

CAPE-OPEN

Expanding Process Modelling Capability
through Software Interoperability Standards

Open Interface Specification: Petroleum Fractions Interface



www.colan.org

ARCHIVAL INFORMATION

Filename	Petroleum Fractions Interface Specification.docx
Authors	CO-LaN
Status	Public
Date	June 2023
Version	version 5.48
Number of pages	47
Versioning	Version 2, reviewed by Jean-Pierre Belaud, August 2003
	Version 5.48 approved for public release by Management Board on June 23, 2023
Additional material	
Web location	www.colan.org
Implementation specifications version	Version 1.1
Comments	

IMPORTANT NOTICES

COPYRIGHT NOTICE

Copyright 2007-2023 CAPE-OPEN Laboratories Network (CO-LaN).

PERMISSION NOTICE

Subject to all of the terms and conditions below, the owner of the copyright hereby grants you a fully-paid up, non-exclusive, non-transferable, perpetual, worldwide license (without the right to sublicense), to use this document to create and distribute software and to use, copy, and distribute this document provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this document; (2) the document will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this document. This limited permission automatically terminates without notice if you breach any of these terms or conditions.

THIS DOCUMENT IS PROVIDED "AS IS," AND CO-LAN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD-PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

CO-LAN WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of CO-LaN may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission obtained from CO-LaN. Title to copyright in this document will always remain with CO-LaN.

TRADEMARK USAGE

Many of the designations used by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in CO-LaN publications, and the authors are aware of a trademark claim, the designations have been printed in caps or initial caps.

SUMMARY

This document proposes one new interface, *ICapeThermoPetroleumFractions*, on the Material Object for handling complex mixtures containing [Petroleum Fractions](#). The proposed interface allows Unit Operations to read and modify properties specific to the petroleum and gas industry, such as cloud point, pour point, and Reid vapour pressure. The interface also allows run-time re-characterization of the properties of a petroleum fraction (e.g., Normal Boiling Point, Molecular Weight, Specific Gravity, Critical Temperature, etc...).

Therefore, the interface allows for extending plug-and-play of Unit Operation software components to the specific process operations present in the petroleum and gas industry (e.g., refining, upstream, etc...). Many of these specific Unit Operations are reactors (cat-crackers, hydrocracker, etc...) that change compound property values of petroleum fractions.

Only interactions between a PME and Unit Operations on petroleum fractions are considered in this interface specification. Furthermore, only thermodynamic services native to the PME are considered. Hence interaction regarding petroleum fractions between a PME and external Property Packages is not in the scope of this interface specification. Curve properties are excluded from the scope. Functionality of blend rules is not carried through the existing and newly proposed CAPE-OPEN interfaces.

The document contains requirements expressed in textual form, as well as Use Cases. From these requirements, the design of the interface is developed.

ACKNOWLEDGEMENTS

Many individuals and organizations have contributed to this document. The following are among the main contributors:

Jasper van BATEN
Krishna Murthy PENUKONDA
Michael HLAVINKA
Harry KOOIJMAN
Richard SZCZEPANSKI
Richard BAUR
Cesar PERNALETE

AmsterCHEM
AVEVA
Bryan Research & Engineering LLC
ChemSep
KBC Advanced Technologies
Shell Global Solutions International BV
University of Ghent

Michel PONS

CO-LaN

CONTENTS

CAPE-OPEN DOCUMENT ROADMAP	9
1. INTRODUCTION	10
2. SCOPE	11
2.1 BASIC PETROLEUM PROPERTIES	11
2.2 INSPECTION PROPERTIES	11
2.3 GRANULARITY OF PETROLEUM PROPERTIES	12
2.3.1 Bulk and compound Petroleum Properties	12
2.3.2 Recovery curves	13
2.3.3 Property curves, blending, granularity conversion	13
2.4 FLOWSHEET SIMULATION	14
3. REQUIREMENTS	15
3.1 TEXTUAL REQUIREMENTS	15
3.1.1 General requirements	15
3.1.2 Requirements pertaining to interaction between UNIT and PME	16
3.1.3 Requirements pertaining to interaction between a PME and a Property Package.....	17
3.1.4 Summary of requirements and new interfaces for Petroleum Fractions.....	17
3.2 USE CASES.....	18
3.2.1 Use Case priorities	18
3.2.2 Actors.....	18
3.2.3 List of Use Cases.....	18
3.2.4 Use Cases Maps.....	19
3.2.5 Use Cases.....	20
3.3 SEQUENCE DIAGRAM.....	26
4. ANALYSIS AND DESIGN	27
4.1 OVERVIEW	27
4.2 INTERFACE DIAGRAM.....	27
4.3 STATE DIAGRAM	27
4.4 OTHER DIAGRAMS.....	27
4.5 INTERFACE DESCRIPTION.....	28
4.5.1 ICapeThermoPetroleumFractions	29
4.6 SCENARIOS – PROPOSED SCENARIO TO VALIDATE THE INTERFACE	37
4.6.1 Introduction	37
4.6.2 Validation Scenarios.....	37
5. NOTES ON THE INTERFACE SPECIFICATIONS	37
5.1 REQUIREMENTS FOR REF-UNIT.....	37
5.1.1 Accepting connections of Material Objects to Ports.....	37
5.1.2 Setting values for Petroleum Properties	37
5.2 REQUIREMENTS FOR PME.....	37
5.2.1 Basis conversions.....	37
5.3 LIST OF PETROLEUM PROPERTIES	37
6. SPECIFIC GLOSSARY TERMS	43
6.1 ASSAY.....	43
6.2 BLEND	43
6.3 CHARACTERIZATION PROPERTIES	43
6.4 CHARACTERIZED PROPERTIES	43
6.5 COLDFLOW PROPERTIES	43
6.6 INSPECTION PROPERTIES	43
6.7 PETROLEUM	43
6.8 PETROLEUM FRACTIONS	43

6.9	PETROLEUM PROPERTIES	43
6.10	PSEUDO-COMPOUNDS.....	43
6.11	REFINERY PROPERTIES	44
6.12	REFINERY UNIT OPERATION	44
7.	BIBLIOGRAPHY	45
8.	APPENDIX 1: REQUIREMENTS PERTAINING TO INTERACTION BETWEEN A PME AND A PROPERTY PACKAGE	46

LIST OF FIGURES

FIGURE 1 USE CASES MAPS 19
FIGURE 2 MAIN SEQUENCE DIAGRAM 26
FIGURE 3 *ICAPETHERMOPETROLEUMFRACTIONS* INTERFACE DIAGRAM 27
FIGURE 4 MATERIAL OBJECT STATE DIAGRAM 27

CAPE-OPEN Document Roadmap

This document belongs to the documentation set of CAPE-OPEN interface specifications in its version 1.1: this documentation set contains Unit Operation interface specification 1.0. This imposes that, for Unit Operation components implementing the interfaces defined in this document, the CapeVersion registry key must be set at the value “1.0”.

This document replaces Petroleum Fractions interface specification which was part of CAPE-OPEN 1.0. Version 1.0 of Petroleum Fractions interface specification is deprecated.

This document is intended primarily for software engineers, who are interested in producing CAPE-OPEN compliant Unit Operation components (i) which are dealing with Petroleum Fractions and for PME developers.

All other readers need not go beyond **Section 2 Requirements**.

1. Introduction

For chemically well-defined fluids, such as petrochemicals, simulators conveniently draw on large data banks of chemical compounds for essential property data. For petroleum, a material of indefinite composition, such a direct means of representation is not practical. The lumping of a very large number of real compounds into a more manageable number of hypothetical compounds is a simplification suitable for many, but not all, process engineering purposes in refining. However, because these hypothetical compounds are not real compounds, some of their physical properties, that are constant for real compounds, are not constant. These hypothetical compounds i.e., Petroleum Fractions, have also additional properties, such as sulphur content, that can be modified by [Refinery Unit Operations \(REF-UNIT\)](#). These additional properties must be specified.

Although petroleum is predominately a hydrocarbon fluid, there are contaminants present in both fossil and synthetic oils, especially the former, e.g., organic sulphur, metals, inert gases, and water/brine. These have a significant impact on separation and reaction processes. For instance, the sulphur treating, and handling aspect of refining entails many reactors and separations, and small amounts of metals affect adversely the catalyst activity in the conversion processes.

A common scenario is that petroleum fractions are generated from petroleum [assays](#). Petroleum assays generally contain data in standard form such as ASTM D86 or True Boiling Point (TBP) analysis. The data is in the form of percent boiled off versus temperature. Given this data and other information such as bulk specific gravity (or curve), and bulk molecular weight (or curve), Petroleum Fractions can be generated. Each petroleum fraction will have assigned to it a molecular weight, specific gravity, and boiling temperature.

From these basic pieces of data, properties such as critical properties (critical temperature T_C , critical pressure P_C , critical volume V_C and critical compressibility Z_C), ideal gas heat capacity, vapor pressure, and liquid viscosity can be estimated from industry standard correlations such as the Lee-Kesler or Riazi-Daubert methods. From the property-model point of view, Petroleum Fractions are the same as real compounds, such as methane or water.

2. Scope

2.1 Basic Petroleum Properties

No two crude oil types are identical and there are crucial differences in crude oil quality. The results of crude oil assay testing provide extensive detailed hydrocarbon analysis data for refiners, oil traders and producers.

Petroleum information is often represented in the form of assays. Inspection assays generally represent the Petroleum Fractions in the form of results of a limited number of standardized tests, mostly for intra-company use. Such tests may include measured properties of the boiling range fractions produced from physical or simulated distillation methods. Information obtained from assays is used for detailed refinery engineering as well as client marketing purposes.

Rarely, but occasionally, a ‘full’ compositional analysis is undertaken; in this case mass spectrometry or gas chromatography could identify compounds or compound families. Such resolution can be necessary in particular to lubricants manufacture. The more usual procedure is to measure the boiling range distribution of crude oil as the cumulative amount distilled versus temperature, and then to slice the resulting S-shaped curve into several histogram-type intervals, each one representing a Petroleum Fraction. A Petroleum Fraction is a hypothetical hydrocarbon of certain normal boiling point and contains an amount of material determined by the slicing regime. The boiling temperature of a Petroleum Fraction is the volumetric average temperature for the histogram interval. A corresponding curve can be measured for density distribution. Typically, even the best laboratory instruments, chromatograph, or distillation apparatus, cannot measure boiling points much above 550 °C, yet a significant fraction of crude oil boils at higher temperatures and the final boiling point is uncertain. This introduces a complicating factor and the ‘end point’ is assumed to be 800+ °C.

The basic properties (normal boiling point, density, molecular weight) of light hydrocarbons are essentially those of the pure compounds. For heavier hydrocarbons, characterised by Petroleum Fractions, the boiling point and density can be obtained from measurements as described above and the molecular weight is predicted as a function of these two properties. Other correlation variants on the inter-relationships between those three properties exist as well. From these properties, compound critical properties and acentric factor can be estimated and thus the data needed by an EOS (Equation Of State) thermodynamic model is furnished. For the heavier end of the boiling range, there is additional measurement uncertainty since any attempt to measure high boilers in conventional apparatus would result in their thermal decomposition. For this reason, vapour pressure type thermodynamic models are sometimes used to predict K-values for refinery separations involving such material, as opposed to customary EOS methods. Vapour pressure can be correlated from basic physical properties and is supported by measurement. In high pressure oil production applications, the uncertainty in the compound critical properties is taken as justification for using these critical properties as tuning parameters to reproduce field Gas/Oil ratios (i.e., VLE or VLLE).

As assay data are not uniformly structured, formulating an interface that deals with assay data in a useful manner is problematic. Therefore, assay data are not part of the scope of this interface specification. Instead, assay data may be represented by both compound properties and bulk properties.

The purpose of the specification is solely to enable flowsheet simulations to deal with Petroleum Properties via Petroleum Fractions interface. The generation of Petroleum Fractions out of assay data remains a task of the simulation environment and is not covered by this interface specification.

2.2 Inspection Properties

In addition to the basic physical properties of density, molecular weight, and so on, which are essential to phase equilibrium calculations, a wide range of other Petroleum Properties are key quality and control indicators used in practice. Inspection properties include sulphur content, flash point, Reid vapour pressure, refractive index, nickel content, vanadium content, Conradson carbon, toxicity parameters and coldflow properties (cloud point, freeze point, ...). There are standard procedures for their measurement, and they are reported in a typical oil assay. If enough data points are available to form a curve (covering the boiling range) for a given property,

such as sulphur content, values can be assigned, by interpolation, to each Petroleum Fraction in a manner analogous to the boiling curves already described.

These inspection properties are tracked throughout processes for a variety of reasons. For example, cloud point may be a control point for distillation product streams, content of metals which are catalyst poisons is required by reactor models and other inspection properties such as sulphur and cetane number must be known for product quality control.

2.3 Granularity of Petroleum Properties

Petroleum Properties may be specified in different levels of details, levels referred to through *granularity*:

- Bulk property - A bulk value is a single value for the entire mixture.
- Curve property - Arbitrary granularity can be defined in terms of a curve of the property versus boiling temperatures or cuts.
- Compound property - A compound-based value is a single value for each petroleum fraction.

The fineness of the granularity is related to the number of points defining the property. A bulk property has always the coarser granularity because it is a single valued property. Usually, a curve property has a coarser granularity than a compound property since a compound property is interpolated from points on a curve property.

2.3.1 Bulk and compound Petroleum Properties

Table 1 lists the Petroleum Properties that can appear as bulk Petroleum Properties and/or as compound Petroleum Properties. These representations of Petroleum Properties can be inter-converted to an extent. The compound representation of any Petroleum Property in Table 1 can be uniquely converted into a bulk value of the property given the mole, mass or volume fractions and an appropriate blend rule. Conversion of the bulk value of a Petroleum Property to compound values of the same Petroleum Property is not unique. Additional assumptions or rules are required on how to distribute the bulk Petroleum Property over compound Petroleum Properties.

The PME delivers both bulk and compound representations of any Petroleum Property stored at a feed (or any complete Material Object). All Petroleum Properties supported by the Material Object should have a value. A REF-UNIT may request bulk or compound representations of any Petroleum Property at the feed and may specify either a bulk or a compound representation of any Petroleum Property at the outlet (or any temporary Material Object).

The PME may store the Petroleum Property in any granularity the PME chooses provided the PME must be able to return the Petroleum Property in the granularity it was specified.

- Example 1: if the REF-UNIT specifies a bulk value of a Petroleum Property, requesting the bulk value of the same Petroleum Property should always return the specified value. Internally, the PME may store the bulk value on the Material Object or may convert the bulk value to compound values. In the latter case the conversion back to bulk value must be consistent with the conversion from bulk value to compound values. As the conversion from bulk value to compound values is not unique, the PME must decide how to distribute the Petroleum Properties of the compounds.
- Example 2: if the REF-UNIT specifies the compound values of a Petroleum Property, the Material Object must be able to return the same compound values. This implies that the compound values themselves must be stored on the Material Object.

The PME provides the inlet Material Objects with as much information as it can (e.g., complete compound slate, bulk Petroleum Properties, etc...). The REF-UNIT provides as much information as it can to its outlet Material Objects and will request from the PME to complete the properties of all Petroleum Fractions using *ICapeThermoPetroleumFractions::UpdatePetroProperties*. In addition, the REF-UNIT specifies Petroleum Properties in the finest granularity it can provide, without making contradictory specifications. For example,

if a REF-UNIT sets Sulphur values, it should do so on a compound basis if it can, and it should not specify both compound and bulk values of that Petroleum Property.

2.3.2 Recovery curves

In principle, Petroleum Properties can be alternatively represented as a curve versus boiling temperatures or versus recovery (mass or volume based) in simulated or real distillation. In practice, the curve properties which are most widely used are the recovery curves.

Recovery curves (ASTM D-86, ASTM D-1160, ASTM D-2887, TBP curve, etc...), also known as distillation, boiling- or yield-curves, relate the total yield versus boiling temperatures. They can be constructed from the boiling ranges and the yields of each compound given the proper correlation for each type of recovery curve. The need may arise for REF-UNITs to determine the recovery curves in a manner consistent with the PME. At that point inclusion of recovery curve functionality may have to be considered.

This version of the interface specification provides no support for obtaining recovery curves directly from the Material Object.

2.3.3 Property curves, blending, granularity conversion

In addition to a representation of a Petroleum Property per compound, the distribution of a Petroleum Property over the boiling range may be also represented by a curve. A Petroleum Property like sulphur for example can be converted to a curve either versus the boiling temperatures or versus the recovery (recovery and boiling point temperatures relate to each other in accordance with the recovery curve). This construction can be done given the Petroleum Fractions, their boiling temperature, yield and sulphur content. Real compound contributions may also be included. Some real compounds may contribute to the curve (such as thiophene in the case of sulphur).

Sulphur content (*SContent*) is a Petroleum Property that blends linearly by mass. Other Petroleum Properties may blend linearly by mole or by volume. Once the sulphur content property curve is constructed, sulphur content can be interpolated from that curve at any desired granularity (this is subsequently referred as granularity conversion). The total of the property curve indicates the bulk value of the sulphur content property, and therefore may be used for bulk blending (determining the bulk value from a curve or a compound property). Two Petroleum Property curves can be blended together (considering their respective flowrates) which makes possible to blend two streams together.

Not all properties blend linearly by mass, mole or volume. Often a transformation of the Petroleum Property is required. For example, viscosity is commonly transformed to a viscosity index. This transformation is often termed a blend rule. The viscosity index in turn is used for constructing a curve. Then granularity conversion, bulk, or stream blending leads to a new value for viscosity index. Viscosity can be obtained from this viscosity index using the inverse transformation of the blend rule.

Blend rules are defined at simulator level as part as the configuration of the simulation, much like a thermodynamic configuration. It would be desirable to transfer the information on blend rules, that is the blending basis, the transformation, and the inverse transformation, from the PME to the REF-UNIT. This would allow the REF-UNIT to internally perform granularity conversion, bulk blending, and stream blending in a manner fully consistent with the PME.

As the transformations involved with the blend rules may depend on fluid properties (both thermodynamic properties and Petroleum Properties) that are not part of the blend rule itself, providing a generic interface to convey blend rules is complicated, and postponed to a later version of this interface specification. This extension to the Petroleum Fractions interface specification would be necessary if REF-UNITs or 3rd party Property Packages geared toward Petroleum Properties, were developed that required access to blending functionality. If such an interface is designed, the correlations required to construct recovery curves in a manner consistent with the PME may also be made available.

2.4 Flowsheet Simulation

Considering real compounds to represent oil is impractical since the nature, identity and consequently properties of these compounds are mostly unknown. It would also require at least hundreds if not thousands of different compounds. Memory and performance issues would arise. So, the pseudo-compound approach makes sense from the perspective of flowsheet simulation.

Most important refinery units are distillation columns and reactors. Distillation is adequately modelled using the pseudo-compound approach for the characterisation of oil blends or oil fractions. For reaction processes, this approach has its limitations. Formal reactions cannot be expressed, elemental balance may be compromised, and heat of reaction is not readily predictable since heat of formation is uncertain. A way to circumvent these limitations involves defining Petroleum Fractions with different properties between inlet and outlet.

Possibilities, of varying acceptability, for defining the product streams for any particular reactor model include:

1. *Inlet is described by a given set of Petroleum Fractions and their properties; outlet is described by the same set of Petroleum Fractions with the same property values as in the inlet.* The amount of material in each Petroleum Fraction in the effluent is determined by the reactor algorithm. Since these processes are known to change both the basic Physical Properties and the Petroleum Properties of the material, the effluent properties assigned to the outlet stream will be inaccurate because represented through the same properties of Petroleum Fractions as the inlet. For instance, across a vis-breaker, or even a milder reactor such as a hydrotreater, there is only a slight shift in boiling range from inlet to outlet but the accompanying shift in density curve is significant.
2. *Inlet is described by a given set of Petroleum Fractions; outlet is described by a different set of Petroleum Fractions.* A severe limitation of most conventional commercial simulators is that a second set of Petroleum Fractions, which will carry the effluent, must be defined a priori (before the simulation starts) and, being of fixed properties, will not change with reactor operating conditions. The simulator carries both sets of Petroleum Fractions throughout the flowsheet. The reactor algorithm 'fills' the second set of Petroleum Fractions perhaps by applying fixed yields (derived from plant measurement). The combined compound list is obviously much longer than for the previous scenario, even though there may be almost complete overlap between the two sets of Petroleum Fractions because of a small shift in boiling range between inlet and outlet. This is computationally inefficient as it affects the solution time for all other unit operations in the flowsheet.
3. *Inlet is described by a given set of Petroleum Fractions; outlet is determined by the same Petroleum Fractions and internally the reactor model 'maps' Petroleum Fractions to real chemical compounds.* In this instance the reactor model applies a formal chemistry to the reaction. The real chemical compounds could be purely internal to the reactor model. The effluent stream can easily be constituted back into Petroleum Fractions for feeding downstream processes.
4. *Inlet is described by a given set of Petroleum Fractions; outlet is determined by the 'dynamically' revised Petroleum Fractions.* This represents an ideal whereby all Properties (Physical and Petroleum) of each Petroleum Fraction, other than boiling point, are updated by the reactor algorithm. Such a solution introduces the concept of stream-dependent, modifiable (compound-) properties.

This document defines the framework that allows for modifiable Petroleum Properties, as described in the fourth possibility outlined above. Reactors can specify values for bulk or compound Petroleum Properties of the product streams.

3. Requirements

3.1 Textual requirements

3.1.1 General requirements

REQ-01: Petroleum Fractions must be identifiable from any other type of Compound.

Rationale: Petroleum Fractions and real Compounds behave differently in terms of setting and getting properties. This requirement is fulfilled by defining an additional Compound Constant property (of Boolean type) that specifies the type of compound: *isPetroleumFraction*. If *isPetroleumFraction* property is not available for a Compound, the caller identifies the Compound as real.

REQ-02: Petroleum Properties assigned to a Petroleum Fraction can change values.

Rationale: Petroleum Properties assigned to a Petroleum Fraction can change during the flowsheet calculations, when, for example, Petroleum Fractions undergo chemical reactions in a reactor: the sulphur content of a Petroleum Fraction may be changed in a desulfurization unit. In fact, a reactor model can also alter the molecular weight or specific gravity of a given Petroleum Fraction. As a result, the Petroleum Fraction will also need to be re-characterised (Note that once the Petroleum Fraction has been re-characterised, the methods used to estimate the property and parameters of the Petroleum Fraction will not be changed).

When the characterized properties and Petroleum Properties of a Petroleum Fraction are changed, the Petroleum Fraction essentially becomes a new species (Compound). In other words, the reactor generates a new set of Petroleum Fractions, and the characterized properties of these new Petroleum Fractions are determined. If the reactor is within a recycle loop, properties of these Petroleum Fractions are typically updated at each iteration.

REQ-03: Compound Constants assigned to real Compounds must not change.

Rationale: real Compounds remain unaltered as part of any re-characterization. Their Compound Constants therefore do not change as part of the simulation calculations. Their Compound Constants can be changed only through configuration of the software component (PME or Property Package) that provides the Compound definitions.

REQ-04: Petroleum Properties of Petroleum Fractions must not be returned as Compound Constants.

Rationale: Petroleum Properties are variables and therefore are not constant. Any software component that obtain Compound Constants assumes that returned values are fixed. Unit Operations depend for their calculations on values returned by *ICapeThermoCompound::GetCompoundConstant* to remain fixed in between calls to *ICapeUnit::Calculate*.

For example, *CriticalTemperature* is a Petroleum Property for a Petroleum Fraction (see Table 1) and a Compound Constant for a real Compound.

See the Thermodynamic specifications 1.1 (ii) on how to deal with properties which are not available.

REQ-05: values of Petroleum Properties must be available for real Compounds (when applicable).

Rationale: many Petroleum Properties have an equivalent Compound Constant (e.g., critical temperature, critical pressure). For some other Petroleum Properties, a constant value can be derived (e.g. sulphur content can be derived from the Chemical Formula). Granularity conversion of a Petroleum Property may require property values for all Compounds, real or hypothetical. It may not be always possible to assign a value for a Petroleum Property of a real Compound (for example cloud point for methane). In such a case UNDEFINED may be returned.

REQ-06: Petroleum Properties cannot be specified for real Compounds.

Rationale: Petroleum Properties for real Compounds are not variables. For example, the critical temperature of a Petroleum Fraction is variable (typically a characterized property) but is fixed for a real Compound. To obtain an overall sulphur content of a mixture, the sulphur content of a real Compound may be needed but cannot be set.

3.1.2 Requirements pertaining to interaction between UNIT and PME

REQ-07: characterized properties for Petroleum Fractions must be updated because of a change of Petroleum Properties.

Rationale: Material Objects contain the compound (species) information. Material Objects track Petroleum Properties as the Petroleum Fractions go through the Flowsheet from one Unit Operation to another. These Petroleum Properties may be modified by a REF-UNIT; a mechanism must therefore be provided that allows for re-characterisation of the Petroleum Fractions once the Unit Operation has changed Petroleum Properties that affect the characterisation.

During its calculation, a REF-UNIT modifies Physical Properties and Petroleum Properties on the Material Object. As a result of the change in the Petroleum Properties, the parameters underlying the thermodynamic property calculations (the characterized properties) may require updating.

The PME is responsible for updating the characterized properties i.e., the re-characterisation). A workflow is required in which the PME performs the re-characterisation, when required. The established methodology is as follows: all the Material Objects connected to inlet Ports of a REF-UNIT are populated by the PME using a single collection of Petroleum Fractions. The PME sets the values of Petroleum Properties carried by each Petroleum Fraction to be equal for each of the Material Objects connected to the inlet Ports of the REF-UNIT. This set of Petroleum Property values remains constant throughout the UNIT calculation since the REF-UNIT is not allowed to modify the contents of the Material Objects connected to inlet Ports. The PME uses the same collection of Petroleum Fractions to populate the Material Objects connected to outlet Ports and assigns to them the same set of Petroleum Property values as for the inlet Material Objects.

If the REF-UNIT requests the creation of a Material Object during *ICapeUnit::Calculate*, using *ICapeThermoMaterial::CreateMaterial*, the PME populates the created Material Object also with the same collection of Petroleum Fractions and assigns to them the same set of Petroleum Property values. Each Material Object not connected to an inlet Port keeps track of a list of modified properties. This list is initially empty. The REF-UNIT is free to modify values of Petroleum Properties and thermodynamic properties as required on any Material Object not connected to an inlet Port.

Once the REF-UNIT is done modifying Petroleum Properties, before the REF-UNIT requires the Material Object to perform property or phase equilibrium calculations, and before the REF-UNIT uses the Material Object as the input argument to *CopyPetroProperties*, the REF-UNIT calls *UpdatePetroProperties* on the Material Object.

At any time prior to calling *UpdatePetroProperties* on a given Material Object, the REF-UNIT must specify the overall composition on the Material Object. During the *UpdatePetroProperties* call, the Material Object decides which Petroleum Properties that have not been modified by the REF-UNIT, require updating (for example adapt values of compound Sulphur such that a value of bulk Sulphur specified by the REF-UNIT is realized), and whether a re-characterisation is required.

The list of modified properties is reset to empty, such that subsequent property modifications by the REF-UNIT will be captured for the next call to *UpdatePetroProperties*. Property calculations, phase equilibrium calculation or *CopyPetroProperties* should not be done on a modified Material Object for which *UpdatePetroProperties* has yet not been called. For UNITS which are not REF-UNIT, the list of modified Petroleum Properties is always empty and *UpdatePetroProperties* need not be called. If the UNIT does not set any Petroleum Property on a Material Object, the UNIT does not need to call *UpdatePetroProperties* on that Material Object.

3.1.3 Requirements pertaining to interaction between a PME and a Property Package

Currently out of scope. See Appendix 1 for the requirements pertaining to the interaction between a PME and a Property Package.

3.1.4 Summary of requirements and new interfaces for Petroleum Fractions

1. **New properties, known as** Petroleum Properties such as Reid vapor pressure, cloud point. Define the list of Petroleum Properties. Completed: *new properties are included.*
2. **Material Object** extension to include Compound type to allow handling of Petroleum Fractions. Define the list of Compound types: real neutral compound, ions, assay, petroleum fraction, polymer, etc. Completed: *The THRM functionality GetCompoundConstant defines a new compound constant "isPetroleumFraction" value of which is TRUE for a Petroleum Fraction, FALSE otherwise. If a value is absent, the Compound is not a Petroleum Fraction.*
3. **Material Object** extension to include Petroleum Properties. Completed: *These will all be included in the Material Object by implementing the ICapeThermoPetroleumFractions interface.*
4. **Workflow to allow for re-characterisation of Petroleum Fractions.** Completed: *ICapeThermoPetroleumFractions::UpdatePetroProperties to be called as part of this workflow.*
5. **New Interface to update Petroleum Properties.**

Methods for setting and getting Petroleum Properties, for enumerating Petroleum Properties, for copying Petroleum Properties, for updating Petroleum Properties. Completed: *functionality available through methods of ICapeThermoPetroleumFractions interface.*

6. **Methodology for interaction between a PME and a Property Package:** define an interface for notification of property changes and request for update of dependent properties and re-characterisation on the Property Package and define an interface that allows the Property Packages to store characterized properties on the family of Material Objects that share the same compound Petroleum Properties. *Not supported by this version of the specification:* the scenario of interaction between a PME and a Property Package clearly is more complex than that between a PME and a REF-UNIT. The functionality required for the interaction between a PME and a Property Package is therefore not currently covered by this specification document (see Appendix 1 for description of what this interaction requires) and will be addressed in a future version of this specification. Requirements listed for that interaction may have to be revised when the corresponding interface design will be defined.

3.2 Use Cases

As a Unit Operation aware of Petroleum Fractions is an extension of a regular CAPE-OPEN Unit Operation, all Use Cases described in the Unit Operation Interface Specification (i) also apply to the special case of Unit Operations aware of Petroleum Fractions. Consequently, these Use Cases are not repeated here.

3.2.1 Use Case priorities

- ❑ **High.** Essential functionality for a Flowsheet Unit. Functionality without which the operation usability or performance of a Flowsheet Unit might be seriously compromised.
- ❑ **Medium.** Very desirable functionality that will make Flowsheet Units more usable, transportable, or versatile. The essence of a Flowsheet Unit is not compromised by this Use Case, although the usability and acceptance of the component can be.
- ❑ **Low.** Desirable functionality that will improve the performance of Flowsheet Units. If this Use Case is not met, usability or acceptance can decrease.

3.2.2 Actors

- ❑ **Flowsheet Builder.** The person who sets up the flowsheet, the structure of the Flowsheet, chooses thermodynamic models and the Unit Operation models that are in the Flowsheet.
- ❑ **PME.** The part of a simulator whose job is to create a Flowsheet or load a previously stored Flowsheet, solve the Flowsheet, and display the results.
- ❑ **REF-UNIT.** A CAPE-OPEN Unit Operation requiring Petroleum Fractions re-characterisation procedures. These are typically reactor operations and blenders/mixers.

3.2.3 List of Use Cases

- ❑ UC-PET-001 Add REF-UNIT to Flowsheet
- ❑ UC-PET-002 Specify Material Connections of REF-UNIT
- ❑ UC-PET-003 Get Petroleum Properties from Inlet Material Connections
- ❑ UC-PET-004 Set Petroleum Properties in Outlet Material Connections.
- ❑ UC-PET-005 Perform calculation on temporary Material Object

3.2.4 Use Cases Maps

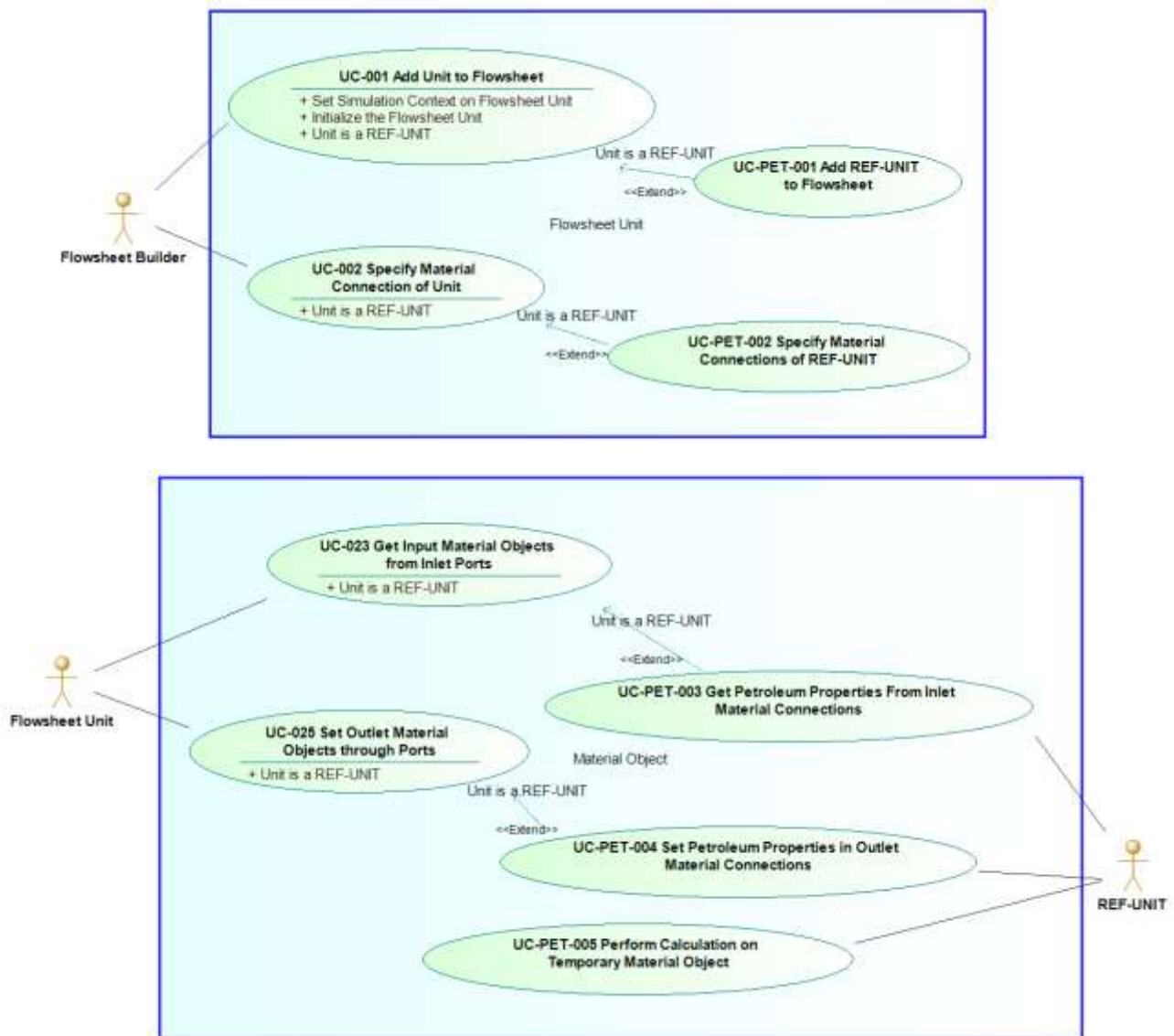


Figure 1 Use Cases maps

3.2.5 Use Cases

UC-PET-001 ADD REF-UNIT TO FLOWSHEET

Actor: <Flowsheet Builder>

Priority: <High> (as in <UC-001: Add Unit to Flowsheet>)

Pre-conditions: as in <UC-001: Add Unit to Flowsheet>

Flow of events:

As in UNIT Use Case Add UNIT to Flowsheet, but the UNIT is a REF-UNIT.

No additional check is performed at this stage regarding support for Petroleum Fractions by the PME. Such checks are performed during UC-PET-002 SPECIFY REF-UNITS MATERIAL CONNECTIONS.

Post-conditions:

As in UNIT Use Case <UC-001: Add Unit to Flowsheet>.

Errors:

As in UNIT Use Case <UC-001: Add Unit to Flowsheet>.

Uses: None

Extends: UNIT Use Case <UC-001: Add Unit to Flowsheet>

UC-PET-002 SPECIFY MATERIAL CONNECTIONS OF REF-UNIT

Actor: <Flowsheet Builder>

Priority: As in UNIT Use Case Specify UNIT's Material Connections

Pre-conditions: As in UNIT Use Case Specify UNIT's Material Connections

Flow of events:

As in UNIT Use Case Specify UNIT's Material Connections, but the UNIT is a REF-UNIT.

And:

If its Port requires support for Petroleum Fractions, REF-UNIT checks, during *ICapeUnitPort::Connect*, that the Material Object supports Petroleum Fractions. Such support can be assumed if the Material Object exposes the *ICapeThermoPetroleumFractions* interface.

If connection fails, the Port raises an appropriate error with message e.g. "Object trying to connect does not support Petroleum Fractions."

Post-conditions:

As in UNIT Use Case Specify UNIT's Material Connections.

Errors:

As in UNIT Use Case Specify UNIT's Material Connections

<The connected Material Object does not support Petroleum Fractions and such support is required by the REF-UNIT.>

Uses:

Extends: UNIT Use Case <Specify UNIT's Material Connections>

UC- PET-003 GET PETROLEUM PROPERTIES FROM INLET MATERIAL CONNECTIONS

Actor: <REF-UNIT>

Priority: <High>

Pre-conditions: As in the UNIT Use Case <UC-023 Get Input Material Objects From Inlet Ports>; <The Material Object exposes *ICapeThermoPetroleumFractions*.>

Flow of events:

As in the UNIT Use Case <Get Input Material Streams From Input Ports>, but the information to gather now includes information concerning Petroleum Fractions.

The REF-UNIT requests certain Petroleum Properties using the *ICapeThermoPetroleumFractions* interface.

The REF-UNIT can obtain the type of a Compound via *GetCompoundConstant* implemented on the Material Object, by querying the value of the Compound Constant "*PetroleumFraction*". If this Compound Constant value is not provided by the Material Object, the REF-UNIT should assume that all Compounds are real.

Post-conditions:

<The REF-UNIT is delivered with the requested property values.>

Errors:

< A Petroleum Property does not exist on the Material Object.>

Uses: None

Extends: UNIT Use Case <UC-023 Get Input Material Objects From Inlet Ports>

UC- PET-004 SET PETROLEUM PROPERTIES IN OUTLET MATERIAL CONNECTIONS

Actor: <REF-UNIT>

Priority: <High>

Pre-conditions:

As in the UNIT UC-025 <Set Outlet Material Objects Through Ports>. <The Material Object exposes *ICapeThermoPetroleumFractions*.>; <All Material Objects connected to inlet and outlet Ports have been populated with the same set of Petroleum Properties prior to invoking *ICapeUnit::Calculate*.>

Flow of events:

As in the UNIT Use Case UC-025 <Set Outlet Material Objects Through Ports>, but the information to set includes information concerning Petroleum Fractions.

Petroleum Properties have been set by the PME on each outlet Material Object prior to invoking *ICapeUnit::Calculate*.

The REF-UNIT can obtain the type of a Compound via *GetCompoundConstant* implemented on the Material Object, by querying the value of the compound constant "*CompoundType*". If this compound constant value is not provided by the Material Object, the REF-UNIT should assume that all Compounds are real. Compound-based petroleum properties may only be set for petroleum fractions, not for real Compounds; for real Compound the values must be UNDEFINED. Performance note: since these values are constant, these actions can be performed instead within *ICapeUnit::Validate*.

The REF-UNIT is free to modify Petroleum Properties and regular CAPE-OPEN properties (flow, *totalflow*, overall fraction and sufficient conditions to calculate the phase equilibrium on the outlet Material Object as part of this flow of events) on the outlet Material Object. The PME keeps track of which properties have been modified by the REF-UNIT since the start of the calculation or the calling of *CopyPetroProperties()*, or the last call to *UpdatePetroProperties* (see Note below).

If the REF-UNIT has modified the value of at least one Petroleum Property, the REF-UNIT calls *UpdatePetroProperties* on the outlet Material Object, to signal the PME that the Material Object should be re-characterised; the PME decides how to best set values of Petroleum Properties of the outlet Material Object that have not directly been set by the REF-UNIT, but are affected by properties that have been directly set by the REF-UNIT. For example, in case the REF-UNIT has specified a bulk Sulphur value, the PME should change compound-based Sulphur values such that the specified bulk Sulphur values are realized. This implies that the REF-UNIT should set overall fraction or overall compound flow rates before calling *UpdatePetroProperties*. In addition, the PME will re-determine characterized properties of the outlet Material Object if the REF-UNIT has changed characterization properties (typically APIGravity, NormalBoilingPoint/InitialBoilingPoint/FinalBoilingPoint and MolecularWeight).

As in the UNIT Use Case <Set Outlet Material Objects Through Ports>, the REF-UNIT will set the flow rate and invoke a phase equilibrium calculation on the outlet Material Objects as part of the calculation sequence; *UpdatePetroProperties* must have been called before this equilibrium calculation.

Note: the REF-UNIT may require property calculations or phase equilibrium calculations to solve its internal equations, e.g. enthalpy calculations at specified conditions to close an enthalpy balance. Such calculations may not be performed on an inlet Material Object (due to the restriction that inlet Material Objects may not be modified by any UNIT) but may be performed on duplicates of inlet Material Objects or directly on outlet Material Objects. If such calculations are to be done on a re-characterised Material Object, *UpdatePetroProperties* must be called on such a Material Object prior to property or phase equilibrium calculations. Therefore, *UpdatePetroProperties* may be called multiple times on a Material Object during the calculation of a REF-UNIT. *UpdatePetroProperties* will therefore consider all (petroleum and regular) property changes since the start of the calculation, the last call to *CopyPetroProperties* or the last call to *UpdatePetroProperties*.

Post-conditions:

<For each of the outlet Material Objects, the specification is complete and fully characterized.>

Errors:

<Specified Petroleum Property values are outside of valid range.>

<*UpdatePetroProperties* is not called before property calculation or equilibrium calculations, while Petroleum Properties were modified.>

<Re-characterisation fails (i.e. the available methods for characterising the pseudo compounds cannot process the newly specified Petroleum Properties).>

<Thermodynamic property or phase equilibrium calculation fails.>

Uses:

Extends: UNIT Use Case <UC-025 Set Output Material Objects Through Ports>

UC- PET-005 PERFORM CALCULATION ON A TEMPORARY MATERIAL OBJECT

Actor: <REF-UNIT>

Priority: <High>

Pre-conditions:

<The Material Object exposes *ICapeThermoPetroleumFractions*.>

Flow of events:

Within its calculation (or during *ICapeUtilities::Edit*), the REF-UNIT may require property calculations or phase equilibrium calculations to solve its internal equations, e.g. enthalpy calculations at specified conditions to close an enthalpy balance.

Such calculations may not be performed on inlet Material Objects (due to the restriction that inlet Material Objects may not be modified by any UNIT) but may be performed on duplicates of inlet Material Objects or directly on outlet Material Objects.

Upon creation through call to *ICapeThermoMaterial::CreateMaterial* by the REF-UNIT, a temporary Material Object is populated by the PME with the same set of Petroleum Properties as all Material Objects connected to inlet Ports of the REF-UNIT. The REF-UNIT may modify Petroleum Properties by copying Petroleum Properties to the newly created Material Object using *CopyPetroProperties* or by setting Petroleum Properties on the Material Object.

If at least one Petroleum Property has been modified on the Material Object, *UpdatePetroProperties* must be called prior to any thermodynamic property or phase equilibrium calculation.

Therefore, *UpdatePetroProperties* may be called multiple times on a Material Object during the calculation of a REF-UNIT. *UpdatePetroProperties* must therefore consider all (petroleum and regular) property changes since the creation of the temporary Material Object, the last call to *CopyPetroProperties* or the last call to *UpdatePetroProperties*.

Post-conditions:

<Thermodynamic and physical property calculations have successfully been performed on the temporary Material Object.>

Errors:

<Specified Petroleum Property values are outside of valid range.>

<*UpdatePetroProperties* is not called before thermodynamic property calculation or equilibrium calculations, while Petroleum Properties were modified.>

<Re-characterisation fails (i.e., the available methods for characterising the pseudo compounds cannot process the newly specified Petroleum Properties).>

<Thermodynamic property or phase equilibrium calculation fails.>

Uses:

None

3.3 Sequence diagram

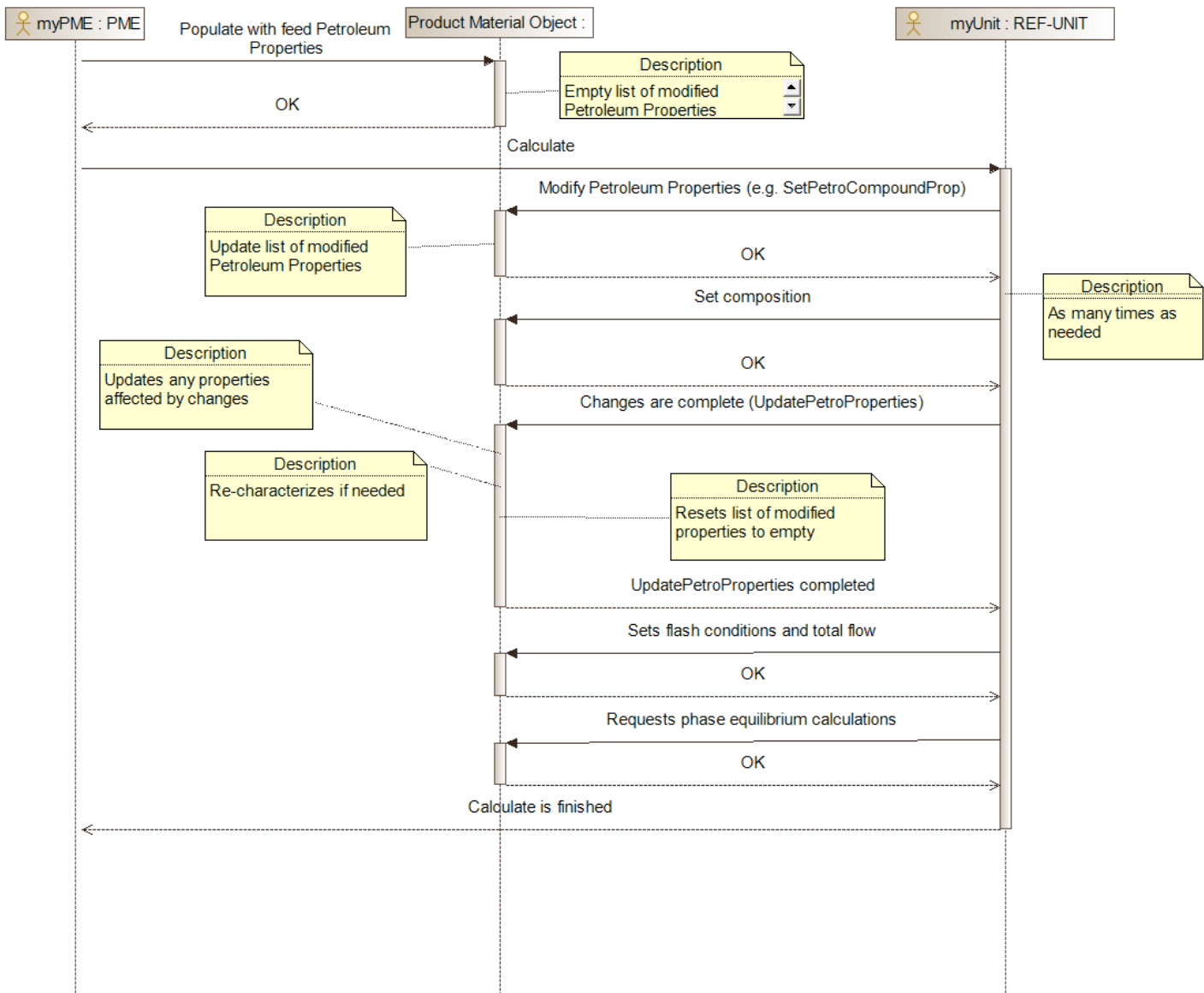


Figure 2 Main sequence diagram

4. Analysis and Design

4.1 Overview

The interface *ICapeThermoPetroleumFractions* is implemented on the Material Object, like the interface *ICapeThermoMaterial*, and serves for Refinery Unit Operations to access properties of Petroleum Fractions.

4.2 Interface diagram

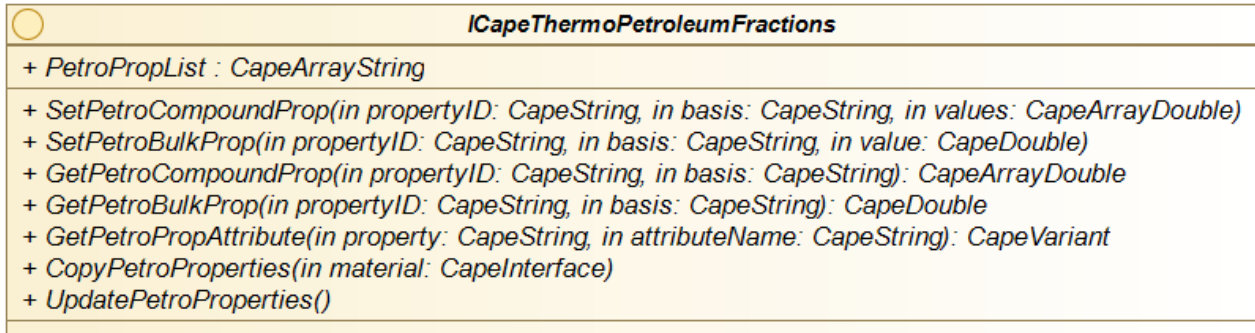


Figure 3 *ICapeThermoPetroleumFractions* interface diagram

4.3 State diagram

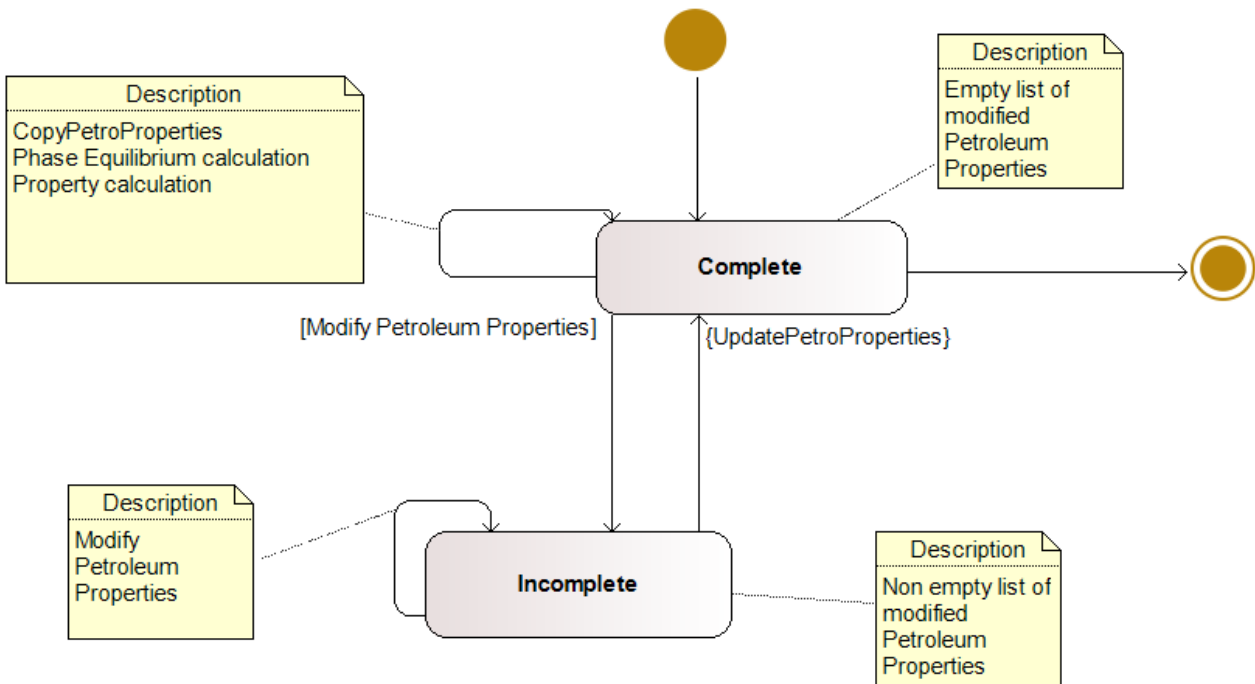


Figure 4 Material Object state diagram

4.4 Other diagrams

None

4.5 Interface Description

The interface proposed to cover the Use Cases presented above is named *ICapeThermoPetroleumFractions*.

Following solutions proposed in CAPE-OPEN for similar functionalities e.g., *ICapeThermoMaterial*, the interface *ICapeThermoPetroleumFractions* is designed as a container of properties. Since it is not reasonable to assume the operating procedures and functionality of every refinery operation, the interface, and the underlying calling patterns (as it will be shown later) do not impose any restriction regarding the amount and type of properties a refinery operation requires/provides from/to the hosting simulation environment.

4.5.1 ICapeThermoPetroleumFractions

Interface Name	ICapeThermoPetroleumFractions
Method Name	CopyPetroProperties
Returns	-

Description

Initializes the set of Petroleum Property values from another Material Object.

Arguments

Name	Type	Description
[in] source	CapeInterface	The Material Object from which to copy the present Petroleum Property values

Errors

No specific error. The following CAPE-OPEN error is applicable:

ECapeInvalidArgument: to be raised if argument material is invalid.

Notes

This method should be called by a REF-UNIT as the first step of defining a product stream. If omitted, Petroleum Properties that are not specified explicitly or indirectly (through property dependency) by the REF-UNIT are undefined and may have UNDEFINED values or may have values remaining from a previous configuration of the Material Object.

Interface Name	ICapeThermoPetroleumFractions
Method Name	GetPetroBulkProp
Returns	CapeDouble

Description

Returns the value of a Petroleum Property for the complete mixture as a whole.

Arguments

Name	Type	Description
[in] <i>propertyID</i>	CapeString	Petroleum Property to get (see Table 1: list of petroleum bulk- and compound properties)
[in] <i>basis</i>	CapeString	Calculation basis for the Petroleum Property (UNDEFINED, “mass”, “mole” or “volume”).

Errors

No specific error. Some CAPE-OPEN errors which are applicable:

ECapeInvalidArgument: to be raised either if *propertyID* does not correspond to a Petroleum Property supported by the PME or if *basis* specified is not supported by the PME for that Petroleum Property.

ECapeUnknown: to be raised in cases where *ECapeInvalidArgument* does not apply.

Interface Name	ICapeThermoPetroleumFractions
Method Name	GetPetroCompoundProp
Returns	CapeArrayDouble

Description

Returns values of a Petroleum Property for all Petroleum Fractions.

Arguments

Name	Type	Description
[in] <i>propertyID</i>	CapeString	Petroleum Property to get (see Table 1: list of petroleum bulk- and compound properties)
[in] <i>basis</i>	CapeString	Calculation basis for the Petroleum Property (UNDEFINED, “mass”, “mole” or “volume”).

Errors

No specific error. Some CAPE-OPEN errors which are applicable:

ECapeInvalidArgument: to be raised either if *propertyID* does not correspond to a Petroleum Property supported by the PME or if *basis* specified is not supported by the PME for that Petroleum Property.

ECapeUnknown: to be raised in cases where *ECapeInvalidArgument* does not apply.

Notes

The length of the array returned matches the number of Petroleum Fractions in compID (or all Petroleum Fractions on the Material Object in case compID is UNDEFINED). Petroleum Properties can only be obtained for Petroleum Fractions, not for real Compounds.

The Petroleum Properties supported by the PME should have been obtained prior to calling *ICapeThermoPetroleumFractions::GetPetroCompoundProp* using *GetPetroPropList* on the same interface. The basis in which the PME stores the Petroleum Property or in which the PME is additionally able to provide the Petroleum Property are attributes of any Petroleum Property that are obtained using *GetPetroPropAttribute* prior to calling *GetPetroCompoundProp*.

Interface Name	ICapeThermoPetroleumFractions
Method Name	GetPetroPropAttribute
Returns	CapeVariant

Description

Returns the value of a Petroleum Property attribute.

Arguments

Name	Type	Description
[in] <i>propertyID</i>	CapeString	Name identifying the Petroleum Property
[in] <i>attributeName</i>	CapeString	Name of the attribute requested

Errors

No specific error. The following CAPE-OPEN error applicable:

ECapeInvalidArgument: to be raised if either of the arguments is not valid.

Notes

If the method returns *ECapeInvalidArgument* for a valid property name, the attribute is not supported by the PME. In this case the default value applies as defined below.

The attributes that have no default value must be supported by the PME.

Name	Type	Default value	Meaning
Basis	CapeString		The internal basis in which the PME stores the Property: mass, mole, volume or UNDEFINED
SupportsMassBasis	CapeBoolean	TRUE if Basis is "mass"	Property can be set or obtained on a mass basis
SupportsMoleBasis	CapeBoolean	TRUE if Basis is "mole"	Property can be set or obtained on a mole basis
SupportsVolumeBasis	CapeBoolean	TRUE if Basis is "volume"	Property can be set or obtained on a volume basis
Method	CapeString	UNDEFINED	Measurement method name. See valid examples in Table 1.
IsBulkProperty	CapeBoolean	TRUE	Property is a bulk property
IsCompoundProperty	CapeBoolean	TRUE	Property is a Compound Property
IsReadOnly	CapeBoolean	FALSE	Property can only be obtained but not set

Interface Name	ICapeThermoPetroleumFractions
Property Name	PetroPropList
Returns	CapeArrayString

Description

Returns the list of identifiers for supported Petroleum Properties.

Arguments

Errors

No specific error.

Notes

This list may include identifiers of Petroleum Properties defined in Table 1, as well as custom Petroleum Properties designed for interaction between specific PME and PMC.

The list must not contain the same Petroleum Property identifier more than once.

Identifiers of Petroleum Properties must comply with the same rules as the component names in the Identification Common interface specification: identifier does not begin or end with white space, identifier does not contain control characters, i.e. any value with an ASCII below space=32 (including carriage return, line feed, tab, escape, form feed, null), delete, the range of U+0080—U+009F (C1 controls) ...).

Additional information on each Petroleum Property can be obtained using *GetPetroPropAttribute*.

Interface Name	ICapeThermoPetroleumFractions
Method Name	SetPetroBulkProp
Returns	-

Description

Sets the value of a Petroleum Property for the complete mixture as a whole.

Arguments

Name	Type	Description
[in] <i>propertyID</i>	CapeString	Petroleum Property to set (see Table 1: list of petroleum bulk- and compound properties)
[in] <i>basis</i>	CapeString	Calculation basis for the Petroleum Property (UNDEFINED, “mass”, “mole” or “volume”).
[in] <i>value</i>	CapeDouble	Value of the property.

Errors

No specific error. The following CAPE-OPEN errors are applicable:

ECapeInvalidArgument: one or more of the input arguments is not valid. For example, an unrecognized value for the *basis* argument.

ECapeUnknown: the error to be raised when other error(s), specified for the *SetPetroBulkProp* operation, are not suitable.

Interface Name	ICapeThermoPetroleumFractions
Method Name	SetPetroCompoundProp
Returns	-

Description

Sets values of a Petroleum Property for all Petroleum Fractions.

Arguments

Name	Type	Description
[in] <i>propertyID</i>	CapeString	Petroleum Property to set (see Table 1: list of petroleum bulk- and compound properties)
[in] <i>basis</i>	CapeString	Calculation basis for the Petroleum Property (UNDEFINED, "mass", "mole" or "volume").
[in] <i>values</i>	CapeArrayDouble	Values of the property. The length of the array must match the number of Compounds, Petroleum Fractions included. Petroleum Properties can only be specified for Petroleum Fractions, not for real Compounds (UNDEFINED for real Compounds).

Errors

No specific error. Some CAPE-OPEN errors which are possible:

ECapeInvalidArgument: one or more of the input arguments is not valid. For example, an unrecognized value for the *basis* argument.

ECapeUnknown: the error to be raised when other error(s), specified for the *SetPetroCompoundProp* operation, are not suitable.

Interface Name	ICapeThermoPetroleumFractions
Method Name	UpdatePetroProperties
Returns	-

Description

Instructs the Material Object to update dependent properties and revise characterised parameters for the Petroleum Fractions.

Arguments

None

Errors

No specific error.

Notes

This operation instructs the Material Object to update dependent properties and revise characterised properties parameters for the Petroleum Fractions, depending on the properties that have been modified since the start of the calculation, the last call to *CopyPetroProperties* or the last call to *UpdatePetroProperties*.

If composition or Petroleum Properties have been modified, it cannot be assumed that the thermodynamic state of the Material Object is consistent until *UpdatePetroProperties* is called. This function should therefore be called before calculating any properties, obtaining any properties, or calculating phase equilibria.

Overall fraction or overall compound flows must be specified prior to calling this method.

4.6 Scenarios – Proposed Scenario to Validate the Interface

4.6.1 Introduction

As described in this document, the proposed interface supports the operations of refinery units, by exposing a series of properties, characteristics of the oil industry. In this sense, the new interface increases the container capabilities of the Material Object.

This simplification of the project does not compromise or modifies the testing of the proposed interface. Thus, to test the *ICapeThermoPetroleumFractions* interface, the only requirements are: 1) to have a UNIT plug, that will act as a client of *ICapeThermoPetroleumFractions*, and 2) to modify the implementation of the Material Object to support the new interface and to provide the UNIT plug with the required petroleum properties.

The UNIT plug will need to be able to estimate pseudo compound properties, but this will be achieved using simulator native characterisation methods.

4.6.2 Validation Scenarios

A prototype UNIT socket has been first implemented in SimSci PRO/II 9.3. A prototype UNIT plug has been implemented by Shell Global Solutions. The interaction between the socket and plug has been tested, validated and is in production (iii).

5. Notes on the interface specifications

5.1 Requirements for Ref-UNIT

5.1.1 Accepting connections of Material Objects to Ports

A REF-UNIT should raise an error when a connection of a Material Object is made to one of its Ports if the Material Object does not implement *ICapeThermoPetroleumFractions*.

5.1.2 Setting values for Petroleum Properties

As part of setting Petroleum Properties, a REF-UNIT may not set values for the same property in different granularities. If possible, the Ref-UNIT should set values for petroleum compounds. The REF-UNIT should set a bulk value only if it cannot predict a finer granularity (compound values).

5.2 Requirements for PME

5.2.1 Basis conversions

As described in section 5.2.4 within the CAPE-OPEN Thermodynamic and Physical Properties interface specification (ii), the responsibility for basis conversions lies with the Material Object. The minimum requirement is that values should be available in the basis in which they have been stored; if data in a requested basis cannot be provided, a CAPE-OPEN error should be raised. Typically, for a particular Petroleum Property, the basis used within one company is fixed to a particular convention. Therefore, the minimum requirement of providing data only in the basis in which the data were stored (and not providing any conversions between bases) often suffices for Petroleum Properties.

5.3 List of Petroleum Properties

The list of standard properties is not exhaustive; a PME and PMC can use properties that do not appear in below tables. Such custom properties require that both the PME and PMC are aware of the definition of these properties.

BULK PROPERTIES AND COMPOUND PROPERTIES

There is one single list which defines bulk properties (one value for the entire mixture) and compound properties (one value for each petroleum fraction); all PME's may support the same properties for bulk and compounds; *GetListPetroBulkProp* and *GetListPetroCompoundProp* may return different lists.

In order to agree what a Petroleum Property reflects, Petroleum Properties are qualified by name, description, basis and a standard test method to obtain it (example "ASTM D611").

Elements of this last string (ASTM and D611 for example) are standardized in their way of display.

The assumption is that a Unit Operation could have the possibility to get a property referenced with a specific measurement method and to transform its value to a different measurement or reference method. Hence the need for the Unit Operation to know which the reference for a property name is. Same on output, using the inverse of the transformation correlation.

Having a separate method acting on a single property name is sufficient because most Unit Operations will make use of a small number of petroleum properties.

The format: method name starts with the acronym of the standard body (e.g., ASTM, EN, IP, ISO or UOP, no space in the acronym), followed by a space, followed by the reference identifier of the method (e.g., D611 or 391). Any additional information (like subversion or year) may follow and must be separated from the reference identifier by any character other than a digit or a letter (may be a hyphen, a space, or a colon for example). Properties that represent a content are expressed as mol/mol or kg/kg according to the basis. Elemental contents represent ratios of atoms of a specific element compared to the total. Other content properties represent the ratio of molecules of a specific family to the total. Properties that represent an index are dimensionless. All other properties that represent temperature, pressure, or thermodynamic properties are expressed in S.I. units.

The following notes apply to Table 1:

- Note 1: this Petroleum Property is only available as a bulk property.
- Note 2: this Petroleum Property is only available as a Compound property.
- Note 3: In case of an incoming request for this Petroleum Property, the PME may optimize the request of Petroleum Properties related to this one, by caching the values of all the related Petroleum Properties.

Table 1: list of petroleum bulk- and compound properties

Name	Description	Remarks
<Element>Content	Content of a particular element	This is a placeholder name for any of the elements in the periodic table. In the name, <Element> is replaced by the atomic symbol, e.g., "CContent", "HContent", "VContent", "NiContent", ...
AcentricFactor ²	Acentric factor	Defined as Pitzer acentric factor; typically part of characterized data.
AnilinePoint	Aniline Point	Expressed as a temperature. Example method: "ASTM D611-12(2016)"
APIGravity	API Gravity	Dimensionless. Given by the formula defined by American Petroleum Institute 141.5/SG-
AromaticsTotalContent	Total Aromatics content	Sum of AromaticsMono, AromaticsDi, AromaticsTri, AromaticsTetra. Example methods: "ASTM D5769", "IP 391", "IP 469", "ASTM D5186", "ASTM D6591-11(2017)", "ASTM D5580

Name	Description	Remarks
AromaticsMonoContent	Mono Aromatics content	Example methods: See AromaticsTotalContent
AromaticsDiContent	Di-aromatics content	Example methods: See AromaticsTotalContent
AromaticsTriContent	Tri-aromatics content	This may be Tri+ content for simulations in which AromaticsTetra is not exposed. Example methods: See AromaticsTotal
AromaticsTetraContent	Tetra+-aromatics content	Example methods: See AromaticsTotal
AsphalteneContent	Asphaltene content	Example methods: “ASTM D6560”, “IP 143”, “IP 469”
BasicNContent	Basic Nitrogen elemental content: represents the nitrogen that reacts away relatively easily.	
BromineIndex	Indication of the quantity of bromine-reactive constituents (olefins).	Example methods: “ASTM D2710–09(2013)”
BromineNumber	Indication of the quantity of bromine-reactive constituents (olefins).	Example methods: “IP 130”, “UOP 304-08”, “ASTM D1159”. Warning: frequently expressed without units but meaning gr bromine per 100 grams of feed. CAPE-OPEN property “BromineNumber” is expressed in kg/kg.
CetaneIndex	Cetane Index, an alternative measure for Cetane Number.	Example methods: “ISO 4264”, “ASTM D4737”, “ASTM D976”
CetaneNumber	Cetane Number, a measure of a fuel’s ignition delay.	Dimensionless. Example methods: “ASTM D613-17b”, “ASTM D7668”
ColdFilterPluggingPoint	Lowest temperature at which diesel passes through standardized filter device within a certain time.	Expressed as a temperature. Example methods: “ASTM D6371”, “EN 116”
CloudPoint	Temperature below which fuel appears cloudy	Example methods: “ASTM D2500”, “ASTM D7683”, “ASTM D5773”, “ASTM D7689”, “ASTM D7307”, “ASTM D5771”, “ASTM D5772”, “ASTM D5551”
ConradsonCarbonContent	Conradson Carbon content	“ASTM D189”
CriticalTemperature ²	Critical Temperature	Typically, part of characterized data.
CriticalPressure ²	Critical Pressure	Typically, part of characterized data.
CriticalVolume ²	Critical Volume	Typically, part of characterized data.

Name	Description	Remarks
D86-IBP ^{1,3}	D86 initial boiling point temperature	“ASTM D86”
D86-T5 ^{1,3}	D86 temperature for 5% volume recovered	“ASTM D86”
D86-T10 ^{1,3}	D86 temperature for 10% volume recovered	“ASTM D86”
D86-T20 ^{1,3}	D86 temperature for 20% volume recovered	“ASTM D86”
D86-T30 ^{1,3}	D86 temperature for 30% volume recovered	“ASTM D86”
D86-T40 ^{1,3}	D86 temperature for 40% volume recovered	“ASTM D86”
D86-T50 ^{1,3}	D86 temperature for 50% volume recovered	“ASTM D86”
D86-T60 ^{1,3}	D86 temperature for 60% volume recovered	“ASTM D86”
D86-T70 ^{1,3}	D86 temperature for 70% volume recovered	“ASTM D86”
D86-T80 ^{1,3}	D86 temperature for 80% volume recovered	“ASTM D86”
D86-T90 ^{1,3}	D86 temperature for 90% volume recovered	“ASTM D86”
D86-T95 ^{1,3}	D86 temperature for 95% volume recovered	“ASTM D86”
D86-FBP ^{1,3}	D86 final boiling point temperature	“ASTM D86”
FinalBoilingPoint	Upper boiling limit	Expressed as a temperature.
FlashPointOC	Flash Point Open cup	Expressed as a temperature. “ASTM D92”, “ASTM D1310”
FlashPointCC	Flash Point Closed cup	Expressed as a temperature. “ASTM D7094”, “ASTM D93”
FreezePoint	Freeze Point	Expressed as a temperature.
FRMax	Volume fraction of aromatic hydrocarbon in a diluent of aromatic and aliphatic hydrocarbon above which flocculation does not occur when mixing an oil into the diluent at any proportion.	“ASTM D7060”
HCRatio	Hydrogen to Carbon ratio	Expressed as moles per mole if on a mole basis. Expressed as mass per mass if on a mass basis.
InitialBoilingPoint	Lower boiling limit	Expressed as a temperature.
IsoParafinContent	Content of IsoParafins	

Name	Description	Remarks
KinematicViscosityAt100F	Kinematic Viscosity at 100°F	
KinematicViscosityAt210F	Kinematic Viscosity at 210°F	
LiquidDensityAt25C	Density of the liquid phase at standard normal conditions	Standard conditions are defined as 25°C and 1 atm.
LiquidDensityAt60F	Density of the liquid phase at 60°F and 1 atm	
LiquidDensityAt15C	Density of the liquid phase at 15°C and 1 atm	
MercaptanSulphurContent	Mercaptan Sulphur content	“ASTM D3227”
MolecularWeight	Relative molar mass	The number is equal to the mass of one molecule divided by 1/12 of the mass of one atom of ¹² C.
MON	Motor Octane Number	Dimensionless. “ASTM D2700”
NaptheneContent	Content of Napthenes	
NormalBoilingPoint	Average normal boiling point	Expressed as a temperature.
NormalParaffinContent	Normal Paraffin Content	
OlefinContent	Content of olefins	
ParaffinTotalContent	Total Paraffin content	
PeptizingPower	Volume fraction of aromatic hydrocarbon in a diluent at which the diluent has the same peptization (deflocculation) as the oil.	“ASTM D7060”
PourPoint	Pour Point	Expressed as a temperature. “ASTM D97”
RamsbottomCarbonContent	Carbon residue after evaporation and pyrolysis	“ASTM D524”
RefractiveIndex	Refractive Index	“ASTM D1218”
ReidVaporPressure	Reid Vapor Pressure	Expressed as a pressure. “ASTM D323”, “ISO 3007:1999”
ResinsContent	Resins content	“IP 469”
RON	Research Octane Number	Dimensionless. “ASTM D2699”
SaturatedRingContent	Content of saturated rings	

Name	Description	Remarks
SaturatesContent	Saturated compounds content	“IP 469”
SmokePoint	Smoke Point	Expressed as a temperature. “ASTM D1322”
SpecificGravity	Liquid density of a substance at 60°F divided by the density of water at 60°F.	
TrueVaporPressure	Bubble point pressure at 100°F	“ASTM D2879”
ViscosityIndex	Viscosity Index	“ASTM D2270”, “ISO 2909”
ViscosityAt100F	Dynamic Viscosity at 100°F	
ViscosityAt210F	Dynamic Viscosity at 210°F	
V50	Dynamic viscosity index based on viscosity at 50°C	Dimensionless.
WatsonK	WatsonK	Watson characterization factor defined as boiling point (Kelvin) times 1.8 raised to the power of one-third divided by the specific
WaxContent	Wax content	“UOP 46-64”

6. Specific Glossary Terms

6.1 Assay

An *assay* is a laboratory evaluation of a crude oil, producing molecular and chemical characteristics. Assay data can include inspection properties such as flash point, cloud point and/or comprehensive data obtained from physical or simulated distillation. Information obtained from the crude oil assay is used for detailed refinery engineering and client marketing purposes. Feedstock assay data are an important tool in the refining process distillation.

6.2 Blend

Mix in the context of Petroleum Fractions.

6.3 Characterization properties

The characterization properties are used to determine the characterized properties. Usually includes normal boiling point, molecular weight and density.

6.4 Characterized properties

Parameters necessary for thermodynamic models within a thermodynamic sub-system. A typical set of such parameters includes (pseudo-) critical temperature, critical pressure, acentric factor, parameters of ideal gas heat capacity correlation, etc... of Petroleum Fractions.

6.5 Coldflow properties

Subset of Inspection Properties that are measured at standard pressure and temperature.

6.6 Inspection properties

Subset of Petroleum Properties and refers to properties that can be quickly and easily measured.

6.7 Petroleum

Petroleum is a broad category that includes both crude oil and petroleum products.

6.8 Petroleum Fractions

A Petroleum Fraction or a pseudo compound is a number of real compounds lumped together into a single representative compound in order to simplify the modelling, typically within a narrow boiling cut or range. Since Petroleum Fractions represent multiple compounds, additional properties, such as sulphur content, are used to describe an average property of the actual compounds.

6.9 Petroleum Properties

Thermophysical properties of Petroleum Fractions (see 2.1, 2.2 and 2.3)

6.10 Pseudo-compounds

See Petroleum Fractions.

6.11 Refinery properties

See Petroleum Properties.

6.12 Refinery Unit Operation

A Refinery Unit Operation (REF-UNIT) is a unit operation that uses or modifies Refinery properties of Petroleum Fractions.

7. Bibliography

- (i) CAPE-OPEN Interface Specifications: Unit Operation Specification
- (ii) CAPE-OPEN Interface Specifications: Thermodynamic and Physical Properties Version 1.1
- (iii) Petroleum Properties via CAPE-OPEN: from prototype to commercial application, Short Abstract AIChE Fall meeting 2012 (Pittsburgh), Richard Baur, Ashok Dewan (Shell Global Solutions), Jasper van Baten (AmsterCHEM), David Jerome, Penukonda Krishna Murthy (Invensys SimSci-Esscor)

8. Appendix 1: Requirements pertaining to interaction between a PME and a Property Package

This section is not part of the specification and is provided only for future reference. It is envisioned that this section will add to the current requirements as defined so far.

If the thermodynamics are provided by an external software component, i.e. by a Property Package, the characterized properties need only be known to the Property Package. From the UNIT point of view there must not be any difference whether the thermodynamics are native to the PME or provided by a Property Package. Therefore, in the interaction between the UNIT and the PME, the methodology for setting and getting petroleum properties may not change from the above. More-over, the same workflow as above holds, so that. The requirement to inform the Property Package that a re-characterisation and update of affected petroleum properties is needed lies with the PME.

REQ-08: the characterized properties are private to the Property Package.

Rationale: content and format of the characterized properties are not prescribed by the standard.

REQ-09: the re-characterization must be performed by the Property Package.

Rationale: content and format of the characterized properties are not prescribed by the standard.

REQ-10: Material Object is associated with only one set of characterized properties.

Rationale: just like a Material Object is associated with a single thermodynamic sub-system, a Material Object can only behave according to a single set of characterized properties.

REQ-11: multiple sets of characterized properties are possible within a Flowsheet.

Rationale: for a Refinery Unit, one would expect the feed and the output to be characterized differently and the interface design should allow for that. Having only a global set of characterized properties for the Property Package, would oblige re-characterization to take place each time the active Material Object on the Property Package is switched, resulting in prohibitive CPU time.

REQ-12: the interface design must allow the PME to share sets of characterized properties between multiple Material Objects.

Rationale: copying characterized properties for all Material Objects that have the same Petroleum Properties may result in a large memory footprint and CPU overhead.

REQ-13: the PME must be able to associate the set of characterized properties of one Material Object to another Material Object.

Rationale: although the characterized properties are private to the Property Package, it is the PME which is aware of the connectivity between Unit Operations and of the sharing pattern between Material Objects.

REQ-14: *UpdatePetroProperties* on a given Material Object must result in providing a new set of characterized properties to that Material Object without breaking the association between other Material Objects sharing the old set.

Rationale: the approach suggested here is that the Material Object allows for storage of a generic object that is owned, defined and maintained by the Property Package; this object can either directly store the required characterized properties or provide the Property Package with a method to access previously stored characterized properties. It must be foreseen that this object is shared between multiple Material Objects, like the storage for the petroleum properties. It is the responsibility of the PME to ensure that getting and setting this object on one Material Object in effect gets and sets the object on all Material Objects that share the same (compound based) refinery property values.

REQ-15: phase equilibrium on each product Port of a REF-UNIT must be called after the last call to *UpdatePetroProperties* (if any).

Rationale: after *Calculate* of UNIT, the product streams of the UNIT must be at phase equilibrium. Should a re-characterization of petroleum properties be performed on a Material Object after a call to *CalcEquilibrium*, the product stream is no longer at phase equilibrium. Consequently, the REF-UNIT must not change Petroleum Properties on any outlet Material Properties after *CalcEquilibrium*.