CAPE-OPEN

Delivering the power of component software
and open standard interfaces
in Computer-Aided Process Engineering

Open Interface Specification: Parameter Estimation and Data Reconciliation Interface

CO-LaN

www.colan.org

ARCHIVAL INFORMATION

Filename	Parameter Estimation and Data Reconciliation Interface Specification.doc
Authors	CO-LaN consortium
Status	Public
Date	August 2003
Version	version 2
Number of pages	39
Versioning	version 2, reviewed by Jean-Pierre Belaud, August 2003
	version 1, December 2001
Additional material	
Web location	www.colan.org
Implementation specifications version	CAPE-OPENv1-0-0.idl (CORBA)
Comments	

IMPORTANT NOTICES

Disclaimer of Warranty

CO-LaN is a non for profit organization established under French law of 1901.

Trademark Usage

SUMMARY

ACKNOWLEDGEMENTS

CONTENTS

LIST OF FIGURES

1. Introduction

1.1 The Data Reconciliation problem

Reliable and accurate process measurements are key aspects to improve efficiency in chemical plant operation. It should be common practice to condition process measurements, so that measurement noise or other type of errors that affect data accuracy are eliminated or compensated. In general, data inherently contain inaccurate information since measurements are obtained with imperfect instruments. These errors could be categorized in two main classes:

- **Random errors:** this type of error could be arising from fluctuations and/or disturbances in non-controlled conditions that imply an irreproducibility of measurement. These errors are normally distributed with zero means and a known variance-covariance matrix. Measurements with such errors are not consistent with the process model constraints.

- **Non-random errors or Gross errors:** There is no relationship between the real and the observed value. This type of error could be generated from an uncorrected calibration or installation of instruments for example. These errors are arising from non-random events and they could be subdivided in (1) Measurement-related errors: malfunctioning instruments (2) Process-related errors: process leaks.

The Data Reconciliation (DR) is a procedure that makes the raw measurements to be consistent with the conservation's laws and other exact constraints, so that the random errors are eliminated and the variable variance is reduced. However the presence of gross errors can invalidate the criteria of DR. In order to use safely DR it is essential to check for the presence of gross errors in the measured data and correct or either deletes the related measurements.

The elimination of gross errors may be performed once they are detected by means of statistical tests. Most of the statistical tests for Gross Error Detection (GED) address linear process and can be applied to non-linear processes only after suitable linearisation of the process constrains. Nevertheless, tests that do not require linearisation of process constraints have recently appeared.

Data Reconciliation can be stated as a weighted no linear least square optimization problem subject to a set of equality and inequality constraints as is shown in Eq (1). The equality constraints are in general the mass, component and energy balance or any other accurate process model. The inequality constraints are the variables bound.

$$
\begin{aligned}
&\underset{x}{Min} \quad (x_m - x)^T Q_x^{-1} (x_m - x) \\
&s.t. \\
&f(x, \dot{x}) = 0 \\
&x_{LB} \le x \le x_{UB}
\end{aligned} \qquad (1)
$$

Where:

Qx = variance-covariance matrix of random measurement error of size [n_x, n_x].

xm = vector of measured values of process variables of size [n_x,1].

$\boldsymbol{x}$ = vector of adjusted values of process variables of size $[n_x,1]$.

$\boldsymbol{f}$ = vector of process constraints (linear or non-linear) of size $[n_q,1]$.

The objective of DR is the adjustment of process measurements to compensate for random errors. The DR procedure consists of the following steps:

(i) Collecting the raw process measurement either from a database or from a Distributed Control System (DCS).

(ii) Improving their accuracy by reducing the discrepancy between them and the conservation balance equations.

(iii) Storing the adjusted data or sending them to other application.

1.2 The Parameter Estimation problem

Another fundamental aspect of this workpackage is the parameter estimation (PE). PE is a common problem in many areas of process modeling, both in on-line applications such as real time optimization and in off-line applications such as the modeling of reaction kinetics and phase equilibrium for example. The goal is to determine values of model parameters that provide the best fit to measured data, generally based on some type of least squares or maximum likelihood criterion. In the most general case, this requires the solution of a nonlinear optimization problem.

With parameter estimation, one takes data from one or more experiments and attempts to find the values of the model parameters, which match the model as closely as possible to the experimental data. There are two general purposes of parameter estimation:

- **Testing a model against experimental data.** One test for a simulation model is how well its predictions match experimental data. If the model cannot be tuned to give acceptable predictions, then it may not represent for instance physical reality accurately.

- **To estimate quantities difficult or impossible to measure directly.** In chemical engineering, there are often rate constants whose values can only be inferred from other measurable quantities.

1.3 The Data Reconciliation and Parameter Estimation Problem similitude

The DR and PE are very similar problems in the sense that both are constrained optimization problems. They have the same input information, a process model (a set of equations), an objective function and a set of measurements (measured variables). Nevertheless, the set of measured data for PE is assumed to be free of any error, but for DR it is assumed to be only affected by random error. Furthermore, a variable classification has to be performed to evaluate the index of solvability (degrees of freedom) of the problem: if this index is zero PE may be performed; if it is greater than zero there is a certain degree of redundancy that allows DR and eventually also PE.

1.4 Scope of the Work

- The Gross Errors Detection (GED) will be omitted as part of the interface. However the main guidelines will be given since most methods of the interface are common to the PEDR module.

- It will be assume that the Parameter Estimation problem and Data reconciliation are totally independent.

    - This means that the model parameters used for reconciling the measured data are assumed to be certain.

    - The measured data used for parameter estimation are considered free of any type of errors either systematic gross or random.

- The specification of the Data reconciliation and Parameter Estimation interfaces were initially conceived for steady state. However, the general approach adopted makes them likely to be of application for the dynamic case.

- Process data and process model are implementation-dependent.

2. Requirements

2.1 Textual requirements

2.2 Use Cases

2.2.1 Actors

An actor is an entity outside the system that can exchange information either with a part or the whole functionality of the system. The actors identified during the development of the Parameter Estimation and Data Reconciliation (PEDR) interface are the following:

- Data Acquisition System or database: This is an external resource responsible for supplying the process variable measurements to the system and storing the result of PEDR procedure. Thus the flow of information is bi-directional. Its is important to note that measured variables could have a value at each time (on-line application) or a set of values (off-line application).

- PEDR Customer: The customer could be a user or application that requires from the system to perform a DR or a PE. The actor supplies and retrieves information to and from the system. It is the principal actor that initiates the communication and interacts with the system to use its functionality. The output from the system to the PEDR Customer is the adjusted values of redundant variables with their variance, the estimated values of the observed variables with their variance (error propagation) and the estimation of observed parameters.

- Solver: The solver is a system that returns the solution of the optimisation problem stated by the PEDR system.

    - **<u>For PE</u>**: The solver should be able to "take control" of a model (without accessing to the model's source code) and run it, as many times as required. This is done <u>to adjust the model parameters</u>, until, the discrepancies between selected model outputs and the measurements is reduced to a minimum, in the weighted least squares sense.

    - **<u>For DR</u>**: The solver should be able to "take control" of a model (without accessing to the model's source code) and run it as many times as required. This is done <u>to adjust the redundant process variables</u> until the inconsistency between selected model and the measurements is reduced to a minimum usually in the weighted least squares sense.

- Models: The model contains the set of equations that represents the system under consideration. Therefore the model can be constructed and specified based on the equation Set Object (ESO) interface defined in CO project. Inequality constraints are required for variable bounds. The construction of the objective function is part of the PEDR system; it means that the proper PEDR module will construct it, based on some criteria. In the case of DR the objective function is usually a quadratic function that depends mainly on the redundant variables and the variance-covariance matrix. The inverse of covariance matrix of measurement errors is determined within the PEDR system if not it is taken as identifying matrix.

2.2.2 List of Use Cases

2.2.3 Use Cases maps

2.2.4 Use Cases

PEDR input

The main information required by a PEDR system, in order to perform a Data Reconciliation or Parameter Estimation is:

- The process measurements
- The process model: ESO

UC-001: GET PROCESS MEASUREMENT

Actors: The PEDR resolution or the PEDR Customer.

Priority:

Classification:

Context:

Pre-conditions: No other use case could be performed before this one

Flow of events:

Inputs: Process Measured data.

Outputs: None.

The measured data are obtained from an external resource: the DCS or a database system. For an on-line application the data obtained directly from the DCS will be accessed by the PEDR through the database. These measurements are grouped into the experimental set.

Post-conditions:

Errors:

Uses:

Extends:

UC-002: GET PROCESS MODEL

Actors: The PEDR resolution.

Priority:

Classification:

Context:

Pre-conditions:

Flow of events:

Inputs: A set of algebraic equation that describes the process under consideration.

Outputs: None.

In order to follow the CO standardization the Equation Set Object is adopted.

Note: It is important to notice that during the creation of the ESO and the database the matching between the variables in both sides has to be undertaken. This mapping is not specified in this interface.

Post-conditions:

Errors:

Uses:

Extends:

PEDR statistical analysis of measured data

This statistical analysis focuses mainly into the Gross Error compensation or elimination.

UC-003: GROSS ERROR DETECTION

Actors: Gross Error Identification.

Priority:

Classification:

Context:

Pre-conditions:

Flow of events:

Inputs: Process Model and process measurements.

Outputs: check for the presence of Gross Error (GE)

Fault detection step involves comparing the observed behavior (process measurement) and a reference model (process model).

Post-conditions:

Errors:

Uses:

Extends:

UC-004: GROSS ERROR IDENTIFICATION

Actors: Gross Error Compensation.

Priority:

Classification:

Context:

Pre-conditions: Gross error is present

Flow of events: **Inputs:** Process Measurement, process model. **Outputs:** Type of the GE measurement-related error or process-related error. Isolation of the error by determining the index of the variable in error or the index of the process in error Post-conditions: Errors: Uses: Extends:

UC-005: GROSS ERROR COMPENSATION

Actors: The PEDR resolution Priority: Classification: Context: Pre-conditions: A Gross error is present and isolated Flow of events: **Inputs:** Measured variable and process model **Outputs:** Variable without gross error Compensation (if possible) of these errors. It means that if the variables in error are not corrected they are removed. The same for process leak. Post-conditions: Error with a know distribution Errors: Uses: Extends:

In the following figure the use case diagram corresponding to the statistical analysis of the process measurement is presented.

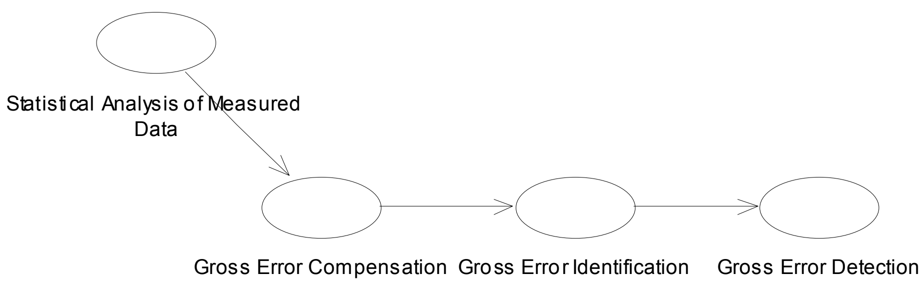

Figure 1 Use case diagram for statistical analysis of measured data

PEDR Resolution

In this section the reconciliation of redundant variables, the estimation of unmeasured observed variables and the estimation of process model parameters are specified in terms of use case.

UC-005: RECONCILE REDUNDANT VARIABLES

Actors: The PEDR Resolution and the PEDR customer. Priority: Classification: Context: Pre-conditions: Steady state mode, Error Normally distributed with zero mean and a positive degree of redundancy. Flow of events: **Inputs:** The redundant process model, process measurement and the objective function. **Outputs:** Reconciled value of measured variable and its variance. The existing conflicts between the redundant variables are eliminated and restrictions are satisfied. Post-conditions: Measurement without random error and consistent with process model. Errors: Uses: Extends:

UC-006: ESTIMATE UNMEASURED OBSERVABLE VARIABLES

Actors: The PEDR Resolution and the PEDR customer. Priority: Classification: Context: Pre-conditions: DR has been done. Flow of events: **Inputs:** Reconciled values of measured variable and observable model. **Outputs:** Estimated value of observed variable and its variance. To calculate an estimate of an unmeasured but observable variable. Post-conditions: Errors: Uses: Extends:

UC-007: ESTIMATE UNCERTAIN PARAMETER

Actors: The PEDR Resolution and the PEDR customer.

Priority:

Classification:

Context:

Pre-conditions: Steady state mode, The solvability condition is ensured and the process measurement is free of error.

Flow of events:

Inputs: A set of process variables measurements and observable model.

Outputs: Estimation value of uncertain parameter.

To calculate an estimate of an unknown quantity from a set of known ones.

Post-conditions:

Errors:

Uses:

Extends:

PEDR Results

UC-008: GET RESULTS

Actors: The PEDR customer.

Priority:

Classification:

Context:

Pre-conditions:

Flow of events:

Inputs: None.

Outputs: parameter estimated, reconciled variable value, observable variable values and statistical analysis (variance) results and variable classification results.

The system returns back the results of PEDR calculation.

Post-conditions:

Errors:

Uses:

Extends:

Global use case Diagram of the PEDR system

The top-level use cases view is represented by the main functionalities of the PEDR system and the interaction with the actors involved. The use cases within the PEDR are basically:

- The PEDR input information that allows the introduction of the process model and the process variables measured, reconciled or estimated.
- The statistical analysis data that allows the filtering of measured data from any Gross error
- The Variable classification
- The data reconciliation, the coaptation and parameter estimation
- The PEDR results

Each one of these use cases encompasses a set of bottom level use-cases as showed above and the whole use case diagram is showed in the next figure.

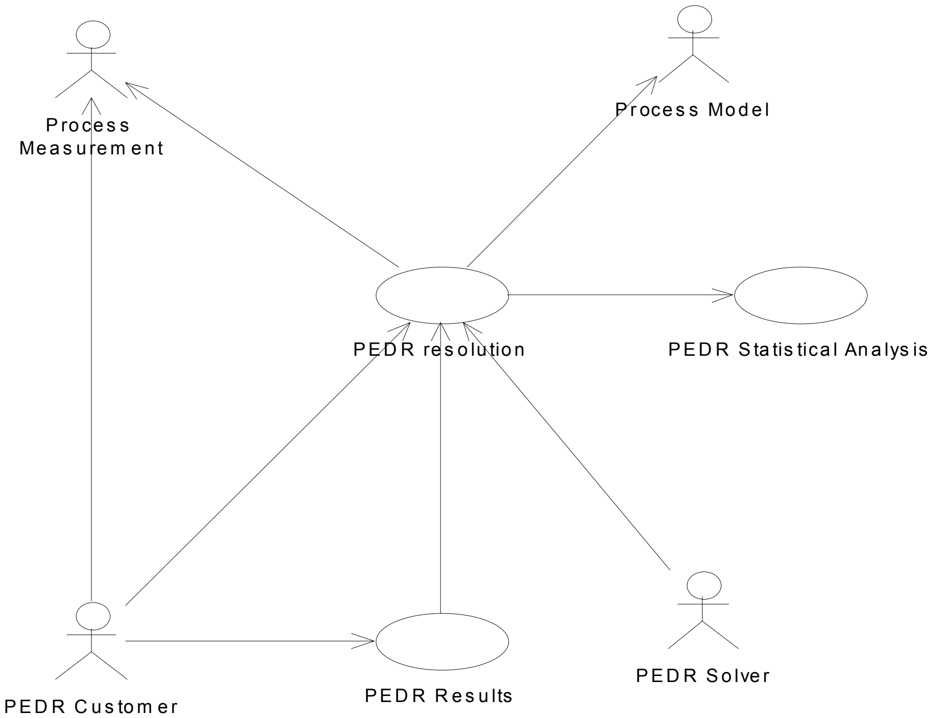

Figure 2 The general use case diagram of the PEDR system

2.3 Sequence diagrams

3. Anaysis and Design

3.1 Overview

3.2 Sequence diagrams

The sequence diagram represents the dynamics of the process. In this diagram the temporal sequence of steps to be performed in order to perform the DR or the PE is shown. The following diagrams encompass the dynamic case as well as the steady state case.

3.2.1 Sequence Diagram for Data Reconciliation

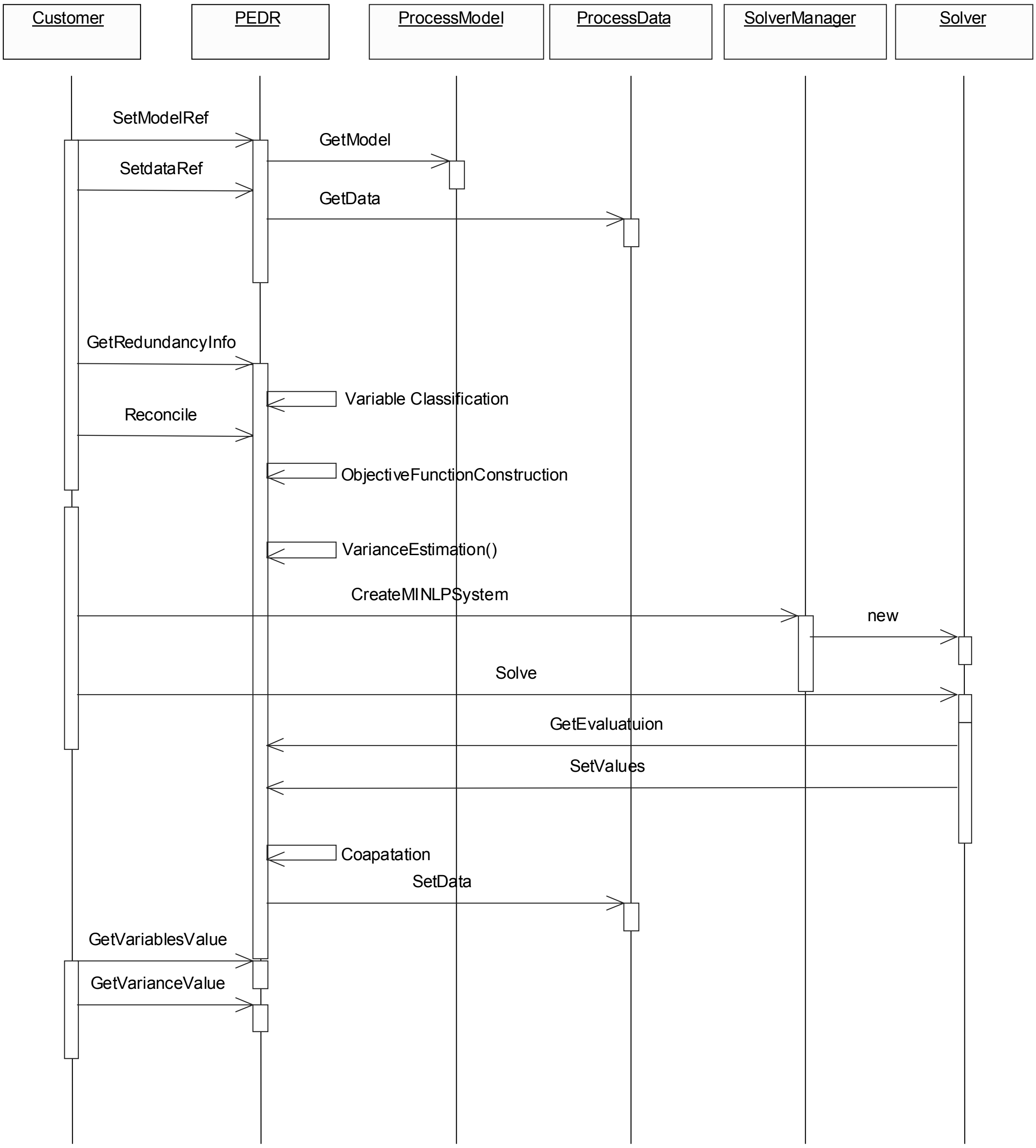

Figure 3 Scenario of object interaction of DR in a time-based Sequence either for steady state or dynamic case

3.2.2 Sequence Diagram for Parameter Estimation

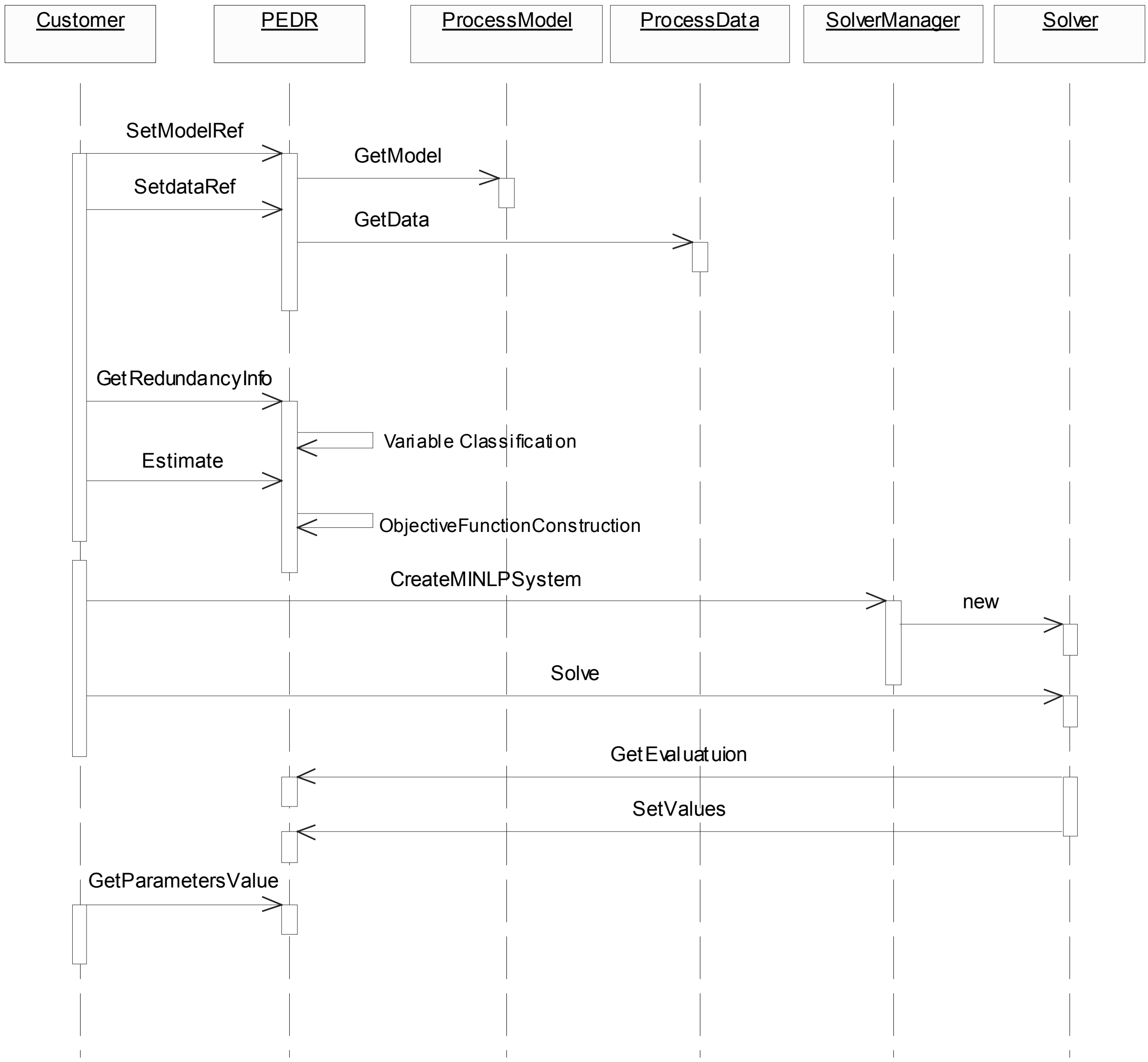

Figure 4 Scenario of object interaction of PE in time-based sequence either in steady state or dynamic case

3.3 Interface diagram

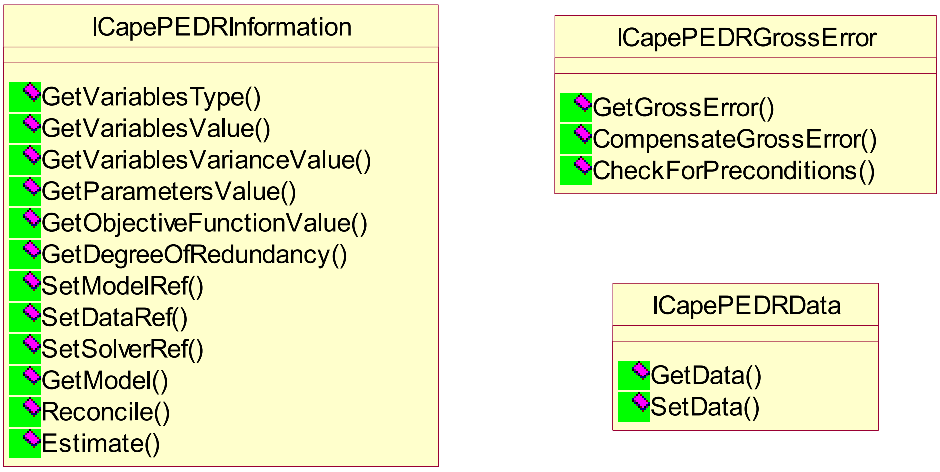

Figure 5 Interface diagram

3.4 State diagrams

3.5 Other diagrams

3.6 Interfaces descriptions

3.6.1 ICapePEDRData

Interface Name	ICapePEDRData
Method Name	GetData
Returns	--

Description

Get measured, reconciled or estimated data, this data could be a vector of measurements or a matrix. It is often necessary to group measured data into sets of experiments (corresponding to experiments on the plant in different operating conditions).

CapePEDRVariableType: CAPE_MEASURED, CAPE_RECONCILIED, CAPE_ESTIMATED

Arguments

Name	Type	Description
[in] experimentName	CapeString	The name of the experiment
[out] variablesType	CapePEDRVariableType	The type of variable (Measured, reconciled, estimated)
[out] variablesName	CapeArrayString	The list of variables specified by a name we want to get their values
[out] variablesValue	ICapeNumericMatrix	It could be a vector or a matrix

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRData
Method Name	SetData
Returns	--

Description

Set measured, reconciled or estimated data, this data could be a vector of measurements or a matrix.

CapePEDRVariableType: CAPE_MEASURED, CAPE_RECONCILIED, CAPE_ESTIMATED

Arguments

Name	Type	Description
[in] experimentName	CapeString	The name of the experiment
[in] variablesType	CapePEDRVariableType	The type of variable (Measured, reconciled, estimated)
[in] variablesName	CapeArrayString	The list of variables specified by a name we want to set their values
[in] variablesValue	ICapeNumericMatrix	It could be a vector or a matrix

Errors

ECapeUnknown, ECapeInvalidArgument, ECapeNoMemory, ECapeOutOfResources

3.6.2 ICapePEDRModel

This interface is specified within CO project introduced the model object that encompass the general mathematical description of a physical system. The fundamental building block employed for this purpose is a set of continuous equations, described by an Equation Set Object. The ESOs are sets of equations that describe mathematically the continuous part of a particular model or subparts of that model.

Finally, it is important to note that CAPE-OPEN does not define any standard mechanisms or interfaces for the construction of ESOs. These are left at the discretion of implementers.

3.6.3 ICapePEDRSolver

This interface is specified as Optimisation interface specification (MINLP specification).

3.6.4 ICapePEDRInformation

The interface that permits to obtain the results of PEDR calculation contain the following methods:

Interface Name	ICapePEDRInformation
Method Name	GetVariablesType
Returns	--

Description

This method returns the type of variable selected by its name. It could be redundant, no redundant, observable or non-observable. The measured and unmeasured types are omitted because this information is inherent in the redundant and observable types.

Arguments

Name	Type	Description
[in] variablesName	CapeArrayString	The names of the variables
[out] variablesType	CapeArrayPEDRVariableType	The Types of the specified variables are returned.

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	GetVariablesValue
Returns	--

Description

Get the values of a variable (measured, estimated or reconciled) specified by the name lower and upper dimension. Unmeasured but observed variables are estimated once the removal of random error is performed and the redundant measured variables are adjusted.

Arguments

Name	Type	Description
[in] variablesName	CapeArrayString	The names of the variables
[in] variableLBdim	CapeArrayLong	The lower bound dimension of the structured variable
[in] variableUBdim	CapeArrayLong	The upper bound dimension of the structured variable
[out] variablesValue	CapeArrayDouble	Get the values of the specified variables

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	GetParametersValue
Returns	--

Description

Get the values of uncertain or unknown parameters

Arguments

Name	Type	Description
[in] parametersName	CapeArrayString	The name of the parameters to be estimated
[out] parametersValue	CapeArrayDouble	Get the values of the specified parameters

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	GetVariablesVarianceValue
Returns	--

Description

Get the variances of reconciled variable. The estimation of the variance of the observable variables could be performed with error propagation.

Arguments

Name	Type	Description
[in] variablesName	CapeArrayString	The name of the structured variable
[in] variableLBdim	CapeArrayLong	The lower bound dimension of the structured variable
[in] variableUBdim	CapeArrayLong	The upper bound dimension of the structured variable
[out] variablesVariance	CapeArrayDouble	Get the variance of the specified variable. It is necessary in order to get a measure of goodness of fit

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	GetObjectiveFunctionValue
Returns	--

Description

Get the value of the objective function

Arguments

Name	Type	Description
[out] objectiveFunctionValue	CapeDouble	Get the value of the objective function

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	GetDegreeOfRedundancy
Returns	--

Description

Get if the degree is equal to zero, then the system is observable (solvable); if it is strictly less than zero the system is neither observable nor redundant; if the system is strictly greater than zero the system is observable and redundant)

Arguments

Name	Type	Description
[in] variableName	CapeString	The name of the variable
[out] degreeOfRedundancy	CapeLong	The degree of redundancy

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	SetModelRef
Returns	--

Description

Set the model reference that will be used within Data reconciliation procedure for example (set of equations that permits to represent the process (Linear non-linear or differential or a combination of them))

Arguments

Name	Type	Description
[in] modelName	CapeString	The name (identifier) of an Equation Set Objects

Error

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	SetDataRef
Returns	--

Description

Set an experiment set specified by name.

Arguments

Name	Type	Description
[in] experimentName	CapeString	The name of the experiment

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	SetSolverRef
Returns	--

Description

Set the solver reference for the solution of the PEDR problem.

Arguments

Name	Type	Description
[in] solverName	CapeString	The name (identifier) of a solver.

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRInformation
Method Name	Reconcile
Returns	--

Description

The client asks the PEDR Manager to reconcile the specified Data with the specified Model using the specified Solver.

Arguments

None

Errors

ECapeUnknown

Interface Name	ICapePEDRInformation
Method Name	Estimate
Returns	--

Description

Estimate

Arguments

None

Errors

ECapeUnknown

3.6.5 ICapePEDRGrossError

Interface Name	ICapePEDRGrossError
Method Name	GetGrossError
Returns	--

Description

Obtain the types of the gross errors present either process-related error, measurement-related error.

Arguments

Name	Type	Description
[in] measuredData	ICapeNumericMatrix	The values of the raw process measurement
[in] redundantModel	ICapeNumericESO	The set of process Constraints.
[out] grossError	CapeBoolean	If there exists a gross error
[out] isolationGrossError	CapeLong	The index of variable in error or process in error.
[out] grossErrorType	CapeString	The cause of the gross error detected is returned, process leak, sensor biased.

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRGrossError
Method Name	CompensateGrossError
Returns	--

Description

Method for eliminating the gross error by using statistical tests or any other similar strategy.

Arguments

Name	Type	Description
[in] measuredData	ICapeNumericMatrix	The set of measured variables
[in] redundantModel	ICapeNumericESO	The redundant model.
[out] correctedData	ICapeNumericMatrix	Measured variable without gross error.

Errors

ECapeUnknown, ECapeInvalidArgument

Interface Name	ICapePEDRGrossError
Method Name	CheckForPreconditions
Returns	--

Description

Check for the necessary and sufficient conditions to perform Steady state or dynamic PE, DR or PEDR

Arguments

Name	Type	Description
[in] measuredData	ICapeNumericMatrix	The set of measured variables
[in] processModel	ICapeNumericESO	The redundant model.
[out] operationMode	CapeBoolean	Operation mode.

Errors

ECapeUnknown, ECapeInvalidArgument

4. Interface specifications

4.1 COM IDL

```
// This specification is not currently available for COM platform.
```

4.2 CORBA IDL

```
// You can get these intructions in CAPE-OPENv1-0-0.idl within the
CAPEOPEN100::Business::Numeric::Pedr module
```

5. Notes on the interface specifications

6. Prototypes implementation

7. Specific glossary terms

8. Bibliography

9. Appendices