEMATECH, a research and development consortium of semiconductor manufacturers, has just completed a strategic program to define a framework for integrating MIES applications. This framework, the CIM Framework defines a standard application model of MIES applications (Hawker, 1996; Doscher, 1997; Doscher, 1998). The application model defines standard interfaces and behaviors for parts (components) of applications that are common across applications. It leverages distributed computing technology standards from the Object Management Group (OMG) (OMG 1995, 1996, 1997) to enable integration of the applications. Manufacturers that implement MIES systems based on the CIM Framework can incrementally build MIES by integrating applications from multiple suppliers, and they can upgrade these systems with evolving distributed computing technologies conformant with OMG standards and with application component functions and technologies conformant with CIM Framework standards. The CIM Framework defines a standard MIES software component architecture and application component interfaces. It enables semiconductor manufacturers to significantly reduce the cost and time required to build, modify and enhance their MIES in response to changing business needs.

Industry consensus

SEMATECH led an industry consensus process to develop the CIM Framework software component architecture and component interfaces. MIES suppliers and users in the semiconductor industry cooperated to develop proposals for twelve provisional standards which are now in review in SEMI Standards, the standards organization for the semiconductor industry. Many suppliers are developing and marketing MIES products conformant with the CIM Framework. SEMATECH also co-chairs the OMG Manufacturing Domain Task Force (MfgDTF) and is involved in other OMG task forces which are adopting CIM Framework architecture and application components (OMG 1997a, 1998).

CIM Framework standards

The twelve CIM Framework standards ballots are based on the SEMATECH CIM Framework Specification (Doscher, 1998) and the CIM Framework Architecture Guide (Hawker, 1996; Doscher, 1997). They are an integrated system of software standards, but are separated so they can evolve in SEMI in response to industry priorities. The twelve standard proposals include three documents to describe goals, architecture and global definitions, plus an application model made up of nine groups of component interface definitions, as follows (SEMI 1998):

- CIM Framework organization and introduction
- CIM Framework architecture
- CIM Framework global definitions, events and abstract base interfaces
- Application component interface definitions
  - Factory services group
  - Factory management group
  - Material management group
  - Material movement group
  - Process specification group
  - Machine control group
  - Advanced process control group
  - Scheduling group
  - Factory labor group

SEMI has three international task forces (Europe, Japan and North America) cooperating toward adoption of the CIM Framework standards. Further information is available at the following worldwide web site:

http://semi-tf-cim-framework.ipa.fhg.de

Paper organization

This paper provides an overview of the CIM Framework software architecture and describes the Product Management application component to illustrate the form and content of the application models in the CIM Framework. Section 2 describes the CIM Framework application component architecture, its basis in OMG standards and the specification methodology the CIM Framework uses. Section 3 describes the scope and structure of the CIM Framework application models. Section 4 provides a view of the Product Management component specification excerpted from the SEMATECH CIM Framework Specification. Section 5 is a conclusion.

2.1 Architecture Objectives

The CIM Framework software architecture is designed to enable integration of MIES applications when those applications are integrated over time from multiple third-party suppliers, internal development and legacy systems. The specific objectives of the architecture are to enable the following capabilities:

- *Interoperability* – applications can cooperate by exchanging data, providing services (client/server method invocation), publishing service exceptions, and publishing and subscribing to events.
- *Substitutability* – MIES implementers can replace an application from one supplier or source with a functionally equivalent application (conformant to standard interface and behavior) from another source without impacting the other applications.
- *Extendibility* – MIES implementers can extend or specialize existing systems and can add new applications, application components or objects, and these extensions can fully use all existing system functionality.
- *Flexibility* – MIES implementers can compose and configure applications in a variety of ways that meet specific needs.
- *Reuse* – leveraging the benefits of substitutability, extendibility and flexibility, MIES implementers can base new systems on the design and implementation of standard components of applications, enabling solutions to be developed more quickly, at lower cost, and with higher quality.

2.2 CIM Framework Component architecture

The CIM Framework architecture is a layered system, as in the overview of Figure 1 and the detail of Figure 2. The architecture enables distributed, object-oriented applications assembled from common software components to interoperate as a single, integrated system.

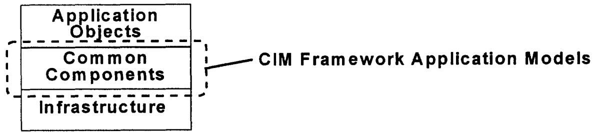

Figure 1 CIM Framework architecture layers.
(Used with permission of SEMATECH, Inc.)

The integration infrastructure (the bottom layer of Figure 1) is based on CORBA, CORBAservices and CORBAfacilities (CORBA: Common Object Request Broker Architecture) specifications in the Object Management Architecture (OMA) from the OMG (OMG 1995, 1996, 1997). CORBA, CORBAservices and CORBAfacilities define standard services for distributed object communications, persistence, transactions, name services, etc. This defines the infrastructure "plumbing" for application integration.

On top of the infrastructure, the CIM Framework architecture defines common application components. Components are software building blocks – "chunks" of functionality that make up software applications. Components are more coarse-grained than objects (components are defined in terms of closely-related objects) but more fine-grained than typical MIES applications. The CIM Framework common components layer defines standard models for application components that are common across MIES applications, such as definitions for a Machine Management Component (including machine resources, sensors, process capabilities, and relations to material and recipes), a Product Management Component (including product material, lots, and relations to product and process specifications) and a Person Management Component (including persons, qualifications and relations to skills and skill requirements). This common application model, defined in terms of common software components, is the *framework* for building integrated MIES applications, that is, the basic MIES application models on which all MIES applications are based. The common components do not define complete MIES application models, only the core application models that are common across MIES software applications.

The application objects layer of the CIM Framework architecture provides additional functionality, extending the common components to make a complete MIES. This layer, which is identified but not specified, enables MIES suppliers and users to define product-specific and site-specific application objects and components that use and extend the CIM Framework common components to implement MIES functions that meet business needs.

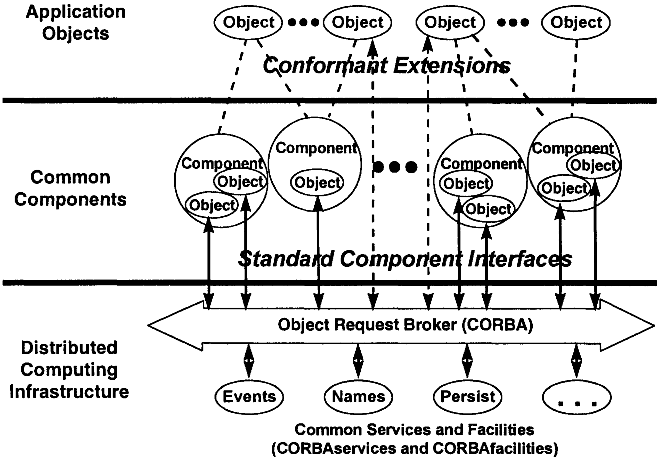

Figure 2 CIM Framework component architecture.
(Used with permission of SEMATECH, Inc.)

A given MIES application, then, implements some common components and application objects and interoperates (via the infrastructure) with other common components and application objects implemented in other MIES applications. Figure 2 illustrates the relationship between the infrastructure, common components and application objects. The collection of interoperating MIES applications provide a complete, integrated, MIES solution. By basing the solution on standard infrastructure services and standard application models for common MIES components, the agile manufacturer can cost-effectively build, modify and enhance the MIES.

2.3 Component specification methodology

The goal of the CIM Framework is to specify application components that MIES implementers can assemble into a tightly-integrated system. The high expectation for tight integration demands functionally rich component specifications. It is not enough to simply specify the syntax of component interfaces. The CIM Framework also specifies interface semantics and component behavior.

The CIM Framework uses the following modeling methods to specify components:

- component relationship models showing interaction between "medium-grained" components (larger than an object, smaller than an application);
- component information models showing object interfaces and relationships in the form of OMT (Object Modeling Technique) diagrams (Rumbaugh, *et.al.* 1991);
- object interface definitions using OMG Interface Definition Language (IDL);
- published and subscribed events using an extension to OMG IDL;
- component interaction diagrams showing scenarios that trace messages and events between components;
- state transition diagrams (as Harel state charts) and state definition tables.

These modeling methods go far toward specifying components that MIES implementers can "plug-and-play" into integrated systems. SEMATECH is also working in the OMG Business Objects Domain Task Force to define and standardize additional methods for even richer semantic models, including the specification of method pre-conditions and post-conditions, roles, rules, and dependencies (OMG, 1998).

3 Application models

The CIM Framework specifies application components for Manufacturing Information and Execution Systems (MIES). MIES perform factory operations functions, in the context of Enterprise Information and Control Systems and systems that automate material processing, storage and movement. Figure 3 shows the MIES functional groups in the CIM Framework.

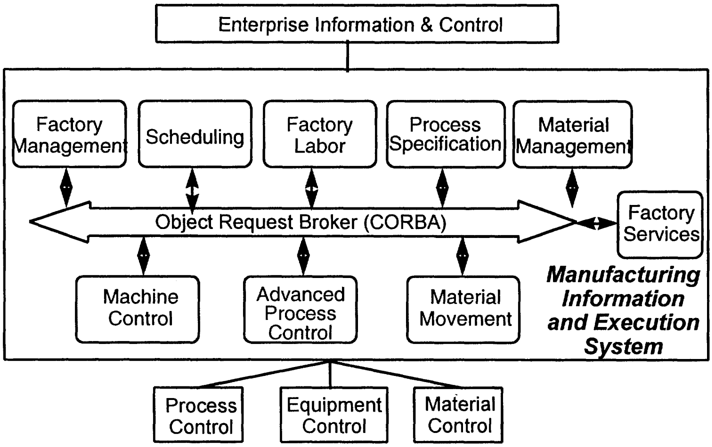

Figure 3 Functional groups for MIES.

Each functional group defines a collection of related application components. Table 1 lists the CIM Framework components in each functional group. The functional groups are a convenient mechanism to organize the CIM Framework components; they are not rigid partitions and suppliers can deliver applications that span functional groups or that implement only some of the components of a group. In contrast, the component is the smallest-grained entity that suppliers can deliver. A supplier must implement all the interfaces and behaviors of a component in order to claim conformance to that component specification.

The value and power of the CIM Framework is in the industry standard application model which specifies medium-grained components common to MIES applications. The SEMATECH CIM Framework Specification Version 2.0 (Doscher, 1998) has almost 300 pages of detailed component models specified using the methodology of Section 2.3. Section 4 presents a portion of the CIM Framework Product Management component to illustrate the style and detail of the specification. By developing industry-wide consensus on these component definitions, SEMATECH has enabled manufacturers to quickly and cost-effectively build, modify and enhance MIES by assembling standards-conformant components from multiple suppliers.

Table 1 CIM Framework application components

Factory Services
Document Management
Version Management
History Management
Event Broker

Factory Management
Factory
Product Release
Factory Operations

Factory Labor
Person Management
Skill Management

Machine Control
Machine Management
Recipe Management
Resource Tracking

Material Management
Product Management
Durable Management
Consumable Management
Inventory Region
Product Specification
Bill of Material

Material Movement
Material Movement

Advanced Process Control
Plugin Management
Plugin Execution
Control Management
Control Execution
Control Database
Data Collection Plan

Process Specification Management
Process Specification
Process Capability

Schedule Management
Dispatching

4 PRODUCT MANAGEMENT COMPONENT

This section presents an excerpt from the SEMATECH CIM Framework Specification Version 2.0 (Doscher, 1998) (used with permission of SEMATECH, Inc.). The component is specified using the methodology of Section 2.3. All capitalized and run-on words (such as ProductManager) refer to software entities specified in the CIM Framework. Portions in Courier font (such as `MaterialLocation getMaterialLocation`) is OMG IDL that can be compiled.

Section 4.1 contains the information model (in OMT notation) for the Product Management Component. Since the focus is on integration, the information model describes only the interfaces to and relations between component objects, not the implementations. A given implementation may have a different object structure that implements the specified interfaces. The terminology is consistent with OMG IDL. Section 4.2 presents a portion of the IDL for the Product object's interface. Section 4.3 presents the dynamic model (the states and state transitions) for Products as a Harel state chart in Figure 5. Table 2 provides a description of some of the states and the query method for asking a Product its state, and Table 3 provides a tabular form of some of the state chart to document the triggers and actions of state transitions. Only a portion of the IDL and state tables are presented in order to provide an introduction without too much detail. Refer to the CIM Framework Specification for complete models. In section 4.1 through 4.3, the style and content is as found in the CIM Framework Specification.

4.1 Description and information model of Product Management

The Product Management Component provides the representation for various types of product to be viewed at the factory level. Behavior concerning product location, product aggregation, and product progress are services provided by the interfaces in this component. Categories of product are wafers, die, and packages. The aggregation of products is represented as a lot. A ProductManager provides lifecycle management and coordination of behavior for objects representing product and lot.

ProcessGroup represents a material aggregate used for processing in a machine. Units of Product or parts are known to be in a process together when they are members of the same ProcessGroup. This interface should be considered for customization to the practices of a particular Factory.

Figure 4 contains the Product Management component informational model.

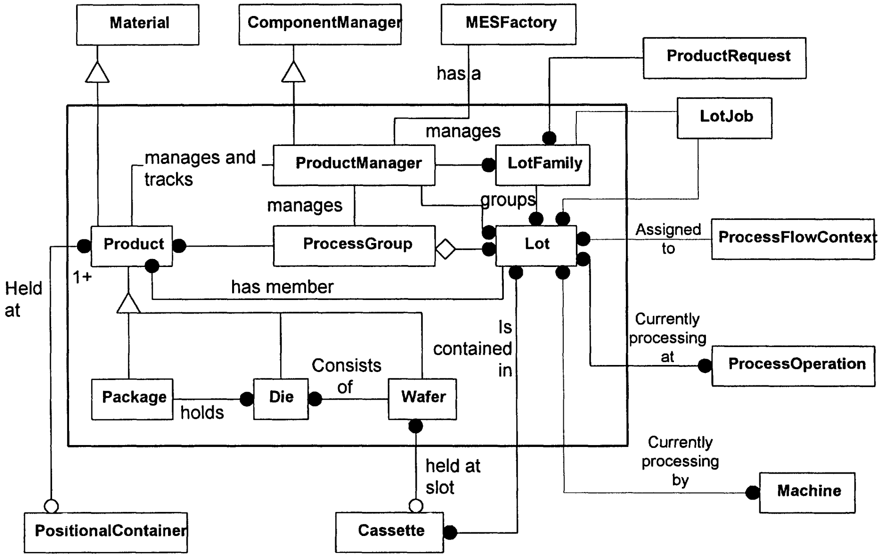

Figure 4 Product management component information model. (Used with permission of SEMATECH, Inc.)

This component defines the following interfaces:

- ProductManager
- Package
- Product
- Lot
- Wafer
- LotFamily
- Die
- ProcessGroup

A portion of the Product interface definition follows.

4.2 Product interface definition (portion)

Interface: Product

Inherited Interface: Material

Description: The Product interface provides for the representation of any material that undergoes processing in a Factory. Product, in the semiconductor industry, includes any unit which is intended to become a functional semiconductor device including functional engineering devices. Associated with each Product is a specification for building it, a flow (or route) created from that specification, a production history, and a possible position in a positional container.

Exceptions:

/* This signal is raised when an operation assumes that a Product is in a positional container when it is not. */

```
exception ProductNotInPositionalContainerSignal
{Product aProduct;};
```

Published Events: None.

Provided Services:

/* Set and get the location of the Product. */

```
MaterialLocation getMaterialLocation ( )
       raises (FrameworkErrorSignal);
void setMaterialLocation
       (in MaterialLocation aMaterialLocation)
       raises
(FrameworkErrorSignal,SetValueOutOfRangeSignal);
```

/* Returns the Lot of which the Product is a member. The set method has no public interface. */

```
Lot getLot( ) raises (FrameworkErrorSignal);
```

/* Set the Product's status to the state indicated. */

```
void makeNotAllocated ( )
       raises
(FrameworkErrorSignal,InvalidStateTransitionSignal);
```

/* Answer whether the status of the Product is that indicated. */

```
boolean isCreated ( ) raises (FrameworkErrorSignal);
```

4.3 Product dynamic model

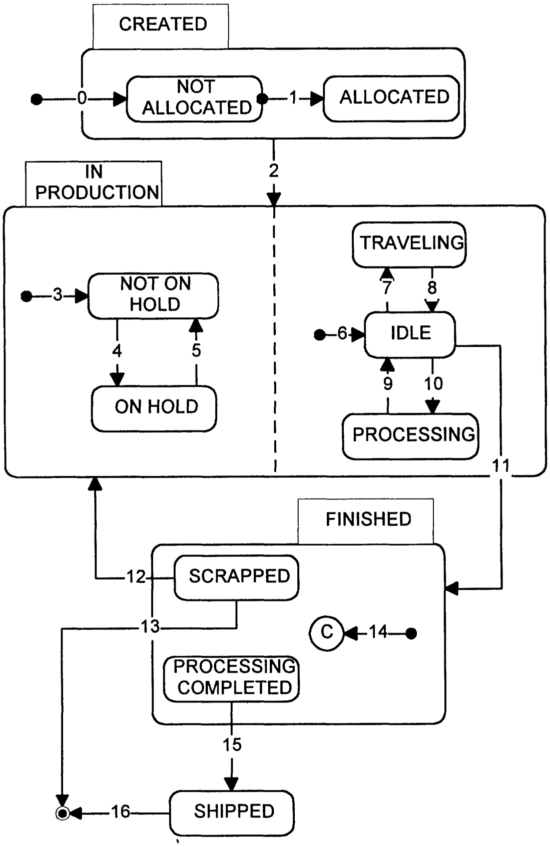

Figure 5 Product dynamic model.
(Used with permission of SEMATECH, Inc.)

Table 2 Product state definition and query (portion) (Used with permission of SEMATECH, Inc.)

State	*Definitions*	*Query for State via*
Created and Not Allocated	The Product is created but not allocated to a Lot.	boolean isNotAllocated ( ); sent to the instance of Product.
Created and Allocated	The Product is created and has been allocated to a Lot. The corresponding Lot state will be Created.	boolean isAllocated ( ); sent to the instance of Product.

Table 3 Product state transitions (portion) (Used with permission of SEMATECH, Inc.)

#	*Current State*	*Triggers*	*New State*	*Action*	*Comment*
0	non-existent	Wafer createWaferNamed (in string identifier);, Die createDieNamed(in string identifier);, Package createPackageNamed(in string identifier);, sent to instance of ProductManager.	Not Allocated	Objects of Specific Product type are created	extensions for new Product types can be added
1	Not Allocated	Lot createLotUsing (in ProductRequest aProductRequest);, Lot createLotUsing _fromProducts (in ProductRequest aProductRequest, in ProductSequence aProductSequence);, Lot createLotUsing _withIdentifiers(...); sent to ProductManager	Allocated	The instance of Product is associated with an instance of Lot.	The instance of Lot that the Product is associated with is not yet placed into Production but is ready for Release (ready for going into production).

5 CONCLUSION

The SEMATECH CIM Framework is important to the semiconductor manufacturing industry. It addresses growing business pressures to provide MIES with the flexibility and functionality that supports agile, competitive manufacturing. The CIM Framework is real. All major semiconductor MIES suppliers and many material control and process automation suppliers are working with many of the largest semiconductor manufacturers to review and adopt CIM Framework standards in SEMI Standards. Many of the suppliers have products which are heavily influenced by early versions of the CIM Framework, and most have committed to make their products CIM Framework conformant. In 1998 and beyond, semiconductor manufacturers will be able to replace obsolete, inflexible MIES with open systems built from standards-conformant components.

6 REFERENCES

Aardal, M. (1994) *Computer Integrated Manufacturing (CIM) Benchmarking Review.* Technology Transfer #94102599A-XFR, SEMATECH, Austin, Texas.

Doscher, D. (1997) *Computer Integrated Manufacturing (CIM) Framework Architecture Concepts, Principles and Guidelines, Version 1.0.* Technology Transfer #97103379A-ENG, SEMATECH, Austin, Texas.

Doscher, D. (Editor) (1998) *Computer Integrated Manufacturing (CIM) Framework Specification Version 2.0.* Technology Transfer #93061697J-ENG, SEMATECH, Austin, Texas.

Hawker, J.S. (1996) *SEMATECH Computer Integrated Manufacturing (CIM) Framework Architecture Concepts, Principles and Guidelines, Version 0.7.* Technology Transfer #96123214A-ENG, SEMATECH, Austin, Texas.

Hawker, J.S., Waskiewicz, F.W. (1996a), Agility enabled by the SEMATECH CIM Framework. *Plug and Play Software for Agile Manufacturing* in SPIE's Photonics East 1996 Conference, 18 November 1996.

Holland, S., Mills, J.J., Christopher, N., *et. al.* (1997) Adaptive, Responsive Information Systems: A Next-Generation Manufacturing Imperative, in *Next-Generation Manufacturing; a Framework for Action.* NGM Project Office, 125 Goodman Drive, Bethlehem, PA 18015-3715.

Object Management Group (1995) *Object Management Architecture Guide, Revision 3.0.* Object Management Group, Framingham, Massachusetts. ftp://ftp.omg.org/pub/docs/ab/97-05-05.pdf.

Object Management Group (1996) *CORBAservices: Common Object Services Specification.* Object Management Group, Framingham, Massachusetts.

Object Management Group (1997) *Common Object Request Broker: Architecture and Specification*, Revision 2.1. Object Management Group, Framingham, Massachusetts, ftp://ftp.omg.org/pub/docs/formal/97-10-01.pdf.

Object Management Group (1997a) *Manufacturing Domain Task Force RFI-3: Manufacturing Execution Systems.* OMG Document mfg/97-11-01, Object Management Group, Framingham, Massachusetts.

Object Management Group (1998) *Combined Business Object Facility Interoperability Specification.* OMG Business Object Domain Task Force BODTF-RFP1 Submission, document bom/98-0203, Object Management Group, Framingham, Massachusetts.

Rumbaugh, J. Blaha, M. *et.al.* (1991) *Object-Oriented Modeling and Design.* Prentice-Hall, Englewood Cliffs, New Jersey.

SEMI (1998) *CIM Framework Pre-Ballots.* SEMI Draft Documents 2817—2828, CIM Framework Standards Task Force, Semiconductor Equipment and Materials International, Mountain View, California.

SIA (1997) *National Technology Roadmap for Semiconductors.* Semiconductor Industry Association, San Jose, California, 114—117, 126—127.

7 AUTHOR BIOGRAPHY

J. S. Hawker received B.S. and M.S. degrees in electrical engineering from Texas Tech University of Lubbock, Texas in 1981 and 1982, respectively, and the Ph.D. degree in electrical engineering from Lehigh University of Bethlehem, Pennsylvania in 1990. He has worked for AT&T Bell Laboratories and Honeywell, developing large-scale signal processing and control systems for military and manufacturing applications. He was at SEMATECH, on assignment from Motorola, from 1995 to 1997, leading the CIM Framework program defining industry standards for manufacturing information and execution systems. He is now back at Motorola, developing next-generation manufacturing system architectures and technologies.