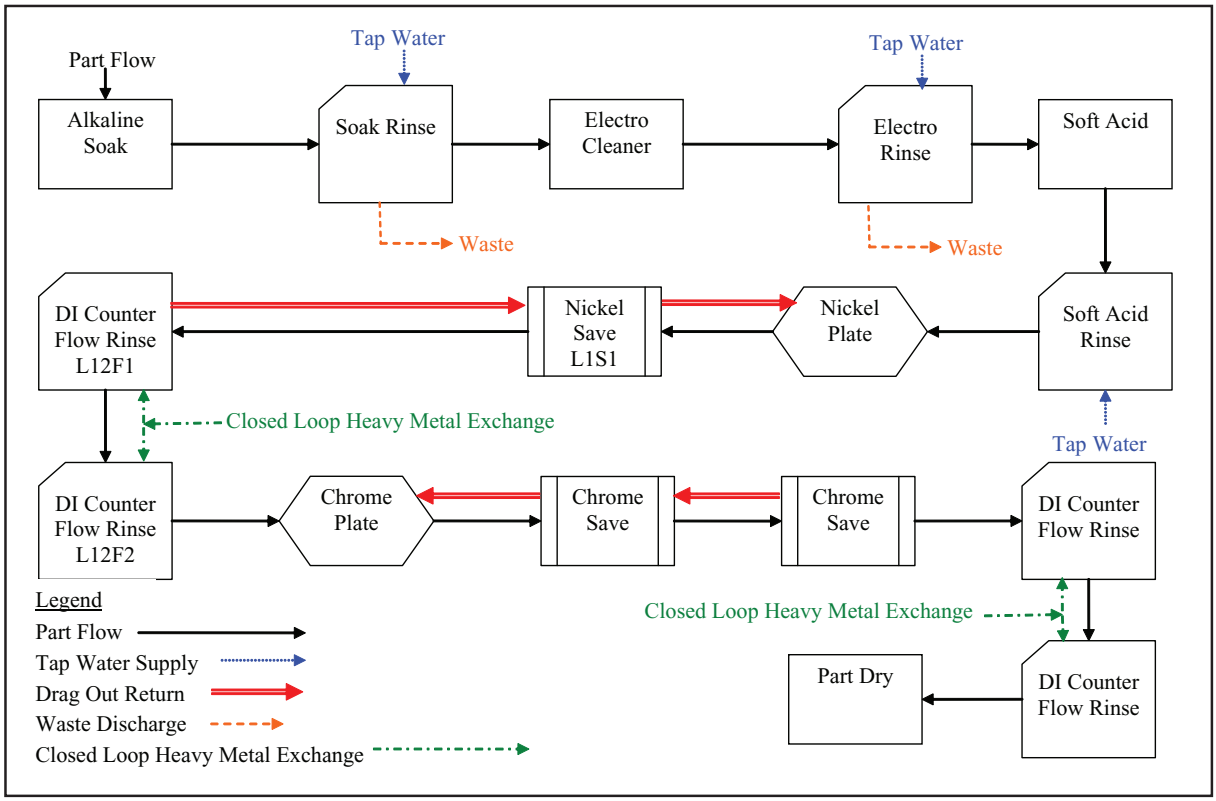
United States
Environmental Protection
Agency

EPA/600/R-07/067
April 2007

Final Report for Verification of the Metal Finishing Facility Pollution Prevention Tool (MFFPPT)



EPA/600/R-07/067
April 2007

FINAL REPORT FOR VERIFICATION OF THE

METAL FINISHING FACILITY POLLUTION PREVENTION TOOL (MFFPPT)

by

William Martin Barrett Jr, Ph.D., P.E.[1]
Jun Yang[2]
Svetlana Strunjas[3]

[1]U.S. Environmental Protection Agency
Office of Research and Development
National Risk Management Research Laboratory
26 West Martin Luther King Drive
Cincinnati, Ohio 45268

[2]Trinity Consultants
12770 Merit Drive
Suite 900
Dallas, TX 75251

[3]University of Cincinnati
Department of Computer Engineering
College of Engineering
University of Cincinnati
Cincinnati, Ohio

EPA REVIEW NOTICE

This report has been peer and administratively reviewed by the U.S. Environmental Protection Agency and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

FOREWORD

The U.S. Environmental Protection Agency (EPA) is charged by Congress with protecting the Nation's land, air, and water resources. Under a mandate of national environmental laws, the agency strives to formulate and implement actions leading to a compatible balance between human activities and the ability of natural systems to support and nurture life. To meet this mandate, EPA's research program is providing data and technical support for solving environmental problems today and building a science knowledge base necessary to manage our ecological resources wisely, understand how pollutants affect our health, and prevent or reduce environmental risks in the future.

The National Risk Management Research Laboratory (NRMRL) is the agency's center for investigation of technological and management approaches for preventing and reducing risks from pollution that threaten human health and the environment. The focus of the laboratory's research program is on methods and their cost-effectiveness for prevention and control of pollution to air, land, water, and subsurface resources; protection of water quality in public water systems; remediation of contaminated sites, sediments, and ground water; prevention and control of indoor air pollution; and restoration of ecosystems. NRMRL collaborates with both public and private sector partners to foster technologies that reduce the cost of compliance and to anticipate emerging problems. NRMRL's research provides solutions to environmental problems by: developing and promoting technologies that protect and improve the environment; advancing scientific and engineering information to support regulatory and policy decisions; and providing the technical support and information transfer to ensure implementation of environmental regulations and strategies at the national, state, and community levels.

This publication has been produced as part of the laboratory's strategic long-term research plan. It is published and made available by EPA's Office of Research and Development to assist the user community and to link researchers with their clients.

Sally Gutierrez, Director
National Risk Management Research Laboratory

(This page intentionally left blank.)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract

The United States Environmental Protection Agency (USEPA) has prepared a computer process simulation package for the metal finishing industry that enables users to predict process outputs based upon process inputs and other operating conditions. This report documents the development of the process simulations package, calculations methodologies used in the simulation, verification of the model, and its ability to simulate metal finishing processes.

The process simulation package has been built using Microsoft's Visual C++ Version 8, also known as Microsoft Visual C++ 2005 and the C++ extensions to the Common Languange Infrastructure (C++/CLI) and the Microsoft .Net Framework Version 2.0. Additionally, the process simulation package supports the CAPE-OPEN computer-aided processes engineering (CAPE) open interface standards. Through the use of the CAPE-OPEN standards, the process simulator has been shown to interoperate with third-party process simulation components, and can be used as a general chemical process simulation package.

The metal finishing process modeling components (PMCs) were developed as an add-in package for the process simulator. The add-in package includes models of a generalized plating tank, rinse tank, and models of alkaline cleaning, acid cleaning, and electrocleaning processes. The model conducts material for each process operation modeled and conducts other calculations as needed to model the changes to the part and process stream. The calculations performed include aqueous ionic speciation, plate thickness, and air emissions from each of the process tanks.

Model verification involved documenting a local plating facility's plating process and collecting samples of the plating line processes to compare the results of the model with actually observed conditions present in the plating facility. A process simulation of the plating facility

was performed using the available process flow and composition data, and the results of this simulation were compared with the conditions actually observed within the facility. Comparison of the MFFPPT model outputs with the observed concentration indicate that the available level of detail known about a typical plating system is lacking. Additionally, time-dependent processes such as static rinses were not well modeled by the steady state approach of this package. Use of a process simulation-based tool such as this would require more in depth process data, such as detailed disclosure of plating solution compositions and more accurate flow measuring devices within the facility, than is typically available in a small plating facility.

1.0 Introduction

The metal finishing pollution prevention tool (MFFPPT) is a computer-based simulation of metal finishing facilities that is intended to allow the facility to evaluate the effect of process modifications on pollution generation within the facility. MFFPPT consists of two basic parts, a process simulation package and an add-in component package that specializes the process simulation to include unit operations found in a metal finishing facility. This document describes the architecture of the program, the mathematical models used to calculate the performance of the metal finishing operations, and the results of a verification study.

1.1 Overview of the Project

The Metal Finishing Facility Pollution Prevention Tool (MFFPPT) is an expansion of the scope of the Metal Finishing Facility Risk Screening Tool (MFFRST) (USEPA 2001) developed by NRMRL and the National Center for Environmental Assessment (NCEA). MFFRST evaluated air emissions from typical metal plating process line ups, estimated the concentration of contaminants that workers were exposed to, and provided an estimate of health risk to the workers as a result of that exposure.

Upon completion of MFFRST, it was determined that the scope of the tool should be expanded to include solid and liquid wastes generated in metal plating processes, which was consistent with updates to the National Metal Finishing Environmental Research and Development Plan (Pacific Environmental Services 2000). Whereas health risk screening was a natural endpoint for air emissions due to their direct impact on individuals living and working near metal finishing facilities, solid and liquid wastes generated by a metal finishing facility are typically managed through the Clean Water Act (CWA) and Resource Conservation and

Recovery Act (RCRA). Management of these wastes under these programs focused on end-of-pipe treatment of liquid wastes or landfilling of solid wastes to reduce environmental exposure and consequent risks associated with environmental exposure to the wastes. As a result, opportunities for reduction of risks associated with these waste streams largely lies within the facility. For this reason, MFFPPT focused on process simulation as a means of identifying pollution prevention techniques that can be applied to metal finishing processes.

Central to pollution prevention is the development of a process model that can be used to understand the effect of process variables on the emissions from the metal finishing processes. In general, metal finishing facilities consist of chemical processes that can be modeled using well-established chemical engineering practices. MFFPPT has been developed to simulate unit operations commonly found in metal finishing facilities. The purpose of this simulation is to enable the facility to evaluate the effect of process changes, such as operating conditions, on the amount of waste generated. Further, the simulation provides mechanisms to calculate the speciation of various ions in solution, and can potentially aid in the development of environmentally cleaner metal finishing processes through the greater insight that can be gained by evaluating the details of the process such as the mechanism of metal deposition on the part surface.

As part of this project, it was necessary to develop a process simulation package that was capable of modeling the metal finishing process. It soon became apparent that a core functionality was required for the objects being simulated. Research on process simulation architecture led to the CAPE-OPEN project, a project founded through European Union grants in the 1990s and supported by major chemical companies such as the Dow Chemical Company, BP, BASF, the Institut Français du Pétrole (IFP), Shell Global Solutions International, B.V., and

TOTAL. The CAPE-OPEN project has developed a set of interfaces that allow engineers to create process model object and use them in modeling packages that support the standards.

The CAPE-OPEN interfaces are built upon two different information technology standard sets, one of which is Microsoft's Component Object Model (COM), which is the basis for ActiveX controls (controls used in web-based applications). As part of this research project, the EPA has played a central role in expanding the CAPE-OPEN model to Microsoft's newer object model, the Microsoft .NET Framework. .NET simplifies the development of web-based software, known as web services, and enables various distributed computing models that can be used to share data within and between organizations. Use of an internet/distributed computing-based object model for process modeling is expected to enable the use of process model components as part of life-cycle-based products evaluations such as life cycle assessment (LCA) and supply chain management.

As indicated above, the CAPE-OPEN programs (CAPE-OPEN 2000) have specified a set of interfaces that can be exposed by process modeling components (PMCs) and used by the process modeling environment (PME) to enable CAPE-OPEN compliant PMEs to utilize PMCs created by third parties, including unit operations, and thermodynamic property models. Another benefit of the common interfaces developed by the CAPE-OPEN programs is the ability to use these interfaces as part of a common application framework for a process modeling application programming interface (API) that will allow sharing of process information across application domains. In other words, through the exposed interfaces, users and application developers have access to the data contained within the simulation and can use that data to make business and operational decisions based on the simulation results. The application described here is intended to provide a starting point for building such an API.

Quality assurance activities for this program were originally stated in a QAPP approved September 3, 2002, and this QAPP was revised on July 10, 2003 and February 15, 2006. A QAPP was approved for the model verification study documented in this report on February 7, 2006.

1.2 Scope of Work

The EPA developed a process simulation package and metal finishing add-in modules capable of determining the quantity of wastes generated by metal plating operations using a combination of in-house and external resources. In addition, EPA conducted a verification of the program by sampling metal finishing plating lines at a plating facility and comparing the operating conditions of the plating tanks with the conditions of the tanks predicted by the model. The following is a list of tasks completed as part of this project:

- Develop a process simulation package that can be used to model generic chemical processing systems.
- Model and verify accuracy of the models of various metal finishing operations, including metal plating, cleaning, rinsing, and other ancillary processes.
- Create a database of thermodynamic properties and equilibrium calculations describing the interactions of chemicals in the metal plating systems.
- Develop a user's guide and online help tool.
- Conduct beta testing of MFFPPT. Beta testing will be conducted separately.

1.3 Quality Objectives for Software Development

The quality objectives for MFFPPT were as follows:

- The software should consistently produce accurate calculations based on the data inputted and the algorithms of the impact indicator models.
- The software should be both user-friendly and educational for users.
- The software should have screen designs that are legible and visually attractive to the users.
- The software should have a useful online help feature that will provide step-by-step instructions on how to use the tool.
- The user's guide should provide a brief introduction to metal finishing, extensive information on the pollution prevention models and methodologies used within MFFPPT, and information on how to use the tool.
- The software should provide relevant and understandable reports of the results, including both tables and graphs.

1.4 Hardware and Operating System Requirements

This program is being developed primarily for use on Microsoft Windows-based personal computers running Version 2.0 of Microsoft's .NET Framework. Any PC capable of running the .NET Framework should be capable of running this program. Microsoft's requirements for running the redistributable version of the .NET Framework (Microsoft's .NET Framework Redistributable web page) are as follows:

- Supported Operating Systems: Windows 2000 Service Pack 3; Windows 98; Windows 98 Second Edition; Windows ME; Windows Server 2003; Windows XP Service Pack 2, Windows Vista
- Required Software: Windows Installer 3.0 (except for Windows 98/ME, which require Windows Installer 2.0 or later). Windows Installer 3.1 or later is recommended. IE 5.01

or later: You must also be running Microsoft Internet Explorer 5.01 or later for all installations of the .NET Framework.

- Disk Space Requirements: 280 MB (x86), 610 MB (x64)

It should be noted that an open-source cross-platform version of the .NET Framework is being created by the Mono-Project. It is anticipated that this application should be able to work under Mono, but at present, there are no plans to test or support this product under the Mono environment as part of the current project.

2.0 Functional Requirements

MFFPPT is envisioned to be a tool used by the regulated community to compare pollution generation under different pollution reduction options, and used by regulators to determine whether facilities have sufficiently explored various common techniques to reduce pollution generation. The target audience for this program is the engineering community from the metal finishing and possibly broader chemical process industry, not in need of excessive introduction to the program during each launch. Rather, the user will benefit from a tutorial introduction that can be accessed through the help facility. Also, MFFPPT will include on-line help features to define key terms and to further explain how to use the system.

The program uses a traditional windows-style graphical user interface (GUI), with the menu bar having minimum choices of "File", "Edit", "View", "Tools", "Window", and "Help". Additionally, a "tool bar" will be available for users to select process objects for building the process flowsheet. Appropriate GUI controls will be available for the user to select and enter model parameters.

MFFPPT will provide results of a simulation in the forms of tables and reports. This report will include status of the simulation, chemical composition of input and output streams, environmental impact of air, liquid and solid wastes produced, and energies used. Because of the uncertainties introduced by the simulation of the metal plating line, the numbers given in the report are only approximations.

MFFPPT is a computer-based modeling package. Users should note that model accuracy needs to be verified against actual conditions. The purpose of MFFPPT is not to predict exactly the amount of wastes generated by a metal finishing line, but only to assist users in the design of their own metal plating line. MFFPPT does this by giving users indications of what

configurations are promising for the reduction of pollution. With this knowledge, the users may try the most promising configurations in their own metal plating facilities to effectively reduce wastes.

3.0 System Design Overview (High Level Design)

A general illustration of the application's architecture is presented in Figure 1. The program consists of two parts, the process modeling component (PMC) class libraries and the process modeling environment (PME), which includes the graphical user interface. The PMC class libraries were built upon the base classes that implement the functionality specified for the appropriate CAPE-OPEN interfaces. In addition, derived unit operation and thermodynamic modeling classes have been created from the appropriate base classes for use within the graphical user interface (GUI) to simulate specific processes, in this case, metal finishing processes. These PMCs were created for use within the GUI by inheriting the generic base classes described here and implementing a limited number of functions specific to the unit operation or thermodynamic property objects; specifically, class constructors and calculate methods.

The application described here consists of a steady-state sequential modular flowsheeting tool. Sequential modular simulation operates by calculating unit operation objects to determine output streams based upon input concentrations and the unit operation's mathematic model. Unit operations are calculated in a sequence based upon their location within the flowsheet, that is, unit operations are calculated only after their input stream concentrations have been determined. In this scheme, recycle loops are problematic in that their concentrations are dependent upon downstream unit operations. Recycle loops are identified using graph theory methods described later. An alternative solution scheme, equation-oriented (EO) simulation, is available. EO simulation involves creation of a set of simultaneous equations that describe the flowsheet and solves these equations using non-linear simultaneous equation solvers. In general, sequential modular simulation is powerful and easily accessible to many engineers for the solution of

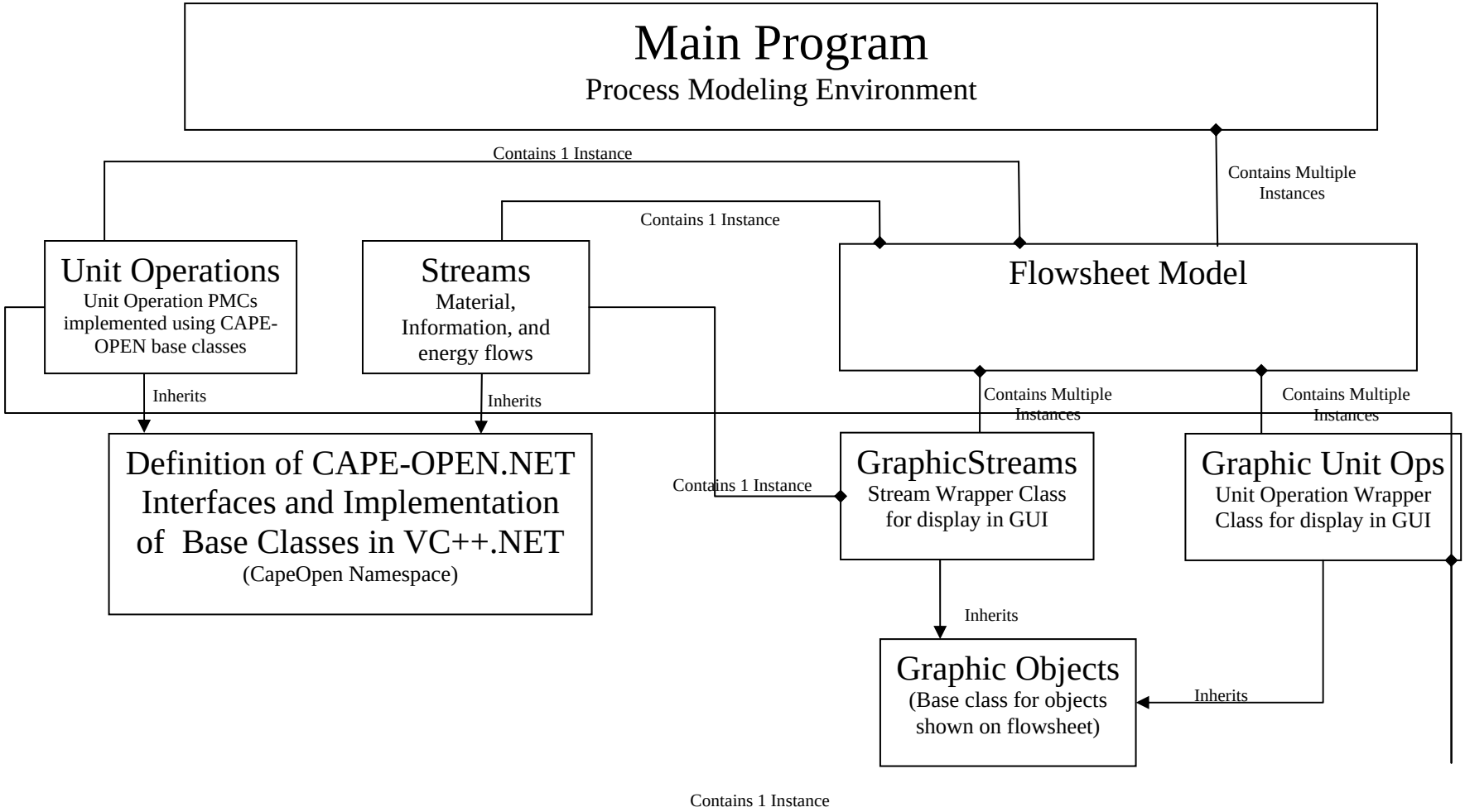


Figure 1. General Structure of the Simulation Application.

steady-state flowsheet problems, as opposed to equation-oriented simulation problems which are more complex and largely used for complicated problems such as dynamic simulation (Marquardt 1996).

In a sequential modular flowsheeting tool, unit operations and streams are connected to form a flow network. The flow network is then evaluated to locate recycle loops using a graph theory routine. The calculation process involves calling the calculate method on the unit operations to determine output stream conditions based upon input stream conditions. Loops are calculated iteratively until convergence criteria are met.

The program is created using Microsoft's Visual C++ Version 2005. The code is written in C++/CLI and the code is compiled into verifiable assemblies using the /clr:safe compiler option. The resulting assemblies can be used within the C# and Visual Basic Languages, as well as other programming languages that create assemblies for the Microsoft .NET framework. Further, verifiable assemblies can be verified not to violate security settings before executing the code.(Microsoft)

4.0 Implementation

This section describes the basic structure of the application and provides details of the construction of the class developed for its implementation. It will also provide an overview of the process for creating and using custom process modeling components within the GUI. The discussion of the architecture will be from the bottom up.

4.1 Process Modeling Components

Process modeling components are objects that can be added to the flowsheet to represent the unit operations or material/information/energy flows within the flowsheet; thermodynamic property models; reactions models; or mathematical model solvers. The process conditions are evaluated by determining the output flows from each unit operation based upon its inputs, operating conditions, and the mathematical relationships between inputs and outputs.

This section provides an overview of the base classes that implement the CAPE-OPEN interfaces developed for this application. Creation of specific material and unit operation class from these base classes to create CAPE-OPEN-compliant process modeling components will also be presented, as well as an example of classes developed for use within the present metal finishing simulation.

4.1.1 CAPE-OPEN Base Classes

Classes that implemented the CAPE-OPEN interfaces, such as the *ICapeIdentification* (Belaud and Piñol 2000), *ICapeCollection*(CO-LaN 2003), and *ICapeParameter* (CO-LaN 2003) interface, were created. The *CCapeIdentification* class implements the *ICapeIdentification* interface, and exposed the *ComponentName* and *ComponentDescription* properties in

ICapeIdentification. This class is inherited by all CAPE-OPEN process modeling component classes to provide a component name and description for the class.

The parameter classes were created as wrappers around the appropriate .NET type. For example, the *CCapeRealParameter* class contains the parameter's value as a double precision value, and implements the *ICapeParameter*, *ICapeParameterSpec*, and *ICapeRealParameterSpec* interfaces in order to comply with the CAPE-OPEN specifications (CO-LaN 2003). In addition, the *CCapeRealParameter* class provides overloads of the addition, subtraction, multiplication, division, and logical comparison operators to enable the developer to directly use the parameter in equations. Similar parameters have been created for *integers*, *boolean*, and *string* data types. Array parameter classes contain an *array<Type>* where type is the respective parameter type, and return an array of the parameter's type, e.g. calling *get_value()* on a *CCapeStringArrayParameter* would return an *array<String>*.

Type-safe parameter and port collection classes were created that implement the CAPE-OPEN *ICapeCollection* interface (CO-LaN 2003) and derives from the .NET framework the *BindingList* class template (Ballard 2005). The *BindingList* class template provides implementation of the .NET framework *IList*, *ICollection*, and *IEnumerable* interfaces as well as standard COM collection functions (*get_count()* and *get_Item()* methods), including an enumerator. The enumerator facilitates iteration through all members of the collection. Use of type-safe collection templates provides compile-time assurance that the port and parameter collections contain only ports or parameters, respectively. The collections contain elements that are either CAPE-OPEN *ICapeParameters* or *ICapeUnitPorts*, and can be accessed from COM using the CAPE-OPEN *ICapeCollection* interface.

The *CCapeObject* class serves as a base class for all process modeling components, inherits the *CCapeIdentification* class, and implements the *ICapeUtilities* (CO-LaN 2003) interface. *CCapeObject* includes a member variable, *m_parameters*, which is the above-described collection of parameters. The *CCapeObject* class is then inherited by unit operation, material object, information object, and energy object classes.

In order to save the objects to a file, all of the base classes are marked with the .NET *Serializable* attribute (Obermeyer and Hawkins 2002). Derived classes must also be marked with this attribute to be serialized. The current application serializes the flowsheet as an extensible markup language (XML) document using .NET framework's Simple Object Access Protocol (SOAP) XML serializer class. This attribute also enables the .NET framework to serialize the object to remote computers. This is the equivalent of Marshalling in COM and CORBA. The ability to interact with remote application domains is one of the primary advantages and primary reasons to use the .NET framework for development. The current application contains wrapper classes for unit operations and property packages that allow persistence of COM-based CAPE-OPEN PMCs. Presently, any CAPE-OPEN COM-based PMC can be used in this application just like a native .NET based PMC.

The primary goal of the unit operation (*CCapeUnit*) and thermodynamic property (*CCapeThermoPropertyPackage*) object base classes is to implement as much of the generic functionality of the CAPE-OPEN interface specification as practical, essentially encapsulating this functionality. Functions whose action may be different in different unit operations are declared as virtual (*Overridable* in Visual Basic) which means that the derived class implementations exist on top of the base class implementation. The *ICapeUtilities*.*Edit*() method is an example of a method defined as virtual in the base class, with a default implementation. A

derived class can then override the base class implementation and have its own implementation. As a default, the *Edit()* function throws a *CapeNoImplException* not implemented exception, which is caught by the flowsheet application as a signal that the flowsheet's default editor should be used. If the *Edit()* function has been overridden and does not return a failure condition (throw an exception), the GUI assumes that the object successfully edited itself, which conforms to the CAPE-OPEN unit operation standard(CO-LaN 2001). If the *Edit()* function throws the *CapeNoImplException* exception, the GUI will call its built-in unit operation editor, which provides a list of parameters and ports. Actual implementation of process modeling components is discussed below.

4.1.2 Exception Classes

One of the principal advantages of .NET over COM is the additional information included in exception handling. In COM, exceptions were handled using the ::*GetErrorInfo* API that consisted of saving an error object and having the function return an *HRESULT* value, which is an integer that indicated whether the function call had successfully returned (Rogerson 1997). Because the *HRESULT* value was a 32-bit integer, it could indicate more information than simply success or failure, but it was limited in that it did not include descriptive information about the exception that occurred. The *ErrorInfo* object was used to provide more information on the source of the error including descriptive text. COM objects implementing the ::*GetErrorInfo* API readily interact with .NET exceptions. CAPE-OPEN chose to not use the ::*GetErrorInfo* API due to issues with previous versions of Microsoft's Visual Basic, and the CAPE-OPEN compliant error handling mechanism differs from, and is only partially compatible with, from the exception handling mechanism provided by .NET. Because PMCs developed as part of this project are not envisioned to be used in a non-.NET environment, a decision was made that the

PMCs would not support the CAPE-OPEN error handling method and instead implement CAPE-OPEN interfaces through application exception objects. The PME does however support the CAPE-OPEN error handling protocols for third-party PMCs, converting the CAPE-OPEN errors into .NET exceptions.

Under .NET, an application exception class is available (*System.ApplicationException*) that should be used as a base class for application-based errors to distinguish them from system-based error conditions. The *System.ApplicationException* class inherits the *System.Exception* class to provide information such as a message and the source of the exception.

The CAPE-OPEN exception definitions all derive from an *ECapeRoot* root error interface (CO-LaN 2003). In the implementation of the CAPE-OPEN exception classes described here, all exception classes derive from the *CapeRootException* class, which itself is derived from the .NET *System.ApplicationException* class. The *CapeRootException* class exposes the *ECapeRoot* interface. In this way, all exceptions that are raised by the process modeling components can be caught either as a *CapeRootException* or as a *System.ApplicationException* in addition to being caught as the derived exception type.

4.1.3 Creation of Unit Operation Classes

Simulations of the various unit operations provide the core calculation component of the flowsheet. During a flowsheet calculation, the unit operation's calculate function is called in the order determined by the flowsheet's sequencing routine. Results of the calculation are based upon the equations used to simulate the unit, the value of parameters of the unit operation, and the material, energy, and information flows to the unit. This section presents a discussion of the structure and assemblage of unit operation components.

The structure of the current implementation of a unit operation model is shown in Figure 2. As previously described, the *CCapeUnit* class inherits the *CCapeObject* class that inherits the *CCapeIdentification* class and contains a collection of parameters. Additionally, the *CCapeUnit* implements the *ICapeUnit* (CO-LaN 2001) interface and contains a collection of *CCapePort* objects. The CCapePort class implements the *ICapeUnitPort* interface, and it contains a collection of port variables, which are themselves parameters.

Figure 3 and **Figure 4** contain Visual Basic and Visual C# code listings that implement a mixer unit operation that were created to demonstrate interoperability with the COM-based CAPE-OPEN to CAPE-OPEN (COCO) simulation environment developed by AmsterChem (http://www.cocosimulator.org/) . These mixer units have been successfully tested within COCO's CAPE-OPEN Flowsheeting Environment (COFE) using COCO's Thermodynamics for Engineering Applications (TEA) property package. COCO's CAPE-OPEN Unit Operations (Simple) (COUS) unit operation models have also been tested in the subject program.

The constructors of both classes create and add two inlet and one outlet port to the port collection. In addition, real-, integer-, boolean- and string (option)-parameters are created and added into the parameter collections to verify parameter interoperability. The calculation method obtains the material objects connected to each port. For each input material object, the molar flows of each chemical component and the total enthalpy are obtained. These flows and enthalpies are then added and set within the outlet material object. The pressure of the outlet material is then set using the first unit operation (real-valued) parameter. Lastly, an enthalpy/pressure flash calculation is performed on the output material to determine vapor/liquid phase equilibrium. This mixer unit operation successfully functioned in COFE with the TEA property package, which verifies compliance with the CAPE-OPEN standard.

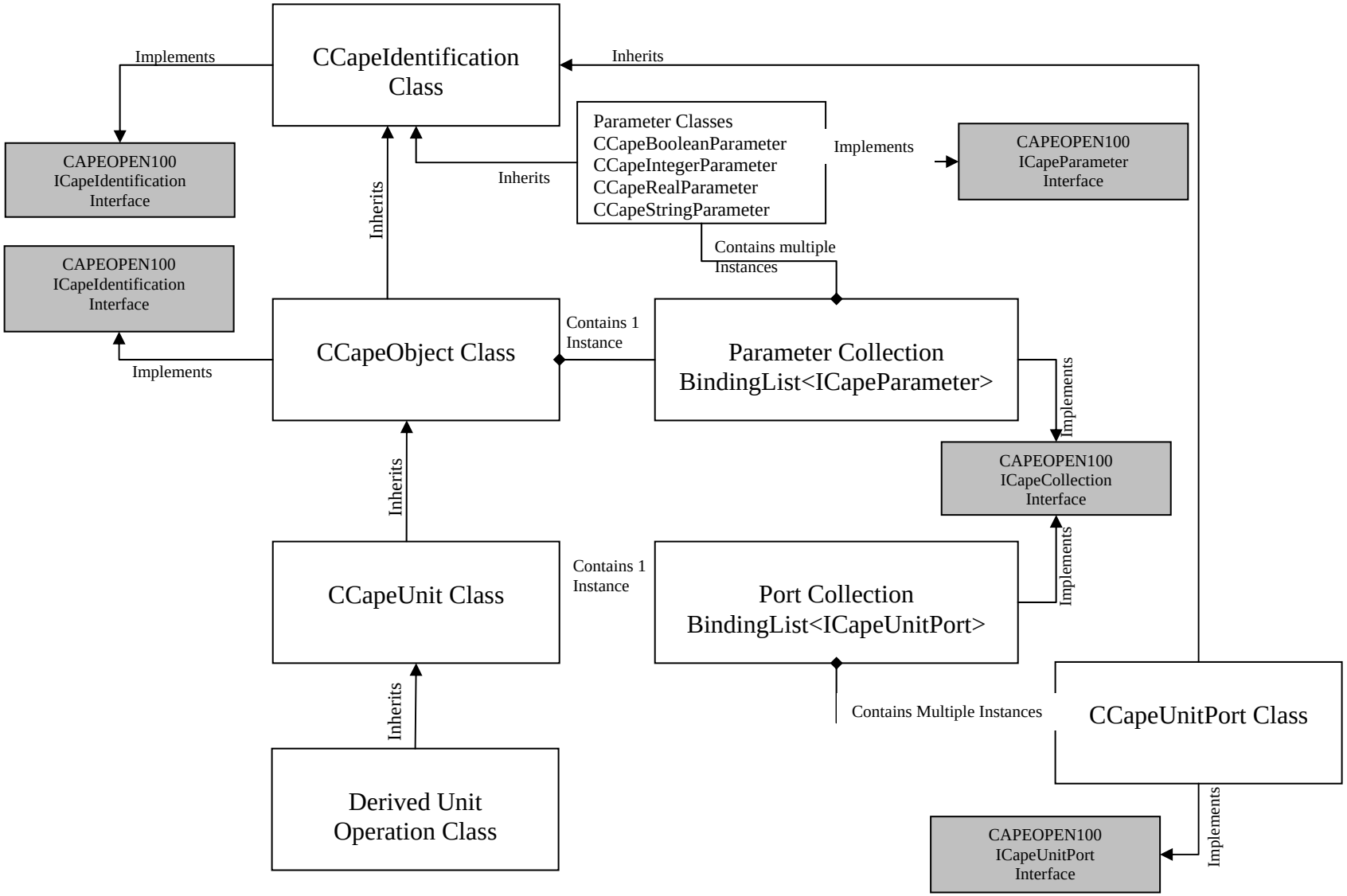


Figure 2. Model of Unit Operation Classes that Derive from the CCapeUnit Class.

```
Imports System
Imports System.Collections.Generic
Imports System.Text
<Serializable()> _
<System.Runtime.InteropServices.ComVisible(True)> _
<System.Runtime.InteropServices.Guid("DDD31647-2E01-4174-827F-A3AF9005B213")> _
Public Class MixerExample
    Inherits UnitBase

    Public Sub New()
        Me.ports.Add(New UnitPort("Inlet Port1", "Test Inlet Port1", CAPEOPEN110.CapePortDirection.CAPE_INLET, CAPEOPEN110.CapePortType.CAPE_MATERIAL))
        Me.ports.Add(New UnitPort("Inlet Port2", "Test Inlet Port2", CAPEOPEN110.CapePortDirection.CAPE_INLET, CAPEOPEN110.CapePortType.CAPE_MATERIAL))
        Me.ports.Add(New UnitPort("Outlet Port", "Test Outlet Port", CAPEOPEN110.CapePortDirection.CAPE_OUTLET, CAPEOPEN110.CapePortType.CAPE_MATERIAL))
        Me.parameters.Add(New RealParameter("Pressure", 250000.0))
        Me.parameters.Add(New IntegerParameter("Parameter2", 12))
        Me.parameters.Add(New BooleanParameter("Parameter3", False))
        Dim options() As String = {"Test Value", "Another Value"}
        Me.parameters.Add(New OptionParameter("Parameter4", "OptionParameter", "Test Value", "Test Value", options, True,
CAPEOPEN110.CapeParamMode.CAPE_INPUT_OUTPUT))
    End Sub
    Public Overrides Sub Calculate()
        Try
            Dim phases() As String = {"Overall"}
            Dim props() As String = {"enthalpy"}
            Dim in1 As CAPEOPEN110.ICapeThermoMaterialObject = Me.ports(0).connectedObject
            Dim in1Comps() As String = in1.ComponentIds
            Dim in1Flow() As Double = in1.GetProp("flow", "Overall", Nothing, Nothing, "mole")
            in1.CalcProp(props, phases, "Mixture")
            Dim in1Enthalpy() As Double = in1.GetProp("enthalpy", "Overall", Nothing, "Mixture", "mole")
            Dim in2 As CAPEOPEN110.ICapeThermoMaterialObject = Me.ports(1).connectedObject
            Dim in2Comps() As String = in2.ComponentIds
            Dim in2Flow() As Double = in2.GetProp("flow", "Overall", Nothing, Nothing, "mole")
            in2.CalcProp(props, phases, "Mixture")
            Dim in2Enthalpy() As Double = in2.GetProp("enthalpy", "Overall", Nothing, "Mixture", "mole")
            Dim outPort As CAPEOPEN110.ICapeThermoMaterialObject = Me.ports(2).connectedObject
            Dim values(0) As Double
            values(0) = in1Enthalpy(0) + in2Enthalpy(0)
            outPort.SetProp("enthalpy", "Overall", Nothing, "Mixture", "mole", values)
            values(0) = Me.parameters(0).value
            outPort.SetProp("Pressure", "Overall", Nothing, Nothing, Nothing, values)
            ReDim values(in1Comps.Length - 1)
            Dim i As Integer
            For i = 0 To in1Comps.Length - 1
                values(i) = in1Flow(i) + in2Flow(i)
            Next i
            outPort.SetProp("flow", "Overall", in1Comps, Nothing, "mole", values)
            outPort.CalcEquilibrium("PH", Nothing)

        Catch p_Ex As System.Exception
            System.Windows.Forms.MessageBox.Show(p_Ex.ToString())
        End Try
    End Sub

End Class
```

Figure 3. Visual Basic Code Implementing a Mixer Unit Operation.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace C_Sharp_Examples
{
    [Serializable]
    [System.Runtime.InteropServices.ComVisible(true)]
    [System.Runtime.InteropServices.Guid("4B7BEC6D-BA0A-42f5-9AA8-4D5904E79035")]//ICapeThermoMaterialObject_IID)
    public class MixerExample : CapeUnitBase
    {
        public MixerExample()
        {
            this.PortCollection.Add(new CapeUnitPort("Inlet Port1", "Test Inlet Port1", CapePortDirection.CAPE_INLET, CapePortType.CAPE_MATERIAL));
            this.PortCollection.Add(new CapeUnitPort("Inlet Port2", "Test Inlet Port2", CapePortDirection.CAPE_INLET, CapePortType.CAPE_MATERIAL));
            this.PortCollection.Add(new CapeUnitPort("Outlet Port", "Test Outlet Port", CapePortDirection.CAPE_OUTLET, CapePortType.CAPE_MATERIAL));
            this.ParameterCollection.Add(new CapeRealParameter("Pressure", 250000.0));
            this.ParameterCollection.Add(new CapeIntegerParameter("Parameter2", 12));
            this.ParameterCollection.Add(new CapeBooleanParameter("Parameter3", false));
            string[] options = { "Test Value", "Another Value" };
            this.ParameterCollection.Add(new CapeOptionParameter("Parameter4", "OptionParameter", "Test Value", "Test Value", options, true,
                        CapeParamMode.CAPE_INPUT_OUTPUT));
        }

        public override void Calculate()
        {
                ICapeThermoMaterialObject in1 = (this.ports as PortCollection)[0].connectedObject as ICapeThermoMaterialObject;
                string[] phases = { "Overall" };
                string[] props = { "enthalpy" };
                string[] in1Comps = in1.ComponentIds as string[];
                double[] in1Flow = in1.GetProp("flow", "Overall", null, null, "mole") as double[];
                in1.CalcProp(props, phases as object, "Mixture");
                double[] in1Enthalpy = in1.GetProp("enthalpy", "Overall", null, "Mixture", "mole") as double[];
                ICapeThermoMaterialObject in2 = (this.ports as PortCollection)[1].connectedObject as ICapeThermoMaterialObject;
                string[] in2Comps = in1.ComponentIds as string[];
                double[] in2Flow = in2.GetProp("flow", "Overall", null, null, "mole") as double[];
                in2.CalcProp(props, phases, "Mixture");
                ICapeThermoMaterialObject outPort = (this.ports as PortCollection)[2].connectedObject as ICapeThermoMaterialObject;
                double[] in2Enthalpy = in2.GetProp("enthalpy", "Overall", null, "Mixture", "mole") as double[];
                double[] values = new double[1];
                values[0] = in1Enthalpy[0] + in2Enthalpy[0];
                outPort.SetProp("enthalpy", "Overall", null, "Mixture", "mole", values);
                values[0] = (double)(this.parameters as ParameterCollection)[0].value;
                outPort.SetProp("Pressure", "Overall", null, null, null, values);
                values = new double[in1Comps.Length];
                for (int i = 0; i < in1Comps.Length; i++)
                {
                    values[i] = in1Flow[i] + in2Flow[i];
                }
                outPort.SetProp("flow", "Overall", in1Comps, null, "mole", values);
                outPort.CalcEquilibrium("PH", null);
        }
    }
}
```

Figure 4. Visual C# Code Implementing a Mixer Unit Operation.

4.1.4 Creation of Thermodynamic Classes

Material Objects provide information about material flows within the flowsheet. The current implementation of the thermodynamic model consists of three basic parts, the Material Object itself, the Property Package, and the Reaction Package. Each of these parts is based upon its respective CAPE-OPEN interface standards (CO-LaN 2002). The basic structure of the material system is shown in Figure 5. As previously discussed, the *CCapeThermoMaterial* class is based upon the *CCapeObject* class, and can be inherited by developers to provide conceptual models of the material that is being simulated. The CCapePropertyPackage class however, derives directly from the CCapeIdentification class. Both the *CCapeThermoMaterialObject* and *CCapeThermoPropertyPackage* classes are built around a data table (.NET framework *DataTable* class) that contains appropriate general, physical, and chemical properties of the chemical species modeled.

The *CCapeThermoMaterialObject* class provides both mechanism to obtain the constant properties of each chemical species from the Property Package, and to have the Property Package calculate the values of chemical properties dependent upon temperature, pressure, and composition. The *CCapeThermoMaterialObject* class contains a parameter collection that stores the Material Object's pressure, temperature, and flow rate, and exposes these overall properties as properties of the object. The *CCapeThermoMaterialObject* class also contains a data table that initially contains the only IUPAC name and concentration of each of the chemical species in the *CCapeThermoMaterialObject*. The *CCapeThermoMaterialObject* class can then obtain the value of constant properties for each chemical desired based upon the chemical's International Union of Pure and Applied Chemistry (IUPAC) name directly from the Property Package, or could

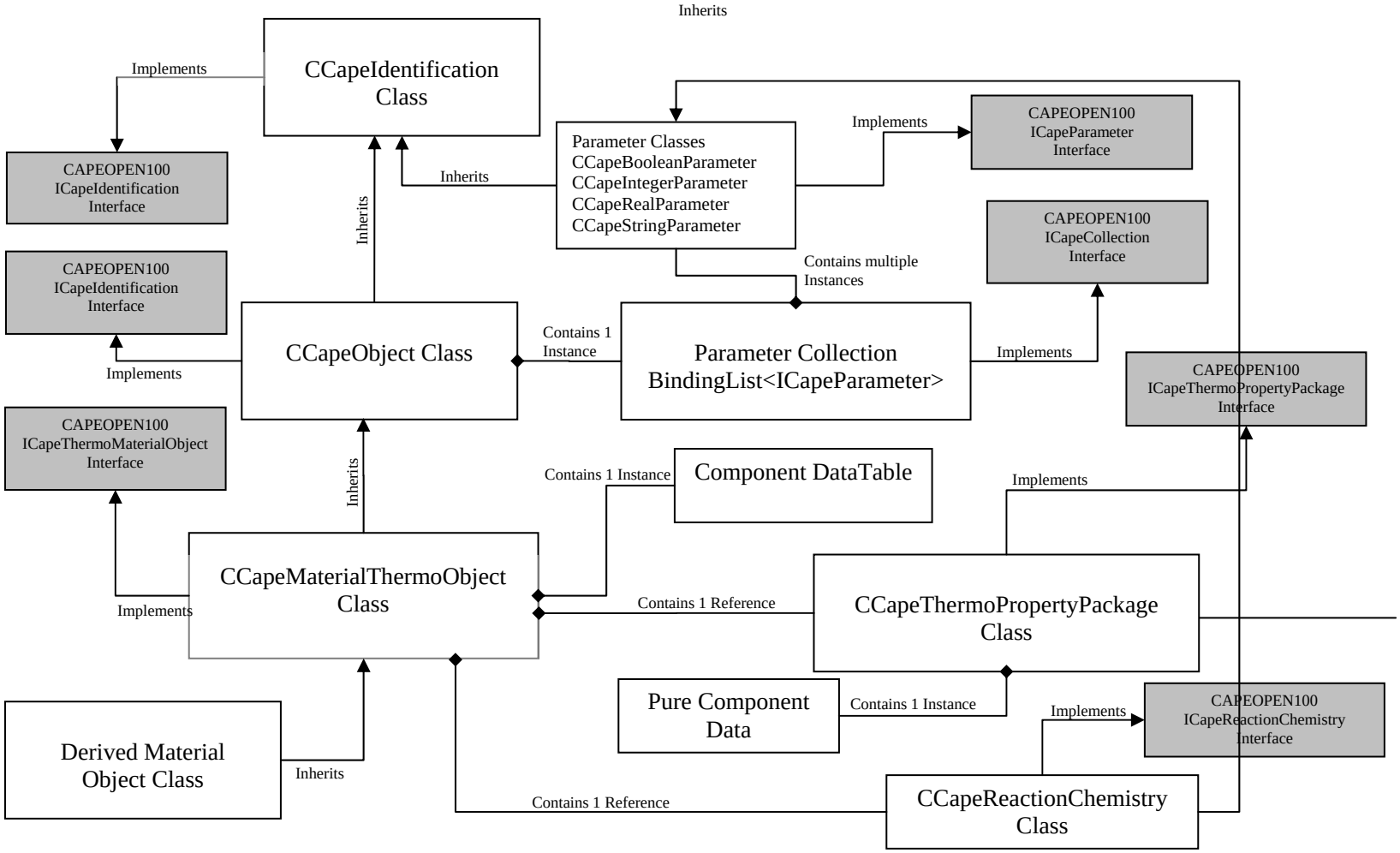


Figure 5. Model of Material Object Classes that Derive from the CCapeThermoMaterialObject Class.

request that the Property Package calculate values of non-constant properties based upon temperature, pressure and composition of the *CCapeThermoMaterialObject*, and the values of these properties can be added to the material object's data table for later use. Additionally, the *CCapeThermoMaterialObject* class can call the Property Package's *CalcEquilibrium* method.

CCapeReactionChemistry class is a chemical reaction database based on the CAPE-OPEN chemical reactions interface specification (CO-LaN 2003) that consists of a collection of chemical reactions. Each chemical reaction is a collection of reactants. A reactant is implemented as a data structure that has the following members: an integer reaction coefficient, a string that contains the name of the chemical species, an integer that contains the ionic charge of the chemical species, a string containing the Chemical Abstracts Services (CAS) number of the chemical species, a Boolean value indicating whether the reactant is the base reactant of the reaction, and a string indicating the phase in which the reaction occurs. Each reaction provides information to the *CCapeReactionChemistry* class to respond to functions about the specific reaction, such as *CCapeReactionChemistry*.*GetReactionCompoundIds*, which returns the names of the chemical species that are involved in the specified reaction.

Currently, the *CCapeThermoPropertyPackage* class provides both the chemical property data of chemical species that can be used in the flowsheet, the ability to calculate chemical properties of mixtures, and the phase equilibrium calculation routines. The *CCapeThermoPropertyPackage* class's data table contains the various constant properties of the chemical species, which includes a subset of the constant properties of the chemical listed in the CAPE-OPEN interface specification. The *ICapeThermoPropertyPackage* interface provides functions to obtain the values of the chemical species' constant properties, and methods to calculate the equilibrium conditions and chemical properties based upon the conditions of a

Material Object passed as a parameter to the *CCapeThermoPropertyPackage* calculate routines. The current implementation of the *CCapeThermoPropertyPackage* class also provides a method for the GUI to obtain a reference to its data table. This allows the GUI to implement functionality to edit the *CCapeThermoPropertyPackage*'s data table and mechanisms to link the property table to external data sources, such as a Microsoft EXCEL spreadsheet or a SQL server. Since different types of material objects may require different methods to calculate their equilibrium, the *ICapeThermoMaterialObject.CalcEquilibrium* function calls the *CalcEquilibrium* method in the property package attached to the material object.

4.2 Flowsheeting Program Architecture

The flowsheeting program consists of five separate projects: the Add-ins model, the CAPE-OPEN base classes described above, a set of graphic objects which are used to draw the process modeling components on the GUI, controls used to present flowsheet information, and the main application. This section will describe the Add-In model, graphic objects, controls, and GUI operations.

4.2.1 Add-In Model

The Add-In model is based upon an extensibility model developed for Visual Basic (Clark 2003). This model creates directories within the application's base directory to contain both trusted and not fully trusted library files. The application loads each of the files within the plug in directories, scans the libraries for types that can be used by the application, and then installs appropriate toolbar buttons or menus into the application that are provided by the plug-in objects.

There are four basic types that the Add-In model searches the libraries for. The first classes added are classes that implement the *IToolStripProvider* interface. This interface

provides two methods, one provides a toolstrip whose buttons are added to the process modeling component toolstrip, and the other provides the Toolstrip Button Click event handler for the toolstrip. In the current implementation, the *Tag* property of the added toolstrip buttons holds the Type of object that will be created when the button is pushed. In this way, when the object is selected to be added to the GUI by pushing its button on the toolstrip, the object created is of the Type indicated by the button's Tag. The toolstrip's button click event handler obtained through the interface is added to the event handlers for the component toolstrip so that pushing the button will implement the desired custom functionality.

Once the toolstrips have been added, menu items are added by identifying classes that implement the *IMenuProvider* interface. The *IMenuProvider* interface has one method that returns an array of menu items to be added to the application menu. Menu items have event handlers built in, so menu items that are added will need to respond to their internal events.

In addition to the interfaces listed in the Add-Ins extensibility mode, the application also looks for classes in the library files that implement the *ICapeUnit* and *ICapeThermoMaterialObject* interfaces. When classes that implement these interfaces are located, the buttons on the toolstrip are searched to determine if a button's *Tag* is the same *Type* as the class. If the class has not been added to the toolstrip, a button with a default icon is added to the toolstrip and its Tag is set to the *Type* of the class. Additionally, the button's *ToolTip*, shown when the mouse hovers over the button, is set to the name of the Type associated with the button. This mechanism allows the addition of COM-based CAPE-OPEN compliant unit operation or material object models to the current flowsheeting program.

4.2.2 Graphic Object Classes

The current application utilizes a graphic object model built around a Graphics Device Interface plus (GDI+) design surface control (Mackenzie 2002), which allows users to create and position various graphical elements. The design surface has a drawing object collection that acts as a container for all graphical objects added to the flowsheet. The graphic objects that can be added include shapes, text, and bitmap images. Graphic objects can be moved around on the design surface by pressing the mouse button while the cursor is located on the object to select it and then dragging the object by moving the mouse while the mouse button is depressed. Another function of the design surface is to format the graphics to be printed out of a printer attached to the computer.

Process modeling components, such as unit operations and flow streams, are added to the flowsheet by placing a container class within the design surface's graphic object collection. Unit operations are placed in a *GraphicUnitOperation* class that displays the default icon for the unit operation on the flowsheet. Stream objects, such as material streams, are contained in a *GraphicStream* container class that displays lines from the source unit operation to the destination unit operation. Both the *GraphicUnitOperation* and *GraphicStream* classes have methods that allow CAPE-OPEN compliant components to be contained and accessed. At present, COM-based CAPE-OPEN unit operations can be instantiated and wrapped in the *GraphicUnitOperation* class which is displayed on the GUI and placed in the flowsheet. Additionally, the unit operation's COM-based port and parameter collections can be browsed and .NET based material objects can be connected to the COM-based ports.

In the current implementation, only the design surface's ability to place a bitmap icon representing a unit operation and lines representing streams on the flowsheet are utilized. However, the GUI's graphical capability can be enhanced to provide the ability to add text notes,

and other non-simulation related graphics as well as options to draw the process modeling components as a collection of shapes supported by the GDI+ graphics model.

4.2.3 Controls

A number of controls have been created to enable users to input information into the unit operations and material objects. The purpose of the controls is primarily to allow users to view and edit the parameters of a process modeling component, or edit the composition of a flow stream. Controls to edit the parameters of a unit operation or a stream can be accessed by double clicking the mouse while the cursor is located over the object to be edited. The controls to edit the property package and equation packages are accessed through the application's menu.

4.3 Main Application

On startup, the application displays the main form (*Form1*) of the application. This form is a multiple document interface (MDI) container. The child documents contained in the MDI form are flowsheets. In addition to displaying flowsheets, Form1 also has a tree view/data table class that show the process modeling components and a data table that lists the material objects in the current child flowsheet.

As described previously, the application has a menu bar and two toolstrips. The menu and top toolstrip provide functionality to load, save, and print the flowsheet; zoom in and out; calculate the unit operations; and edit the property and reaction chemistry packages. The second toolstrip contains the icons of the process modeling components that can be added to the flowsheet. Unit operations are added to the flowsheet by clicking on the desired unit operation's icon and clicking on the flowsheet at the location where the user wants that object placed. Flow streams are added by clicking the toolstrip button corresponding to the desired stream and the selecting, in order, the start and end unit operations for the stream. Once the ends are selected, a

control is shown that allows the user to select the port to connect the stream to on both the source and destination unit operations.

4.3.1 Process Modeling Environment

The process modeling environment consists of the graphical user interface (GUI) and functionality required to create the flow network being modeled; input, review and modify values for parameters of components; input, review and modify material or energy flows; and calculate the conditions of the flowsheet based on the inputs. To accomplish the functionality, a large number of classes have been created. The purpose of this section is to provide an overview of the nature of interactions the GUI has with the flowsheet, not to present detailed description of the implementation of the classes that provides this interaction.

The GUI is shown on Figure 6. The interface has a typical windows style menus and toolstrip. The menus and top toolstrip contains commands to open, close, and save files; change the view of the flowsheet, add process modeling component libraries, sequence and calculate the flowsheet, and view and edit databases of chemical properties and reactions that may be used by the simulation. The lower toolstrip contains icons for the process modeling components that can be added to the flowsheet by users to construct the flow network.

The flowsheet is in the upper right of the GUI's window. It shows the connections between unit operations as directed lines. The unit operations are currently added to the flowsheet using the same icons that are shown on the toolstrip. Unit operations are added to the flowsheet by selecting the appropriate button on the toolstrip and clicking on the flowsheet at the location where the user wants the object added.

As unit operations and stream objects are added to the flowsheet, they are also added to the tree view on the left side of the GUI. Once added, the unit operation's and stream's tree nodes can be expanded to view more details about the object. Properties of the selected node are

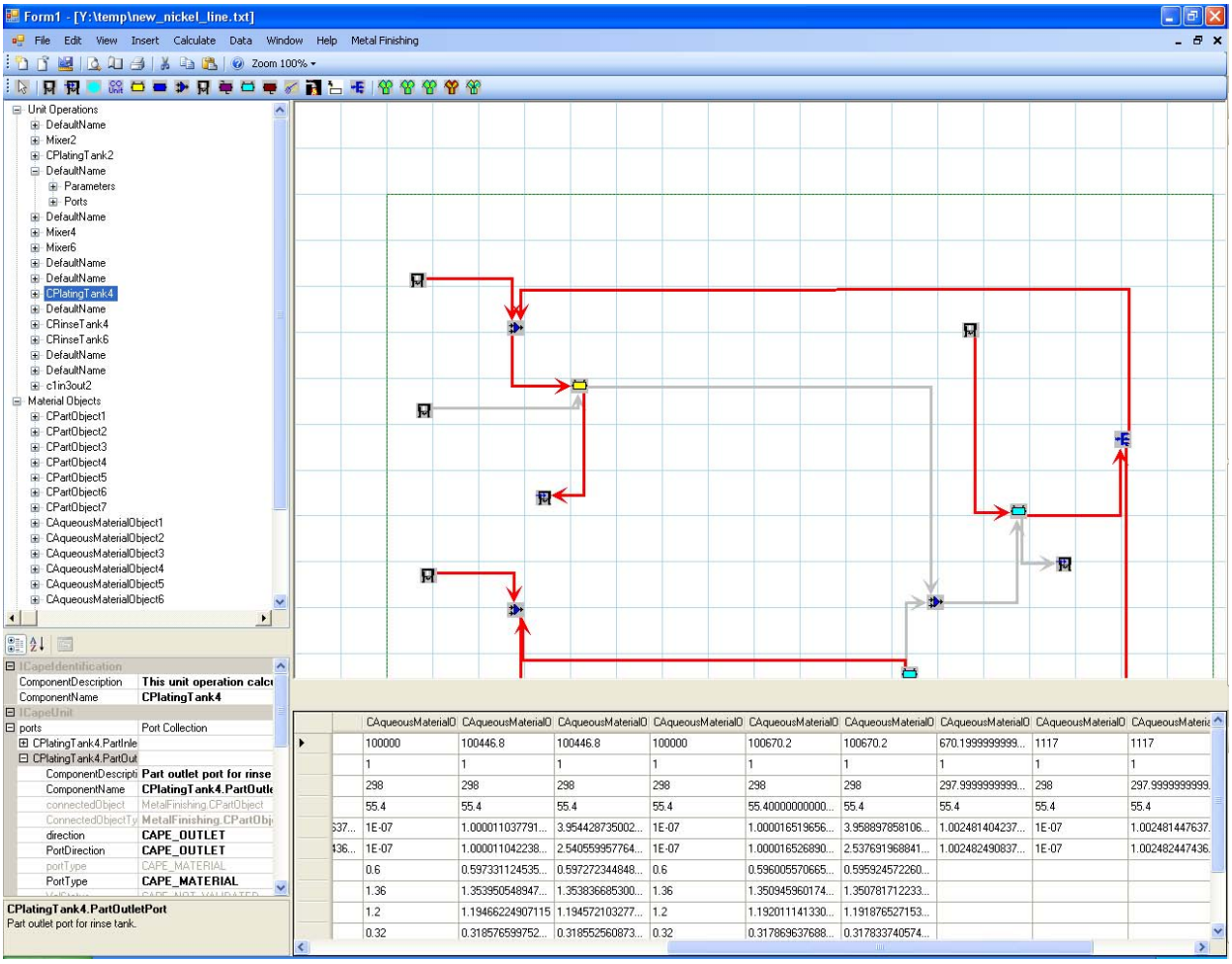


Figure 6. Graphical User Interface for the Process Flowsheeting Tool.

shown in the data grid on the lower left hand side of the GUI. Double clicking on the unit operation's icon or the flow stream brings up a separate editor window that can be used to edit parameters of the object or concentrations of materials present. The bottom right of the GUIwindow contains a material balance data table showing flow rate, temperature, pressure, and concentration of each stream, which are updated as changes are made to the flowsheet.

After the flowsheet has been entered by the user, it must be calculated. The first step in the calculation process is to determine if there are any loops present through graph analysis. The flowsheet is a directed graph, that is, a collection of vertices (the unit operations), connected by

directed edges (streams that contain information about the flows between the nodes). When a stream returns to a node that is upstream of the node that the stream left, a recycle loop is present. One way to identify recycle loops is through a depth-first search (Tarjan 1972). Depth – first search identifies strongly connected components of a graph, which a collection of nodes that can be reached from each other through a set of directed edges. Once the strongly connected components of the graph are identified, the tear stream and calculation sequence are identified using the tearing algorithm of (Varma, Lau et al. 1993).

The flowsheet is then calculated in order by calculating the unit operations until the flowsheet converges. Flowsheet convergence is tested by checking to see if the change in the values of the torn stream is smaller than a convergence criterion.

4.3.2 Metal Finishing Add-ins

The metal finishing specialization requires the creation of specialized unit operations to represent various process operations, material objects to represent flows of solutions and part objects, and a property package/equilibrium server to calculate chemical equilibrium conditions in the tanks. This section will provide a brief description of the creation of these objects.

Unit Operations will be created as described above using the *CCapeUnit* base class. Specialization involves creation of the port collection containing the appropriate port objects for attaching the material inputs and outputs flows. The parameter collection will contain information specific to the operating conditions of the tank. The calculation routine will perform the appropriate manipulation of the components and calls to the equilibrium calculations routines. For example, a plating tank will determine the loss of mass of metal from a plating solution through addition to the surface of the part object. This mass change will be converted to an addition of metal to the part surface and a decrease in solution concentration of the plated

metal. Once the final solution composition have been determined, the equilibrium concentrations of various aqueous species present will be determined through a call of the material object's *CalcEquilibrium* method.

Material object class specialization is performed to allow two different material abstractions, the aqueous solutions and the part. The aqueous solution is a generalization of the plating and other solutions used in the process. The procedure used for the equilibrium calculations described below can be used throughout the pH and concentration ranges anticipated to be encountered in the equilibrium calculation. The part object is a further generalization made by adding parameters that indicate the type of metal plated on the part and the thickness of that metal on the part. Solution drag out is envisioned as an aqueous flow attached to the part.

In the aqueous property package class created for the EPA's metal finishing program, the equilibrium calculation involves the creation of Morel's tableau (Morel and Hering 1993), a table of reaction coefficients based on the acid/base, solubility, complexation, and oxidation reduction reactions that may occur in the solution. The aqueous Property Package class has a routine that utilizes the reaction data to create the tableau by eliminating a chemical species from the solution components whose concentrations are independent variables in the material for each reaction. After a chemical species has been removed for all reactions that may occur, the remaining chemical species are solution components whose concentrations are the independent composition variables. A set of non-linear equations is generated and solved by using a Newton-Raphson type non-linear solver to determine the concentration of the chemical species that result from the calculated chemical component concentrations (van der Lee 1998).

5.0 Calculation Methodology

This section documents the methods used to calculate the individual plating unit operations. The calculations consist of three basic types: 1) metal deposition, 2) aqueous chemistry, and 3) gas-phase emissions. Each of these will be discussed below.

5.1 Calculation of Metal Deposition

The first step in the calculation process is to determine the amount of metal ions removed from solution and placed on the surface of the part. This amount is calculated using Faraday's Law (Paunovic and Schlesinger 1998):

$$\frac{dw}{dt} = \frac{I}{F}$$

where w is the number of moles plated, F is the Faraday constant, 96,487 C-mol^{-1}, I is the electric current in Amps, and t is the plating time in seconds. The change in concentration of the plating solution is given by a material balance, considering the change in concentration of the solution is a result of the metal plating the part, or:

$$V\frac{dC}{dt} = \frac{dw}{dt} = \frac{I}{F}$$

where V is the volume of the tank and C is the concentration of the metal in solution.
Once the number of moles plated is determined, the thickness of the plate is determined from the ionic charge n, atomic weight M, surface area a, and density d of the metal using the equation:

$$\frac{dh}{dt} = \frac{I}{FnMad}$$

As a result, both the change in metal concentration and the thickness of metal is dependent upon the applied current and residence time of the part in the plating tank.

5.2 Aqueous Equilibrium Calculations

In order to calculate the equilibrium concentrations for all species present, a solution scheme was developed that compared the total mass of each principal component in the solution with the amount of each component initially present. Principal components are defined in (Morel and Hering 1993) as "a set of chemical entities that permits a complete description of the stoichiometry of a system." Two concepts are expressed in this definition:

1. The mole balances corresponding to the components must provide a complete expression of conservation for the chemical system, and
2. Components must provide a complete and unique stoichiometric formula for each chemical species in the system.

Since the components provide sufficient information to express the concentration of all species present at equilibrium, the number of components present (N_{Cp}) equals the number of species (N_s) minus the number of independent reactions. This can be shown by considering the degrees of freedom of the system and subtracting the number of chemical reactions occurring as each chemical reaction itself represents an equation governing the condition of the system.

Using the above description of components, species, and the relationship of chemical reactions between them, a table (Morel's tableau) can be created that expressed the stoichiometric relationship between components and species. This tableau is created by forming a matrix having one row and one column for each species present. Each chemical reaction is then used to remove a column from the tableau by distributing the amount of each species present in the component being removed amongst the remaining components. Table 1 contains the Morel's tableau created from the following set of chemical reactions that occur in a typical Watts plating bath:

Table 1. Simplified Morel's Table for the Watts Nickel System.

Species	Reaction	Components					
		H_2O	H^+	H_3BO_3	Ni^{2+}	SO_4^{2-}	Cl^-
H_2O		1					
H^+			1				
OH^-	$H2O \leftrightarrow OH^- + H^+$	1	-1				
$NiCl_2$	$NiCl_2 \leftrightarrow Ni^{2+} + 2Cl^-$				1		2
$NiSO_4$	$NiSO_4 \leftrightarrow Ni^{2+} + SO_4^{2-}$				1	1	
H_3BO_3				1			
$B(OH)_4^-$	$H^+ + B(OH)_4^- \leftrightarrow H_3BO_3 + H_2O$	1	-1	1			
$B_3O_3(OH)_4^-$	$H^+ + B_3O_3(OH)_4^- + H_2O \leftrightarrow 3H_3BO_3$	-2	-1	3			
Ni^{2+}					1		
SO_4^{2-}						1	
$NiCl^+$	$NiCl^- \leftrightarrow Ni^{2+} + Cl^-$				1		1
Cl^-							1

NiCl2 ↔ Ni2+ + 2Cl-
NiSO4 ↔ Ni2+ + SO42-
NiCl- ↔ Ni2+ + Cl-
H+ + B(OH)4- ↔ H3BO3 + H2O
H+ + B3O3(OH)4- + H2O ↔ 3H3BO3

In this case, the species present represent the speciation of boric acid (Edwards 1953) and nickel salts in water. Other nickel and boric species aren't considered in this paper due to the acidic condition of usual Watts plating baths. Water and the hydrogen ion (H^+) as a general rule are always selected as components. It should be noted that any species present in the chemical reaction could be selected as a component, and the choice of the component set often can have effect the rate of convergence of the calculation, and even whether the system will converge.

A complete specification of the chemical system includes a recipe, list of reactions, and a list of species. The amount of each component represents the "recipe" for making the solution and this "recipe" is equal to the sum of the amount of each component present in the species initially added to the solution. Quantitatively, this "recipe" is determined by multiplying the initial concentration of each species present by its stoichiometric coefficient in the Morel's

tableau. At equilibrium, the total amount of each component present in solution will equal the amount of each component in the "recipe", or:

$$C_{i-Total} = \sum_{j}^{N_a} \alpha_{i,j} S_j$$

where:

$C_{i\text{-}Total}$ = the total concentration of component *i*.

N_S = the number of species present.

$\alpha_{i,j}$ = the stoichiometric coefficient relating the mass of component *i* in species *j*.

S_j = the concentration of species *j*.

The values in Morel's Tableau are the stoichiometric coefficient ($\alpha_{i,j}$) of the chemical reaction where species *j* is in equilibrium with the amount of component *i* present in the solution. The concentration of each species S_j is determined from the concentration of each component using mass-action laws:

$$S_j = K_j \prod_{i}^{N_{Cp}} c_i^{\alpha_{i,j}}$$

where:

c_i = the concentration of free component i in solution.
K_j = the equilibrium coefficient for the reaction between species j and the components present.
N_{Cp} = the number of components present.

Combining the equations above, and assuming that $[H_2O]$ is 1, gives the following equations for the concentrations of the different components:

$$[H_2O]_{Total} = [H_2O] + \frac{K_w}{[H^+]} + \frac{[H_3BO_3][H_2O]}{[H^+]K_{B(OH)_4^-}} - 2\frac{[H_3BO_3]^3}{[H^+][H_2O]K_{B_3O_3(OH)_4^-}}$$

$$[H^+]_{Total} = [H^+] - \frac{K_w}{[H^+]} - \frac{[H_3BO_3][H_2O]}{[H^+]K_{B(OH)_4^-}} - \frac{[H_3BO_3]^3}{[H^+][H_2O]K_{B_3O_3(OH)_4^-}}$$

$$[H_3BO_3]_{Total} = [H_3BO_3] + \frac{[H_3BO_3][H_2O]}{[H^+]K_{B(OH)_4^-}} + 3\frac{[H_3BO_3]^3}{[H^+][H_2O]K_{B_3O_3(OH)_4^-}}$$

$$[Ni^{2+}]_{Total} = \frac{[Ni^{2+}][Cl^-]^2}{K_{NiCl_2}} + \frac{[Ni^{2+}][SO_4^{2-}]}{K_{NiSO_4}} + [Ni^{2+}] + \frac{[Ni^{2+}][Cl^-]}{K_{NiCl^+}}$$

$$[SO_4^{2-}]_{Total} = \frac{[Ni^{2+}][SO_4^{2-}]}{K_{NiSO_4}} + [SO_4^{2-}]$$

$$[Cl^-]_{Total} = 2\frac{[Ni^{2+}][Cl^-]^2}{K_{NiCl_2}} + \frac{[Ni^{2+}][Cl^-]}{K_{NiCl^+}} + [Cl^-]$$

where: [X] = the concentration of component *X*
$K_{species}$ = the equilibrium coefficient of the reaction for the species.

At equilibrium, $C_{i\text{-}Total}$ will equal the amount of each component added in the recipe, $C_{i\text{-}Recipe}$ as shown:

$$C_{i-Total} = C_{i-\mathrm{Re}cipe} \text{ or } C_{i-Total} - C_{i-\mathrm{Re}cipe} = 0$$

where $C_{i\text{-}Recipe}$ is the amount of each component present by adding the amount of each component in each species added, or:

$$C_{i-\mathrm{Re}\ cipe} = \sum_{j}^{N_s} \alpha_{i,j} S_{j,inlet}$$

where S_j, inlet is the inlet concentration of species *j*. The set of mole balance equations defined by Equation 1 can be solved using the non-linear Newton-Raphson method (van der Lee 1998). In this case, we are trying to find the root of the set of functions f_i:

$$f_i = C_{i-Total} - C_{i-\mathrm{Re}cipe} = \sum_{j}^{N_a} \alpha_{i,j} K_j \prod_{i}^{N_{Cp}} c_i^{\alpha_{i,j}} - \sum_{j}^{N_s} \alpha_{i,j} S_{j,inlet}$$

The Newton-Raphson method updates the guess of the concentration of the component concentration using the following equation (Westerberg, Hutchison et al. 1979):

$$\overline{C}^{n+1} = \overline{C}^{n} - \overline{F}\left(\overline{C}\right)J^{-1}$$

where:

$\overline{C}^{n}$ = the nth iteration of the concentration vector containing the ci values.

$\overline{F}$ = a vector containing the values of the functions, fi.

J^{-1} = the inverse of the Jacobian matrix for the function.

The Jacobian matrix (J) is defined as follows:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial c_1} & \frac{\partial f_1}{\partial c_2} & \cdots & \frac{\partial f_1}{\partial c_{N_{cp}}} \\ \frac{\partial f_2}{\partial c_2} & \frac{\partial f_2}{\partial c_2} & \cdots & \frac{\partial f_2}{\partial c_{N_{cp}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{N_{cp}}}{\partial c_1} & \frac{\partial f_{N_{cp}}}{\partial c_2} & \cdots & \frac{\partial f_{N_{cp}}}{\partial c_{N_{cp}}} \end{bmatrix}$$

Each element of the Jacobian is calculated as follows:

$$\frac{\partial f_i}{\partial c_n} = \frac{\partial}{\partial c_n}\left(\sum_j^{N_a} \alpha_{i,j} K_j \prod_i^{N_{Cp}} c_i^{\alpha_{i,j}}\right) = \sum_j^{N_a} \alpha_{i,j} K_j \frac{\partial}{\partial c_n}\left(\prod_i^{N_{Cp}} c_i^{\alpha_{i,j}}\right)$$

The partial derivative of the product term can be found by assuming the concentrations of all components except the n[th] component are constants, using the power rule to obtain the derivative with respect to the concentration of the compound of interest. As an example, the derivatives of concentrations of the components with respect to [H^+] are calculated as follows:

$$\partial[H_2O]_{Total} \Big/ \partial[H^+] = -\frac{K_w}{[H^+]^2} - \frac{[H_3BO_3][H_2O]}{[H^+]^2 K_{B(OH)_4^-}} + 2\frac{[H_3BO_3]^3}{[H^+]^2[H_2O]K_{B_3O_3(OH)_4^-}}$$

$$\partial[H^+]_{Total} \Big/ \partial[H^+] = 1 + \frac{K_w}{[H^+]^2} + \frac{[H_3BO_3][H_2O]}{[H^+]^2 K_{B(OH)_4^-}} + \frac{[H_3BO_3]^3}{[H^+]^2[H_2O]K_{B_3O_3(OH)_4^-}}$$

$$\partial[H_3BO_3]_{Total} \Big/ \partial[H^+] = -\frac{[H_3BO_3][H_2O]}{[H^+]^2 K_{B(OH)_4^-}} - 3\frac{[H_3BO_3]^3}{[H^+]^2[H_2O]K_{B_3O_3(OH)_4^-}}$$

$$\partial[Ni^{2+}]_{Total} \Big/ \partial[H^+] = 0$$

$$\partial[SO_4^{2-}]_{Total} \Big/ \partial[H^+] = 0$$

$$\partial[Cl^-]_{Total} \Big/ \partial[H^+] = 0$$

The system of equations is then calculated iteratively until the system converges, which is defined as the point at which the differences between the concentrations of all components in two successive iterations has varied less than the convergence criteria, typically 1 percent.

5.3 Gaseous Emissions Calculations

Following calculation of the amount of material plated onto the part and the final concentration and speciation of the ions in solution, potential air emissions are estimated. Air emissions are estimated from the methodology utilized in MFFRST, which is based upon AP-42 (USEPA 1995). The following sections document the calculation procedure.

5.3.1 Electrolytic Processes

According to the AP-42 emssions factor for electroplating, emissions of metals and other components to the atmosphere from electrolytic (electroplating and anodizing) tanks are proportional to:

- The current density applied to perform the plating operation (CD);
- The inverse of the cathode efficiency (CE); and
- The concentration of the chemical component in the process tank (CC).

Cathode efficiency is the fraction of the applied electrical power that results in the deposition of metal on the substrate. For most processes, the cathode efficiency is greater than 90 percent. However, for the hard and decorative chromium plating process, the cathode efficiency is typically less than 20 percent. The proportion of the electrical power that does not result in metal deposition is used to decompose water into gaseous hydrogen and oxygen. As the hydrogen and oxygen bubbles rise to the surface of the tank and escape into the atmosphere, they can entrain a

substantial amount of the plating solution resulting in atmospheric emissions. The rate of gas evolution is a function of the chemical and electrochemical activity occurring in the tank, the strength and temperature of the solution and the current density in the tank. For hard chromium electroplating, the AP 42 chromium emission factor is 7.78 mg/A-hr, and controlled chromium emissions range from 0.0027 to 0.96 milligrams per dry standard cubic foot of air (mg/dscm) (USEPA 1995). The typical technique used to control emissions from a chromic acid plating tank include add on control devices and chemical fume suppressants. The most common add on control devices include mist eliminators and wet scrubbers. AP-42 presents the emissions rates from chromium electroplating processes incorporating these emissions control technologies.

According to AP-42, the emissions from non chromium electroplating are calculated using the following equation:

$$RC_C = \left[\frac{CC_C}{CC_{Cr}}\right]\left[\frac{CD_C}{CD_{Cr}}\right]\left[\frac{CE_C}{CE_{Cr}}\right]$$

where:

RC_c	= concentration of the contaminant over the plating bath relative to the concentration of hexavalent chrome
CC_c	= concentration of the contaminant in the plating bath
CC_{Cr}	= concentration of the hexavalent chromium in the plating bath
CD_c	= current density of the plating bath
CD_{Cr}	= current density of the hexavalent chromium plating bath
CE_c	= cathode efficiency of the contaminant plating bath
CE_{Cr}	= cathode efficiency of the hexavalent chromium plating bath

5.3.2 Non Electrolytic Processes

The emissions from non electrolytic tanks are determined based on the assumption that emissions are the result of turbulence caused by the use of compressed air to mix the tanks. As the mechanical agitation of mixing causes the emissions, tanks that are not mixed do not emit any of the inorganic compounds because little or no volatile materials are present in the electroplating process tank. Emissions from tanks not required by occupational safety

regulations to be vented outside the plant (e.g. acid etch/bright dip processes and phosphate coating) are released into the air within the plant and exit the plant into the surrounding community as fugitive emissions. Emissions from tanks mixed with air are calculated using the following equation (USEPA 1995):

$$E = \frac{1.9\sigma}{R_b}\left[\frac{(1-2a+9a^2)^{0.5}+(a-1)}{1+3a-(1-2a+9a^2)^{0.5}}\right]^{0.5}$$

where:

E = emissions factor in grains per cubic foot of aerated air
σ = surface tension of the bath in pounds force per foot (lbf/ft)
R_b = average bubble radius, inches
a = $\frac{0.072R_b^2}{\sigma}$

In the above equation, physical properties and unit conversion factors are incorporated into the constants, resulting in a dimensionally consistent equation. The above equation is used for tanks without emissions controls. For tanks that have emissions control, the emission rate is multiplied by the ratio of emissions from the chromium electroplating line with controls to the emissions of the chromium electroplating line without controls (Schwartz and Lorber 1999). This ratio would represent the fraction of the contaminant emitted by employing this emission control technology.

6.0 Model Verification Study

The MFFPPT process simulation model was verified against the four plating lines present at Leonhardt Plating Company (Leonhardt) in Cincinnati, Ohio. Leonhardt primarily plates decorative chrome onto steel parts. The decorative chrome process includes both nickel and chromium plating on the same plating process. The flow sheets for Leonhardt's plating lines are shown on Figures 7 through 10. The MFFPPT model was validated by comparison of model results with the actual conditions present in each line. A one-time sampling event was conducted to provide process chemistry data necessary to verify and validate a computer-based chemical process simulation program.

The sampling methodology was described in a Quality Assurance Project Plan (QAPP) entitled "Quality Assurance Project Plan for Verification of the Metal Finishing Facility Pollution Prevention Tool (MFFPPT)" dated February 7, 2006. Detailed quality assurance (QA) considerations for this project are documented in this QAPP, and the following section discusses the general sampling procedure and the results of the investigation.

This section will present an overview of the plating lines and the results of field and laboratory sample analysis. Section 7 will present a detailed discussion of the modeling of the plating process and a comparison of observed and modeled plating bath concentrations.

6.1 Plating Line Descriptions

The first step in model verification is obtaining a detailed description of the plating line and processes to which the parts are exposed. Leonhardt had four similar plating lines. Parts were typically steel. Prior to plating, each part was cleaned using alkaline cleaners, rinsed and

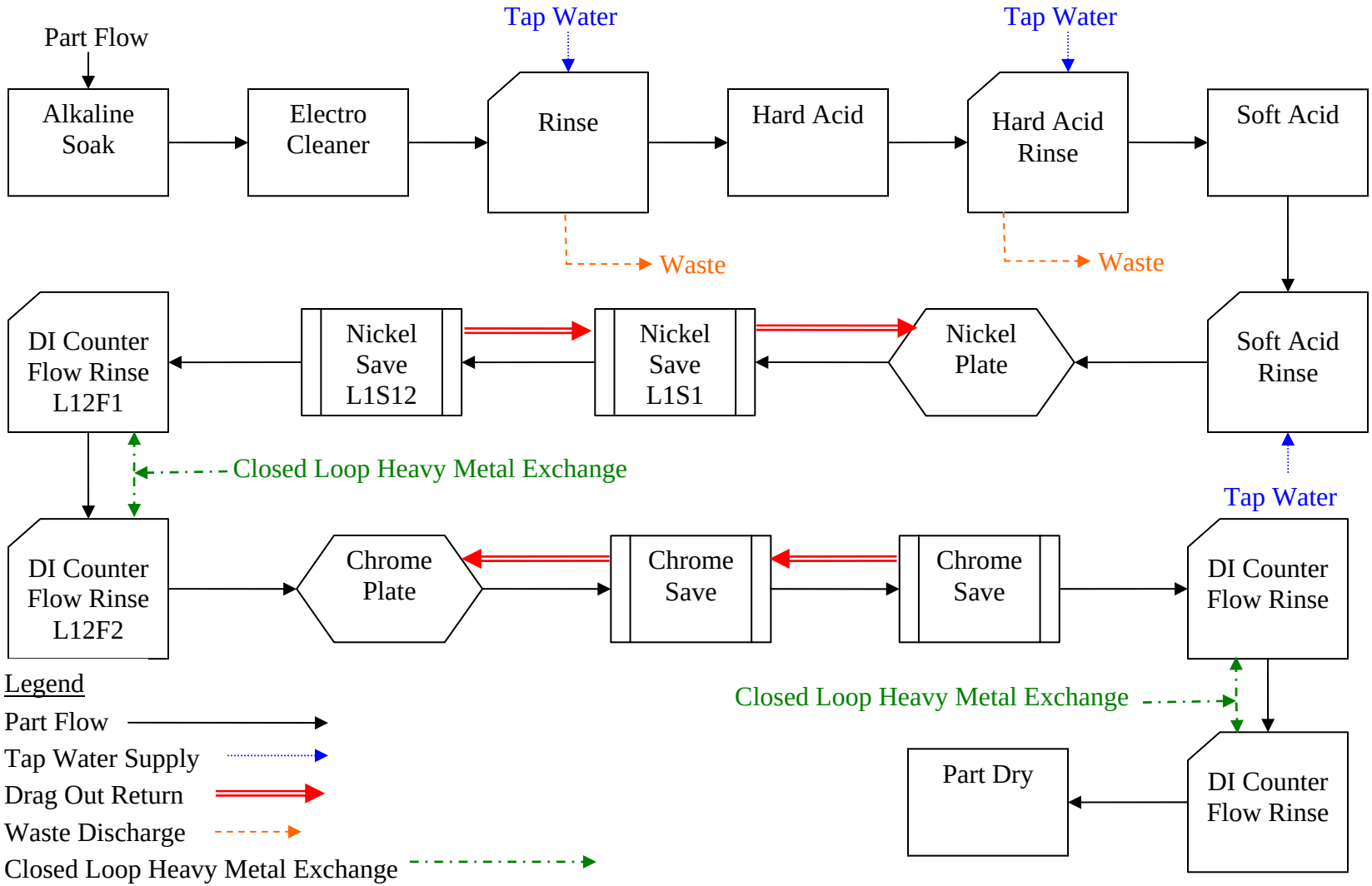


Figure 7. Flow Diagram of Leonhardt Plating Line Number 1.

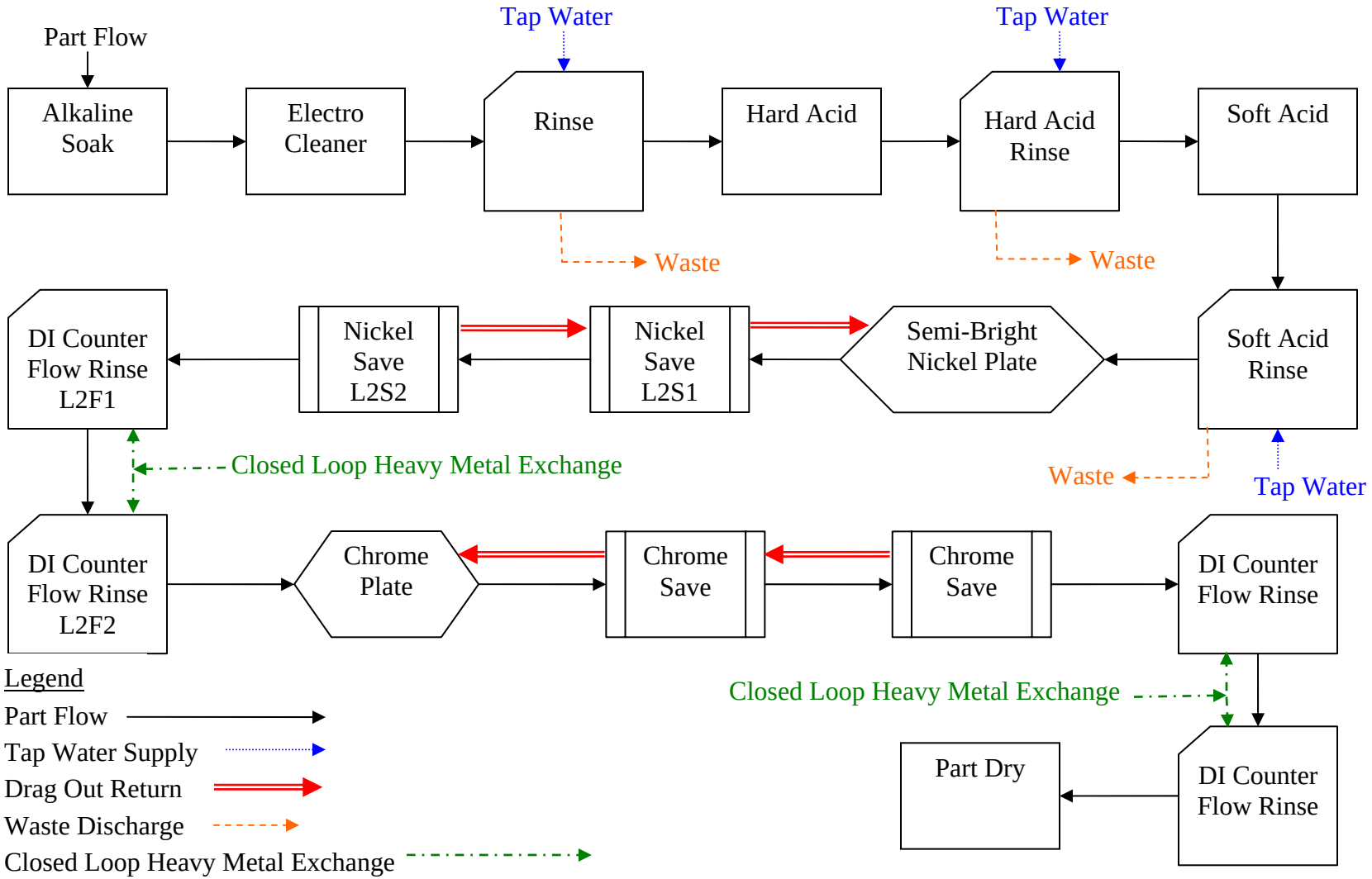


Figure 8. Flow Diagram of Leonhardt Plating Line Number 2.

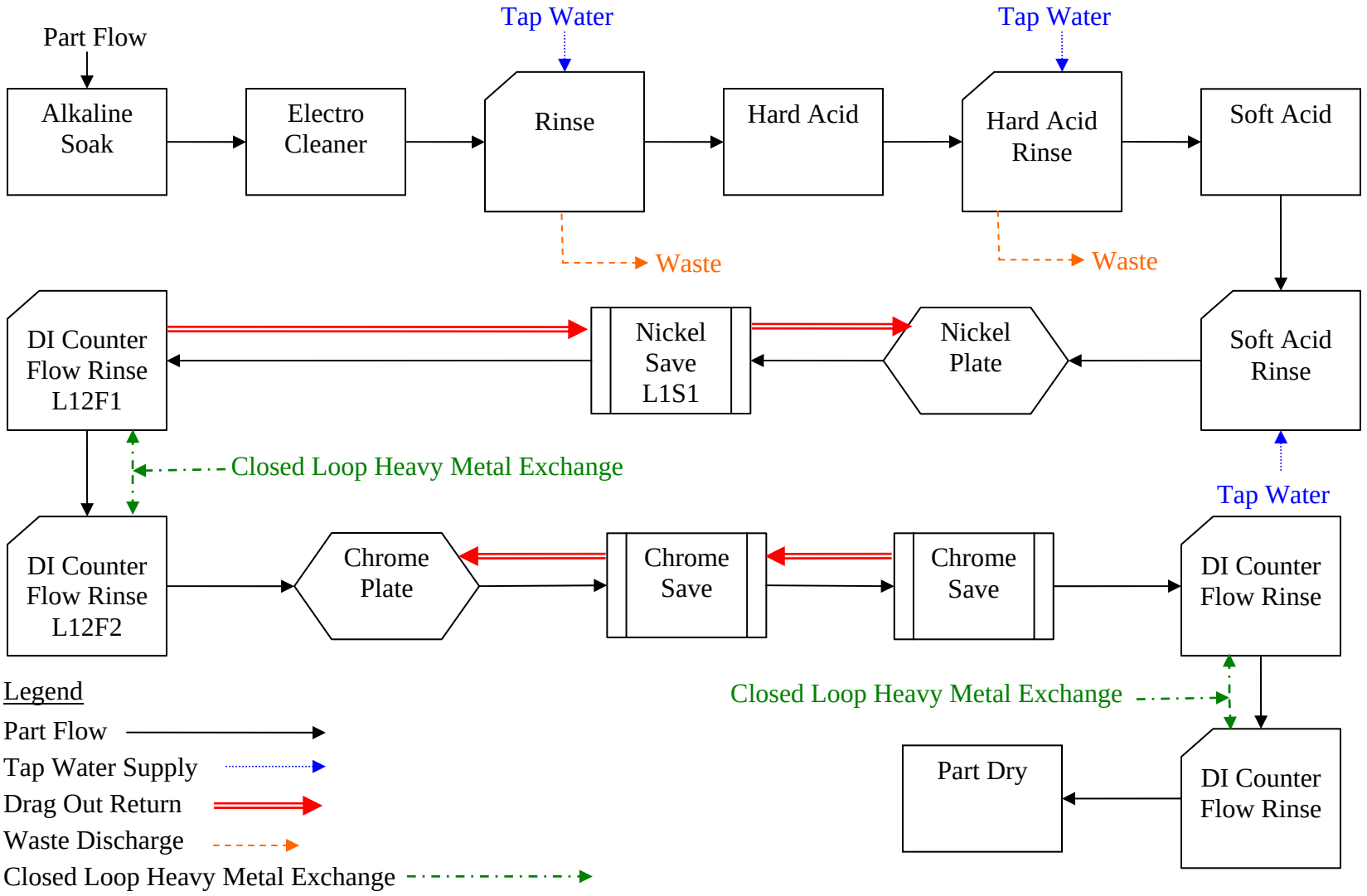


Figure 9. Flow Diagram of Leonhardt Plating Line Number 3.

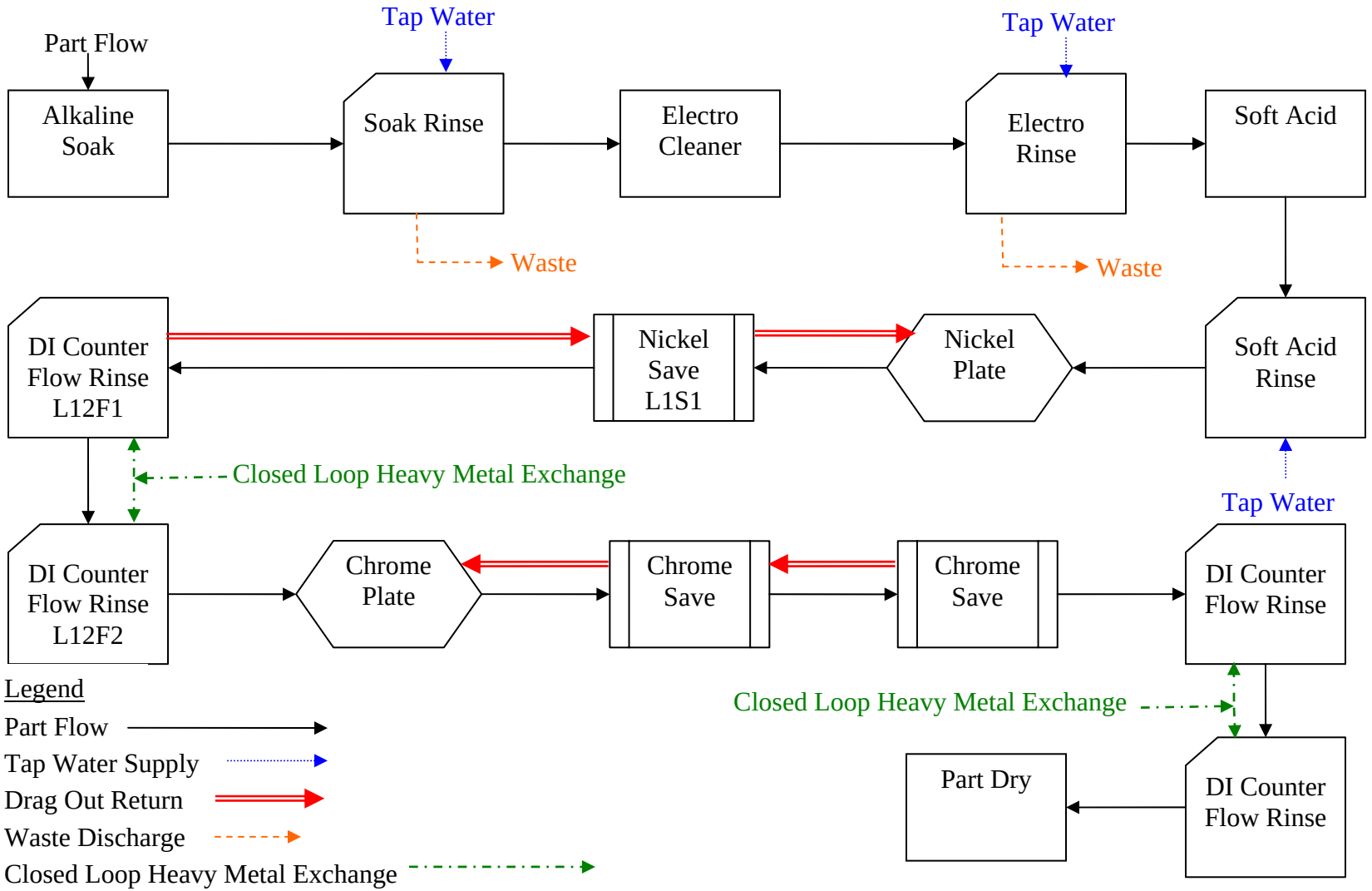


Figure 10. Flow Diagram of Leonhardt Plating Line Number 4.

then cleaned with both hard and soft acid cleaners. Following the acid cleaning, the parts were nickel plated in a Watts nickel bath. The plating solutions were rinsed from the part in a static rinse followed by a countercurrent rinse process. The nickel plated parts were then plated with chromium in a hexavalent chromium bath. Following chromium plating, the parts were rinsed and dried.

Wastewater generated by the process was treated using a closed-loop heavy metal ion exchange process. The closed-loop water was returned to the process lines for use in the post-plating rinse tanks. Makeup water used in the post-cleaning rinses was obtained from the Greater Cincinnati Water Works (GCWW) Department.

6.2 Sample Collection and Analysis

Samples were collected from each fluid flow and/or tank to obtain the process chemistry. At each sampling point, one sample was collected, unless the sample was identified as a duplicate, in which case, a second sample was collected.

None of the tanks had sample collection ports for the collection of samples. As a result, the samples were collected by dipping a new, precleaned Teflon® beaker into the tank to be sampled. The tanks at Leonhardt were small (typically less than 2 meters in any dimension) and mechanically agitated, providing a near uniform composition distribution, as a result, the samples obtained using the beakers were deemed to be well mixed. The sample was then transferred into appropriate laboratory-supplied sample containers for shipment to the laboratory. At the time of sample collection, every sample was assigned unique sample identifications, and then tested for pH, specific conductance, temperature, and oxidation reduction potential using the

methods described in Table 2. Samples were shipped to the laboratory, Environmental Testing and Consulting, Inc (ETC) of Memphis, Tennessee using FedEx overnight service.

Table 2. Sample Analytical Methods.

Analyte	Method[1]	
	Preparation	Analysis
pH	NA[2]	9040C
Total Dissolved Solids	NA[2]	SM2540C[3]
Total Alkalinity	NA[2]	SM2320[3]
Total Acidity	NA[2]	SM2310[3]
Hexavalent Chromium	NA[2]	7196A
Total Calcium	3010A	6010B
Total Chlorides	NA[2]	9253
Total Sodium	3010A	6010B
Total Nickel	3010A	6010B
Total Boron	3010A	6010B
Dissolved Calcium	3005A	6010B
Dissolved Sodium	3005A	6010B
Dissolved Nickel	3005A	6010B
Dissolved Boron	3005A	6010B

[1] Test Methods for Evaluating Solid Waste, Physical/Chemical Methods, Office of Solid Waste, United States Environmental Protection Agency, unless otherwise noted
[2]Not Applicable
[3]Standard Methods for the Examination of Water and Wastewater, APHA, AWWA, WEF.

6.2.1 Field Analysis

All samples were analyzed in the field to determine pH, specific conductivity, temperature, and oxidation-reduction potential using a field-portable electronic meter and sensor instrumentation. The field analytical instrument used for this project was a Thermo Electron Orion 5-Star Multiparameter Benchtop Meter. The instrument was calibrated and operated in accordance with manufacturer's instructions and published analytical methods. All calibration fluids were new. Calibration was documented in field notes.

6.2.2 Laboratory Analyses

Samples collected were submitted to Environmental Testing and Consulting, Inc (ETC) of Memphis, Tennessee for analysis of parameters as listed in Table 2. The laboratory reports indicated that results of one total dissolved solids sample and two alkalinity samples were

outside QC limits. These general parameters were collected to augment pH and ion concentration data, and were not used in the analysis of the model. These discrepancies have no impact on the results of the model verification.

The laboratory indicated that the recovery of nine sodium, four nickel and two boron matrix spike/matrix spike duplicates were outside the recovery limits. For eight of the nine sodium samples that had matrix spikes out of range, the spike level was inappropriate, the spike level was inappropriate, less than 15 percent of sample concentration. In the ninth sodium sample, the spike concentration was 5 mg/L, and the sample concentration was reported as below the detection limit of 2.5 mg/L. The actual spikes sample concentration was 7.21 mg/L. Based on the detection limit and spike level, no conclusion can be made about the recovery. In all four of the nickel samples that had spikes out of range, the spike level was inappropriate, less than 10 percent of the sample concentration. For the two boron samples outside of the recovery range, the actual spike recoveries were 136 and 142 percent, just outside of the QC limits of 75 to 125 percent. The laboratory conducted post digestion spikes and serial dilution in accordance with their quality assurance program. Spike recoveries for the post digestion analyses were within acceptance criteria. Based on the spike concentrations being inappropriate, or closeness to the QC criteria and the corrective actions taken by the laboratory, these discrepancies are not anticipated to have an adverse effect on the results of the study.

Results of field duplicate analyses are presented with the discussion of the model verification, including the impact of the duplicates on the results of the model verification.

7.0 Plating Line Model Results

This section presents a tank-by-tank comparison of the results of the model with the tank conditions observed during the model verification study. Information about the process conditions were obtained by discussions with shop owner, Joseph Leonhardt, and other plating facility personnel to determine process flows and operating conditions, process flow diagrams provided by Mr. Leonhardt, and a review of available material safety data sheets (MSDS) to determine process chemistry. It should be noted that some of the chemicals present in the plating lines are proprietary, and their chemical composition is not listed on their MSDS, and, as a result, the concentrations of these chemicals can not be predicted by the model and verified by standard chemical analytical techniques. These proprietary materials are typically a small proportion of the chemicals present in the plating operation, but their presence is noted for each process tank.

In general, the plating lines were similar, the rinse flows and input chemicals were, in general, identical between the plating lines. Slight differences were observed in the lines, such as the presence of an alkaline soak rinse tank in line 4 that was not present in lines 1, 2, or 3. Because of the similarity of the lines, a single plating line model was constructed based on plating line 1. The results of the model of this line will be compared to the conditions observed in each of the actual lines to judge model accuracy. In general, process inputs for the plating operations modeled are the output flow from the upstream plating operation, the results of the model focuses on the ability of the model to predict the effluents from the plating operations using the input chemical concentrations from the prior unit model. Chemical composition of process inputs materials, such as fresh cleaning and plating solutions are included in the description of each modeled process.

This section will discuss the results of the model and compare the observed effluent concentrations with the model predicted effluent concentrations. Where appropriate, chemical reaction equilibrium calculations are listed and discussed. The general operating conditions for the simulation assumed a part flow rate of 5 batches per hour and work schedule is 8 hours/day. The drag-out volume for each batch of parts was estimated to be 0.1 liters per batch. Rinse flows were typically 1.75×10^{-3} liters per second (approximately 40 gallons per day). The tanks in the plating line had capacities of approximately 1000 to 1200 liters (250 to 300 gallons). In order to conduct steady-state modeling, assumptions needed to be made regarding make-up solution additions, as follows: make-up solution flow rates were estimated to be 5×10^{-6} liter per second and static rinse (save rinse) flow rates estimated to be 1×10^{-8} liters per second.

7.1 Alkaline Cleaner Tank Model

The cleaning process began by placing the part in an alkaline cleaner bath. Alkaline cleaners are the most commonly used cleaners for removing oils, greases and general soils (Pavone 2002). The alkaline cleaner tank model consists of 4 ports, an inlet part port, an inlet solution port, an outlet part port and an outlet solution port. The model object conducted a material balance by combining the inlet solution with the drag-in on the part object. Once the solution concentration in the tank has been established, the aqueous equilibrium conditions were calculated. The resulting concentrations were set as the drag-out concentration and the solution outlet flow concentration. The flow rate for outlet solution was set equal to the flow rate for the solution inlet. Drag-out volume for the part inlet was set equal to assumed value of 0.1 liters per batch of parts.

The soak cleaner solution used at Leonhardt was Whirl-a-Way 207 manufactured by Derby Chemical, Inc. of Louisville, Ky. According to the MSDS, this solution consists of about

35 percent sodium hydroxide and contains other proprietary chemicals that were not specified on the MSDS. Based on the limited chemical data for this material, the component of interest is primarily the sodium ion. The concentrations of the aqueous input stream used for the alkaline cleaner model was 59,774 mg/L (2.6 moles per liter, molar) sodium and 17,750 mg/L (0.5 molar) chloride ion. Measured and modeled concentrations for the Alkaline Cleaner tank are listed in Table 3. The model estimated the concentration of sodium and chloride ions in line 1 almost exactly. Line 2 chloride concentration was again fairly accurate but sodium concentration was low. Lines 3 and 4 both had lower observed sodium and chloride concentrations than predicted. Additionally, solution pH model was within 0.1 units for 3 of the 4 lines. The modeled concentrations for plating line 1 were nearly identical to the observed concentrations for that line.

Table 3. Comparison of Modeled and Observed Solution Concentrations for the Alkaline Cleaner Tank.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	12.2	12.2	12.2	12.3	11.7
Sodium	58,158	52,800	37,000	34,000	32,700
Chloride Ion	17,270	18,500	18,500	3,710	5,560

7.2 Electrocleaner Tank Model

Each process line used an electrocleaning tank. Electrocleaning is a process where parts are placed in the tank with a low applied voltage to remove metal smuts and prevent deposition of non-adherent metal films. In general, the part itself is the anode, and because of the reverse current, the electrocleaning process works by removing a thin layer of metal from the surface of the part (Dargis 2006). The electrocleaner tank model consisted of 4 ports, an inlet part port, an

inlet solution port, an outlet part port and an outlet solution port. The model conducted a material balance by combining the inlet solution with the drag-in on the part object. Once the solution concentration in the tank has been established, the equilibrium conditions were calculated. The resulting concentrations were set as the drag-out concentration and the solution outlet flow concentration. The flow rate for the outlet solution was then set equal to the flow rate for the solution inlet. Drag-out volume for the part inlet was set equal to the volume dragged in with the part.

The electrocleaner solution is Whirl-A-Way EC, manufactured by Derby Chemical, Inc. of Louisville, Ky, which consists of about 60 percent sodium hydroxide and sodium metasilicate. The components of interest are primarily the sodium ion, and the solution input concentrations used were 54,486 mg/L (2.37 molar) sodium and 7,100 mg/L (0.2 molar) chloride. Measured and modeled concentrations for the Alkaline Cleaner tank. Plating line 4 has an alkaline cleaner rinse tank prior to the electrocleaner. Because of the high concentration of sodium in the electrocleaning solution, the presence of a rinse tank prior to the electrocleaner cleaner tank was not anticipated to have a significant effect the observed concentrations for the line 4 model. A comparison of model and observed concentrations is included in Table 4. As expected, the modeled concentrations in plating line 1 matched the observed concentrations from the method verification study. The other lines show minor differences between the modeled and actual concentrations due to changes in operating conditions for those lines.

No air emissions were calculated for the electrocleaner process.

Table 4. Comparison of Modeled and Observed Solution Concentrations for the Electrocleaner Tank.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	12.8	12.8	12.8	12.4	12.7
Sodium	54,585	54,500	60,900	28,000	40,900
Chloride Ion	7,374	7,410	1,850	44.5	141

7.3 Acid Cleaner Models

Two types of acid cleaner processes were used within the plating lines at Leonhardt; soft and hard acid rinses. This section will discuss the overall model of an acid cleaner tank model and provide specific results for modeling of each of the different rinse tank scenarios within the plating line.

Acid cleaner tank models consist of 4 ports, an inlet part port, an inlet solution port, an outlet part port and an outlet solution port. The model conducts a material balance by combining the inlet solution with the drag-in on the part object. Once the solution concentration in the tank has been established, the equilibrium conditions are calculated. The resulting concentrations are set as the drag-out concentration and the solution outlet flow concentration. The flow rate for outlet solution is set equal to the flow rate for the solution inlet. Drag-out volume for the part inlet is set equal to the volume dragged in with the part.

7.3.1 Hard Acid Cleaner Tank Model

Hard acid cleaning involves dipping the part into a hydrochloric acid solution. Acid dips are used to remove metal oxides from the surface of parts, also called pickling. Hydrochloric acid is a good pickling agent because it can be used at atmospheric temperatures and iron

chloride salts have high solubility (Snavely and Faust 1971). At the time of the model verification study, a hard acid rinse was only used in plating line 1 at Leonhradt. The hard acid solution was made by a 1 to 1 dilution of muriatic acid (another name for hydrochloric acid) in water. The MSDS for muriatic acid indicated that it was 26 to 37 percent hydrogen chloride, supplied by Reagent Chemical and Research, Inc. of Flemington, NJ. For the model, an input concentration of 35,500 mg/L (1 molar) was used. The comparison of actual and modeled concentrations is presented in Table 5. The observed chloride ion concentration and pH were about one-third the modeled concentrations, indicates either the bath was diluted more than required in the solution's recipe, or that drag-in from previous rinse operations diluted the cleaning solution.

No air emissions were calculated for the hard acid cleaning process.

Table 5. Comparison of Modeled and Observed Solution Compositions for the Hard Acid Cleaning Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)
		Line 1
pH	0.4	0.7
Sodium	273	1,510
Chloride Ion	14,897	11,100

7.3.2 Soft Acid Cleaner Tank Model

The soft acid cleaner tank used at Leonhardt Plating is a sodium bisulfate ($NaHSO_4$) solution, which forms a buffer, stabilizing the pH of the solution. The soft acid solution used at Leonhardt was Derby Assaults 595, manufactured by Derby Chemical, Inc. of Louisville, Ky. According to the MSDS, Derby Assaults 595 consists of <98% sodium bisulfate and <10% sodium hydrogen sulfate. The input solution concentration used for the model was 26,842 mg/L

(1.2 molar) sodium and 56,101 mg/L (0.6 molar) sulfate. Unfortunately, sulfate concentrations were not included as part of the laboratory analysis so no comparison could be made between actual and modeled sulfate concentrations. As an indication of the ability of the model to calculate buffer conditions, the parameter of interest for this tank is the solution pH. The modeled and observed pH values and sodium ion concetrations listed in Table 6 were consistent between the model and the observed values, with the model predicting a pH of 2.9 and the observed values in the range of 2.3 to 2.8

Table 6. Comparison of Modeled and Observed Solution Compositions for the Soft Acid Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	2.9	2.8	2.5	2.3	2.8
Sodium	26,482	35,100	15,500	12,700	13,600
Sulfate	56,101	NM	NM	NM	NM

NM = Not Measured.

7.4 Plating Tank Models

All four of the tested plating lines at the Leonhardt facility the application of a decorative chrome plate to a steel part. Decorative chrome plating consists of two different layers, first a nickel plate is applied to the surface. This nickel is covered with a thin, bright chrome layer, which is typically a 10 µm thick over an underlying nickel plate (Wikipedia 2007). The following sections will discuss the modeling of each of the nickel plating and chromium plating processes.

7.4.1 Nickel Plating Tank Model

At the Leonhardt facility, nickel plating was done using a Watt's nickel plating solution. The Watt's plating solution was developed by Professor Oliver Watt's at the University of Wisconsin in 1916, and consists of nickel chloride, nickel sulfate and nickel borate. Decorative nickel plating solutions differ from the original Watt's plating bath in that organic and metallic compounds have been added to brighten and level the nickel deposit (Di Bari 2000).

At the Leonhardt facility, the nickel plating solution consisted primarily of nickel sulfate, nickel chloride and boric acid. A nickel metal anode is present that serves to maintain the nickel concentration by providing nickel to the solution as nickel is plated on the part. In addition, the solution contained 1.5 percent of Ni Brightener 63 manufactured by Ethione-OMI, Inc (Ethione) of New Haven, Connecticut. NI Brightener 63 contains <15 percent sodium saccharine, and <0.1 percent formaldehyde. Additionally, the solution contains less than 1 percent Ethione's Ni Brightener 610 CFC-Free (<40 percent sodium allyl sufonate and <1 percent trade secret ingredients), Ethione's Wetting Agent 62A (<15 percent unspecified surfactants and <0.5 percent isopropyl alcohol); and trace quantities of Ethione's Ni Brightener 66E CFC-Free (<20 percent proprietary trade secret materials and <2 percent trade secret sulfonated ether butyne diol), Ethione's Curpalite 30 Additive (<10 percent trade secret polymer salt of organic acid, <1 percent ingredients not known to be hazardous, and <0.1 percent formaldehyde), and Ethione's Exylite HP Additive (1 to 5 percent 2-propene-1-sulfonic acid, sodium salt, 1 to 5 percent unspecified proprietary additives, and 1 to 5 percent boric acid). Because many of the additives occur in trace quantities in the solution, and the fact that these additives are difficult to determine analytically, the chemicals that will be modeled in the nickel plating tank are nickel (104,003 mg/L or 1.772 molar), chloride (32,815 mg/L or 0.9 molar), and boron (8,038 mg/L as boric acid or 0.13 molar).

Chemical reactions included in the model of the Watt's plating solution chemistry include:

$$H_2SO_4 \leftrightarrow 2H^+ + SO_4^{2-}$$
$$H_2O + H_3BO_3 \leftrightarrow H^+ + B(OH)_4^{-1}$$
$$H3BO3 \leftrightarrow H^+ + H_2BO_3^{2-}$$
$$HCl \leftrightarrow H^+ + Cl^-$$
$$NiOH^+ \leftrightarrow 1Ni^{2+} + 1OH^-$$

The resulting Morel Tableau is presented in Table 7. The results of the model, as shown in Table 8, indicate that the modeled concentrations observed in the plating lines were similar to the actual concentrations of these components observed in the laboratory analysis of the plating solution.

Table 7. Morel's Tableau Used in the Model of the Nickel Plating Tank.

Species	Reaction	Components				
		H^+	H_3BO_3	Ni^{2+}	SO_4^{2-}	Cl^-
H^+		1				
OH^-	$H2O \leftrightarrow OH^- + H^+$	-1				
HCl	$HCL \leftrightarrow H^+ + Cl^-$	1				1
HSO_4	$H_2SO_4 \leftrightarrow 2H^+ + SO_4^{2-}$	2			1	
H_3BO_3			1			
$B(OH)_4^-$	$H_2O + H_3BO_3 \leftrightarrow H^+ + B(OH)_4^{-1}$	-1	1			
$H_2BO_3^{2-}$	$H3BO3 \leftrightarrow H^+ + H_2BO_3^{2-}$	-1	3			
Ni^{2+}				1		
SO_4^{2-}					1	
$NiOH^+$	$NiOH^+ \leftrightarrow 1Ni^{2+} + 1OH^-$	-1		1		
Cl^-						1

As one would expect, the model of the plating tank is capable of conducting a material balance around the tank, and can make speciation calculations of the various acid/base and coordination aqueous chemical reactions that can occur in the nickel plating process. Presently, the chemistry and deposition mechanism of nickel from a plating bath is not well understood. One could proffer that use of a plating tank model capable of speciation calculations and

mechanics of the nucleation process at the part surface could improve the plating process and possibly result in lower waste generation by the process.

Air emissions from the tank were calculated to be 223,563 mg/day of nickel.

Table 8. Comparison of Modeled and Observed Solution Concentrations in the Nickel Plating Tank.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	3.4	3.6	4.1	4.3	4.5
Nickel	103,108	104,000	83,500	104,000	88,100
Boron	7,968	8,190	6,920	7,660	6,960
Chloride Ion	31,675	37,100	20,400	22,000	18,500

7.4.2 Chromium Plating Tank Model

Chromium was electroplated over the nickel-plated part following rinsing to remove nickel plating bath solution. The nickel plating provides the smoothness, much of the corrosion resistance, and most of the reflectivity. The chrome plate adds a bluish cast (compared to the somewhat yellowish cast of nickel), protects the nickel against tarnish, minimizes scratching, and symbiotically contributes to corrosion resistance (Mooney Accessed 2007).

The chromium plating bath at the Leonhardt facility consists of Ethione's Cromylite K-40-ZM (100 percent chromic acid , 1.5 percent of two trade secret fluoride compounds, and < 1 percent ingredients not known to be hazardous), Ethione Cromylite 107A (<5 percent fluoride as F, and <2 percent trade secret silicates), and barium carbonate. The barium carbonate was obtained from Ashland Chemical Company of Columbus Ohio. The modeled concentration of hexavalent chromium (chromic acid) in the influent was 116,000 mg/l (1 molar) as chromate ion, and the input sulfate concentration was 1,962 mg/L (0.02 molar) as sulfuric acid. The modeled

Table 9. Comparison of Modeled and Observed Solution Concentrations from the Chromium Plating Tank.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	1.5	1.3	NM	1.3	1.7
Hexavalent Chromium	115,001	96,800	NM	110,000	120,000
Sulfate	1,905	NM	NM	NM	NM
Nickel	0.066	4,820	NM	2,710	2,650
Boron	0.0051	935	NM	640	621
Sodium	$1.2x10^{-6}$	2,710	NM	1,580	1,950
Chloride Ion	0.020	18,500	NM	1,480	3,710

NM = Not Measured

concentration of hexavalent chrome corresponded well with the actual concentrations reported from the tank contents, as shown in Table 9. Nickel, boron, sodium and chloride were present at concentrations well above the modeled concentrations. According to the post-nickel plating rinse results discussed below, these materials were removed in the rinses following the nickel plating tanks. For this reason, the elevated concentrations of these materials were unexpected and likely present either present in the chromium acid solution or dissolved from the part by exposure to the relatively strongly acidic chromic acid plating solution.

Air emissions from the chromium plating process were estimated to be 85,460 mg/day of chromate ion (CrO_4^{2-}).

7.5 Rinse Tank Models

Rinse tanks occur at various locations in the plating lines. This section discusses the general mathematical model used for a rinse tank and provides a comparison of the model effluent concentrations with the actual concentrations measured for each of the different rinse tank scenarios within the plating line.

Rinse tank models consist of 4 ports, an inlet part port, an inlet solution port, an outlet part port and an outlet solution port. The model conducts a material balance by combining the inlet solution with the drag-in on the part object. The inlet solution can either be tap water, return water from the closed-loop heavy metal exchange system, or from an upstream counter-current rinse tank. Because of the low rinse flow rates, in the case of tap water or return from the closed loop ion exchange system, the input is assumed to be pure water. In the case where the input is from a counter-current rinse system, effluent from the upstream tank is used to verify the process simulation package's ability to tear the recycle stream on the flowsheet and converge the calculation result.

Once the solution concentration in the tank has been established, the equilibrium conditions are calculated. The resulting concentrations are set as the drag-out concentration and the solution outlet flow concentration. The flow rate for outlet solution is set equal to the flow rate for the solution inlet. Drag-out volume for the part inlet is set equal to the volume dragged in with the part.

7.5.1 Electrocleaner Rinse

The electrocleaner rinse tank simulation models the rinse tank that follows the electrocleaner unit operation. The dragout solution concentrations modeled from the electrocleaner had a pH 12.8, sodium concentration of 54,585 mg/L, and chloride concentration of 7,374 mg/L. The resulting modeled effluent concentrations from the electrocleaner rinse tank and the actual measured concentrations are listed in Table 10. The modeled concentrations were consistent with the actually observed concentrations.

Table 10. Comparison of Modeled and Observed Solution Concentrations for the Electrocleaner Rinse Tank.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	10.7	9.7	11.4	10.7	11.5
Sodium	470	148	438	192	282/277[1]
Chloride Ion	63	44.5	48.2	40.8	40.8/40.8[1]

[1] Line 4 electrocleaning rinse sample was a field duplicate. Results of both laboratory analysis are shown.

7.5.2 Hard Acid Rinse

The hard acid rinse tank simulation models the rinse tank that follows the hard acid pretreatment unit operation. Use of a hard acid cleaning and rinse was only done in plating line 1. The dragout solution concentrations modeled from the hard acid cleaning process had a pH of 0.5, and a chloride concentration of 14,866 mg/L, while the observed chloride concentration dragout concentration was 11,100 mg/L. The modeled dilution was a factor of 116, while the actual dilution was only 15. This indicates that the measured flow rate likely was less than the modeled flow, resulting in a greater effluent concentration. The resulting modeled effluent concentrations from the hard acid rinse tank and the actual measured concentrations are listed in Table 11.

Table 11. Comparison of Modeled and Observed Solution Concentrations for the Hard Acid Rinse Tank.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)
		Line 1
pH	2.5	2.1
Chloride Ion	128	741

7.5.3 Soft Acid Rinse

The soft acid rinse tank simulation models the rinse tank that follows the soft acid cleaning unit operation. The dragout solution concentrations modeled from the soft acid had a pH 2.9, and a modeled sulfate concentration of 60.5 mg/L. The resulting modeled effluent concentrations from the electrocleaner rinse tank and the actual measured concentrations are listed in Table 12. The modeled pH for the soft acid rinse was below the observed pH, but this may be due to the low buffering capacity of the modeled soft acid rinse solution.

Table 12. Comparison of Modeled and Observed Solution Compositions for the Soft Acid Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	5.5	3.3	2.5	2.8	2.8
Sodium	231	84.1	264	134/120[1]	187
Sulfate	482	NM	NM	NM	NM

[1] Line 2 soft acid rinse sample was a field duplicate. Results of both laboratory analysis are shown.

7.5.4 Nickel-Plating Static Rinse

The nickel-plating static rinse tank simulation models the static rinse tank that immediately follows the nickel plating unit operation. A static rinse tank does not have any rinse water flowing through it; instead the concentration of the rinse changes with time as parts are rinsed in it. This is not the situation simulated by a steady state process model, but can be simulated for discrete times by assuming a low flow rate. This would essentially indicate a change-out frequency for the tank that would maintain the observed rinse concentration. The drag-out solution concentrations modeled from the nickel plating process had a pH of 3.4, nickel concentration of 103,108 mg/L, a boron concentration of 7,968 mg/L and a chloride

concentration of 31,675 mg/L. The resulting modeled effluent concentrations from the nickel plating static rinse tank and the actual measured concentrations are listed in Table 13. The differing concentrations found in the rinse tanks of the various plating lines indicate different period since last rinse replacement and static rinses should be modeled as time-dependent processes.

Table 13. Comparison of Modeled and Observed Solution Compositions for the Nickel Plating Static Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	4.2	3.6	3.9	3.5	4.5
Nickel	103,034	829/780	1,330	1,820	88,100
Boron	7,962	92.9/90.6	161	169	6,960
Chloride Ion	31,652	11,000/3,710	927	927	18,500

[1] The Line 1 Nickel-Plating Static Rinse sample was a field duplicate. Results of both laboratory analysis are shown.

7.5.5 Nickel-Plating Counter-Current Rinse

The nickel-plating counter-current rinse tanks simulation models the rinse tanks that follow the nickel plating unit operation. Counter-current rinses involve two rinse tanks, the first tank that the part enters receives the part directly from the static rinse and the second rinse tank receives the part after the first rinse. The rinse solution first enters the second tank that the part is rinsed in. Effluent solution from the second rinse tank is used to rinse the part in the first rinse tank. This process reduces the amount of water required to rinse the part and therefore results in a cleaner part with less wastewater generated. The intermediate and final rinse solution concentrations are listed in Table 14and Table 15, respectively.

Table 14. Comparison of Modeled and Observed Intermediate Solution Compositions for the Nickel Plating Counter-Current Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	4.2	3.6	NM	3.1	3.1
Nickel	894	22.7	NM	3.24	21.7
Boron	69	14.7	NM	13.3	13.8
Chloride Ion	274	741	NM	14.8	25.9

Table 15. Comparison of Modeled and Observed Final Solution Compositions for the Nickel Plating Counter-Current Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	4.2	3.4	2.9	3.0	2.9
Nickel	7.6	4.30	0.477	0.281	4.03
Boron	0.6	12.0	12.1	12.4	12.2
Chloride Ion	2.4	371	81.5	18.5	25.9

The ratios of the drag out concentration from the nickel static rinse tank to the effluent concentration of the first stage of the counter rinse to the second stage of the counter rinse are listed below:

Nickel Predicted	103,108	:894	:7.6	or	13,566	:117	:1
Nickel line 1	829/780	:22.7	:4.3	or	187	:5.3	:1
Nickel line 3	1820	:3.24	:0.281	or	6,476	:11.5	:1
Nickel line 4	88,100	:21.7	:4.03	or	21,861	:5.38	:1
Boron Predicted	7,962	:69	:0.6	or	13,270	:115	:1
Boron line 1	92.9/90.6	:12.1	:12.2	or	7.5	:0.99	:1
Boron line 3	169	:13.3	:12.4	or	13.6	:1.07	:1
Boron line 4	6,960	:13.8	:12.2	or	570	:1.13	:1
Chloride Predicted	31,652	:274	: 2.4	or	13,188	:114	:1
Chloride line 1	11,000(3,700)	:741	:371	or	19	:2.0	:1
Chloride line 3	927	:14.8	:18.5	or	50	:0.8	:1
Chloride line 4	18,500	:25.9	:25.9	or	714	:1	:1

In each phase of the model, the concentrations decreased by the dilution factor of about 115, or about 115^2 (13,225) for both stages of the counter-current rise. The lower concentration ratios observed in the actual plating lines in general indicate that the time dependence of the effluent concentrations from the static rinse resulted in lower observed influent concentrations to the rinses than predicted. Additionally, none of the plating lines followed the X^2:X:1 ratio anticipated from a material balance conducted for the modeled line.

The observed concentrations are inconsistent with the modeled ones and the results that would have been anticipated based upon the conduct of a simple mass balance around each of the process tanks. It appears that the actual flow rate differs from the modeled flow rate, but the dilution of each chemical present between the tanks is not even consistent, as would be expected if the flow was known. To resolve the differences between the model and the actual concentrations, better flow measurements and incorporation of the time dependence of the process into the model would be required.

7.5.6 Chromium Plating Static Rinse

The chromium plating static rinse tank simulation models the static rinse immediately follows the chromium plating unit operation. The static rinse tank does not have any rinse water flowing through it; instead the concentration of the rinse changes with time as parts are rinsed in it. This is not the situation simulated by a steady state process model, but can be simulated for discrete times by assuming a low flow rate. This would essentially indicate a change-out frequency for the tank that would maintain the observed rinse concentration. The drag-out solution concentrations modeled from the chromium plating process had a pH of 1.4, hexavalent chromium concentration of 114,919 mg/L, a sulfate concentration of 1,904 mg/L, a nickel concentration of 0.066 mg/L, a boron concentration of 0.0051 mg/L, a sodium concentration of

$1.3x10^{-6}$ mg/L, and a chloride concentration of 0.020 mg/L. The resulting modeled effluent and actual measured concentrations from the chromium static rinse tank are listed in Table 16. The modeled chromium concentrations were greater than the actual observed chromium concentrations because the steady state process model does not consider the time dependency of solution concentration in a static tank. The chromium concentration difference could be corrected by either changing the model to account for time dependencies, or using a different assumption for the rinse flow rate.

The presence of nickel, boron, sodium, and chloride at concentrations significantly greater than the predicted concentrations, is consistent with their elevated concentrations in the chromium plating tank, and results from either their undocumented presence in the chromium plating solution or dissolution of these chemical from the part in the chromium plating tank.

Table 16. Comparison of Modeled and Observed Solution Compositions for the Chromium Plating Static Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	1.4	1.5	1.7	1.4	1.9
Hexavalent Chromium	114,919	34,200	7,790	16,500	14,400
Sulfate	1,904	NM	NM	NM	NM
Nickel	0.066	644	366	438	356
Boron	0.0051	156	94.5	145	109
Sodium	$1.3x10^{-6}$	387	<250	338	297
Chloride Ion	0.020	9,270	3,710	3,710	1,850

7.5.7 Chromium Plating Counter-Current Rinse

The chromium-plating counter-current rinse tanks simulation models the rinse tanks that follow the chromium plating unit operation. Counter-current rinses involve two rinse tanks, the

first tank that the part enters receives the part directly from the static rinse and the second rinse tank receives the part after the first rinse. The rinse solution first enters the second tank that the part is rinsed in. Effluent solution from the second rinse tank is used to rinse the part in the first rinse tank. This process reduces the amount of water required to rinse the part and therefore results in a cleaner part with less wastewater generated. The intermediate and final rinse solution concentrations are listed in Table 17 and Table 18, respectively. The modeled effluent chromium concentrations from the intermediate rinse tank was larger than the actual measured chromium concentration, but the final rinse tank concentration was 0.22 mg/L, which was in the same range as the actual concentrations of 0.23 to 0.94 mg/L. The presence of nickel, boron, sodium, and chloride at concentrations significantly greater than the predicted concentrations was consistent with their elevated concentrations in the chromium plating tank and static rinse tank as described above.

Table 17. Comparison of Modeled and Observed Intermediate Solution Compositions for the Chromium Plating Counter-Current Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	4.3	3.3	2.9	2.8	2.7
Hexavalent Chromium	160	7.45	21.2	40.6	50.1
Sulfate	2.6	NM	NM	NM	NM
Nickel	$9.2x10^{-5}$	0.368	1.32	1.01	1.66
Boron	$7.1x10^{-6}$	16.6	17.6	17.7	17.8
Sodium	$1.7x10^{-9}$	27.6	32.2	<25	<25
Chloride Ion	$2.8x10^{-5}$	185	1,480	185	14.8

Table 18. Comparison of Modeled and Observed Final Solution Compositions for the Chromium Plating Counter-Current Rinse Process.

Component	Modeled Concentration (mg/L)	Observed Concentrations (mg/L)			
		Line 1	Line 2	Line 3	Line 4
pH	6.8	3.4	3.4	3.2	3.3
Hexavalent Chromium	0.22	0.387	0.230	0.842	0.942
Sulfate	0.004	NM	NM	NM	NM
Nickel	$1.3x10^{-7}$	0.0780	<0.025	0.0815	0.166
Boron	$9.9x10^{-9}$	17.5	17.4	16.9	15.5
Sodium	$2.5x10^{-12}$	<2.5	<2.5	<2.5	<25
Chloride Ion	$3.9x10^{-8}$	148	7.41	7.41	7.41

8.0 Conclusions and Recommendations

MFFPPT has been used to model a metal finishing process and the results of the model were compared with the concentrations of the different plating and rinse solutions within the facility. Comparison of the MFFPPT model outputs with the observed concentration indicate that the available level of detail known about a typical plating system is lacking, limiting the ability of the model to predict the concentrations and flows within the plating line. Clearly use of a tool such as this would require more in depth process data, such as detailed disclosure of plating solution compositions and installation of accurate flow measuring devices within the facility.

The steady state modeling in MFFPPT is useful for modeling processes that have relatively stable input and output flow rates, which are most of the processes that are present in a metal finishing facility. The model did not perform well on the static rinse processes present at the facility. Accurate modeling of these processes will require a dynamic process simulation package.

The model of various processes in the plating operation provided not just material balance information presented above to verify the model, but also aqueous speciation data such as metal ion complex concentrations and acid/based buffer pair concentrations. This information can be useful in preparing more detailed surface cleaning and metal plate deposition models of the finishing process and can lead to improved understanding of the chemical interactions occurring in the processes and improved plating operations.

The ability of the model to calculate concentrations of various chemicals in different waste streams can be used to optimize process flows to reduce the quantity of wastewater generated by the process. As better models of the cleaning and plating processes are developed

and implemented, the model can be used to evaluate potential processes that can be used in place of process in the plating line to identify which process replacements can be used to reduce the amount of wastewaters generated.

In order to improve the modeling of the processes, improvements to the modeling package, including improving the aqueous property package to include newer aqueous equilibrium models, and a greater coverage of ions and reactions should be considered. Further, the model retains chemicals having concentrations well below the ability to measure their presence, resulting in longer calculation times. Chemicals with concentrations below $1x10^{-18}$ molar should be deleted from the equilibrium calculations to reduce calculations time.

The MFFPPT model is built around a chemical process simulation package that incorporates the CAPE-OPEN process modeling standards. AmsterChem's TEA property package and COUS unit operations models have been tested and found to operate successfully within this process simulation environment using the CAPE-OPEN interfaces. This means that the process simulation can be widely used to model general chemical processes used in a wide range of industries.

Further, because the model allows users to obtain material flow data within the process, and share that data using a well-defined information technology-based manner, the process simulation package can provide detailed process-specific data for material flow analysis (MFA) and life cycle assessment (LCA). In particular, an internet-based MFA/LCA can be developed where specific process models can be used in the material flow or life cycle evaluation instead of general industry-wide data as is the current practice.

9.0 References

Ballard, P. (2005). "Give Your Everyday Custom Collections a Design-Time Makeover." MSDN Magazine **20**(8): 68-78.

Belaud, J.-P. and D. Piñol (2000). Open Interface Specification: Identification Common Interface, Version 2, Global CAPE-OPEN

CAPE-OPEN (2000). Conceptual Design Document 2 (CDD2) for CAPE-OPEN Project, Global CAPE-OPEN.

Clark, J. (2003). "Plug-Ins: Let Users Add Functionality to Your .NET Applications with Macros and Plug-Ins." MSDN Magazine **18**(10).

CO-LaN (2001). CAPE-OPEN Open Interface Specifications: Unit Operations, Version 2.0, CAPE-OPEN Laboratories Network.

CO-LaN (2002). CAPE-OPEN Open Interface Specifications: Thermodynamic and Physical Properties, Version 1.0, CAPE-OPEN Laboratories Network.

CO-LaN (2003). Interface Specification: Parameter Common Interface, Version 5, CO-Laboratories Network.

CO-LaN (2003). Open Interface Specification: Collection Common Interface, CAPE-OPEN Laboratories Network.

CO-LaN (2003). Open Interface Specification: Utilities Common Interface, Version 2.0, CAPE-OPEN Laboratories Network.

CO-LaN (2003). Open Interface Specifications: Chemical Reactions Interface, Version 2, CAPE-OPEN Laboratories Network.

CO-LaN (2003). Open Interface Specifications: Error Common Interface, Version 7, CAPE-OPEN Laboratories Network.

CO-LaN (2003). Open Interface Specifications: Identification Common Interface, Version 3, CAPE-OPEN Laboratories Network.

Dargis, R. (2006, February 1, 2006). "Better Electrocleaning " Retrieved January 30, 2007, 2007, from http://www.pfonline.com/articles/0206qf2.html.

Di Bari, G. A. (2000). Electrodeposition of Nickel. New York, John Wiley and SOns, Inc.

Edwards, J. O. (1953). "Detection of Anionic Complexes by pH Measurements. I. Polymeric Borates." Journal of the American Chemical Society **75**(24): 6151 - 6154.

Mackenzie, D. (2002). "Creating a Design Surface Using Windows Forms and GDI+ in Microsoft .NET." from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/designsurface.asp.

Marquardt, W. (1996). "Trends in computer-aided process modeling." Computers & Chemical Engineering **20**(6-7): 591-609.

Microsoft "Visual C++ : Pure and Verifiable Code." Microsoft Developers Network **Volume**, DOI:

Mooney, T. (Accessed 2007). "FAQ: Chrome Plating Tutorial." Retrieved January 30,2007, 2007, from http://www.finishing.com/faqs/chrome.html.

Morel, F. M. M. and J. G. Hering (1993). Principles and Applications of Aquatic Chemistry. New York, Wiley-Interscience.

Obermeyer, P. and J. Hawkins. (2002). "Object Serialization in the .NET Framework, Microsoft Developers Network." from

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/objserializ.asp.
Pacific Environmental Services, I. (2000). Nation Metal Finishing Environmental Research and Development Plan: An Update. U. S. E. P. Agency, U.S. Environmental Protection Agency.
Paunovic, M. and M. Schlesinger (1998). Fundamentals of Electrochemical Deposition. New York, John Wiley & Sons, Inc.
Pavone, V. (2002, January 15, 2002). "Aqueous Cleaning Basics." Retrieved January 30, 2007, 2007, from http://www.pfonline.com/articles/010202.html.
Rogerson, D. (1997). Inside COM. Redmond, Washington, Microsoft Press.
Schwartz, S. and M. Lorber (1999). Characterizing Site-Specific Source Emissions for EPA's Risk Assessment Tool for the Metal Finishing Industry. AESF/EPA Conference for Environmental Excellence, Orlando.
Snavely, C. A. and C. L. Faust, Eds. (1971). Oxide Removal. Electroplating Engineers Handbook. New York, Van Norstrand Reinhold Company.
Tarjan, R. (1972). "Depth-First Search and Linear Graph Algorithms." SIAM Journal on Computing **1**(2): 146-160.
USEPA (1995). AP-42: Compilation of Air Pollutant Emission Factors Volume I: Stationary, Point and Area Sources, US Environmental Protection Agency.
USEPA (2001). The Metal Finishing Facility Risk Screening Tool (MFFRST): Technical Documentation and User's Guide. U. S. E. P. Agency, U.S. Environmental Protection Agency.
van der Lee, J. (1998). Thermodynamic and mathematical concepts of CHESS. Fontainebleau, France, CIG-École des Mines de Paris.
Varma, G. V., K. H. Lau, et al. (1993). "A new tearing algorithm for process flowsheeting." Computers & Chemical Engineering **17**(4): 355-360.
Westerberg, A. W., H. P. Hutchison, et al. (1979). Process Flowsheeting. Cambridge, Cambridge University Press.
Wikipedia. (2007, January 30, 2007). "Chrome plating." Wikipedia, from http://en.wikipedia.org/wiki/Chrome_plating.