INTERNATIONAL UNION OF PURE AND APPLIED CHEMISTRY

COMMITTEE ON PRINTED AND ELECTRONIC PUBLICATIONS [a]

XML-BASED IUPAC STANDARD FOR EXPERIMENTAL, PREDICTED, AND CRITICALLY EVALUATED THERMODYNAMIC PROPERTY DATA STORAGE AND CAPTURE (ThermoML)[b,c]
(IUPAC Recommendations 2005)

Prepared for publication by
M. FRENKEL,[1,d] R. D. CHIRICO,[1] V. V. DIKY,[1] Q. DONG,[1] K. N. MARSH,[2]
J. H. DYMOND,[3] W. A. WAKEHAM,[4] S. E. STEIN,[5] E. KOENIGSBERGER[6]

[1]Thermodynamics Research Center (TRC Group), Physical and Chemical Properties Division, National Institute of Standards and Technology, Boulder, CO 80305-3328, USA
[2] Department of Chemical and Process Engineering, University of Canterbury, Private Bag 4800, Christchurch, New Zealand
[3] Chemistry Department, University of Glasgow Glasgow G12 8QQ, UK
[4] School of Engineering Sciences, University of Southampton Highfield, Southampton SO17 1BJ, UK
[5]Physical and Chemical Properties Division, National Institute of Standards and Technology, Gaithersburg, MD 20899- 8380, USA
[6]Division of Science and Engineering, School of Mathematical and Physical Sciences, Murdoch University, Murdoch, WA 6150, Australia

---

[a] Membership of the Committee on Printed and Electronic Publications during the final preparation of the report (2005) was as follows: *President:* L. Glasser (Australia); *Secretary:* A. Davies (Germany); *Members:* J. R. Bull (South Africa), S. Heller (USA), D. Martinsen (USA), S. E. Stein (USA), B. Valter (Czech Republic), B. Vickery (UK).

[b] This article is a contribution of the National Institute of Standards and Technology and is not subject to copyright in the United States.

[c] Republication or reproduction of this report or its storage and/or dissemination by electronic means is permitted without the need for formal IUPAC permission on condition that an acknowledgment, with full reference to the source, along with use of the copyright symbol @, the name IUPAC, and the year of publication, are prominently visible. Publication of a translation into another language is subject to the additional condition of prior approval from the relevant IUPAC National Adhering Organization.

[d] To whom correspondence should be addressed. Email: frenkel@boulder.nist.gov.

Abstract

ThermoML is an XML-based emerging IUPAC standard for storage and exchange of experimental, predicted, and critically evaluated thermophysical and thermochemical property data. The basic principles, scope, and description of all structural elements of ThermoML are discussed. ThermoML covers essentially all thermodynamic and transport property data (more than 120 properties) for pure compounds, multicomponent mixtures, and chemical reactions (including change-of-state and equilibrium reactions). The ThermoMLEquation schema for representation of fitted equations with ThermoML is also described. The role of ThermoML in global data communication processes is discussed. The text of a variety of data files (use cases) illustrating the ThermoML format for pure compounds, mixtures, and chemical reactions, as well as the complete ThermoML schema text, are provided as Supporting Information.

Introduction

Thermodynamic property data represent a key foundation for development and improvement of all chemical process technologies. However, rapid growth in the number of custom-designed software tools for engineering applications has created an interoperability problem between the formats and structures of thermodynamic data files and required input/output structures for the software applications. Establishment of efficient means for thermodynamic data communications is absolutely critical for provision of solutions to such technological challenges as elimination of data processing redundancies and data collection process duplication, creation of comprehensive data storage facilities, and rapid data propagation from measurement to data-management system and from data-management system to engineering application. Taking into account the diversity of thermodynamic data and numerous methods of their reporting and presentation, standardization of thermodynamic data communications is very complex.

A brief review of the standardization efforts for thermodynamic data communications was compiled recently [1]. Efforts to develop a standard for thermophysical and thermochemical property data exchange [2] were first initiated in the early 1980s, reflecting a new trend in data collection through design of electronic databases, which became possible due to the rapid development of computer technology. From 1985 to 1987, the Thermodynamics Research Center (TRC, then with Texas A&M University) developed the first prototype of such a standard called COSTAT (Codata STAndard Thermodynamics) [3]. This prototype was discussed extensively among numerous institutions worldwide through the auspices of CODATA. This effort played an important role in establishing the necessity of a standard and in formulating the basic principles that must be incorporated. Practical implementation of COSTAT was hindered significantly by limitations of software tools available at the time.

At the beginning of the 1990s, Global CAPE Open (initially Cape- Computer Aided Process Engineering- Open) technology was developed [4]. The Global CAPE Open project was established to develop standards for interfaces of software components of a process simulator. The main objective of the project was to enable native components of a simulator to be replaced by those from another source with minimal effort in as

seamless a manner as possible. This approach was proven successful; however, the Global CAPE Open approach is not naturally modular, and therefore, implementation of any modifications of the thermodynamic data representation requires significant programming effort.

In 1998, TRC was selected as one of four data centers worldwide to be a part of a similar project funded by CODATA (IUCOSPED Task Group). A number of experts from NIST actively participated in this project, which ended in 2002. This project led to the development of the SELF [5] files closely associated with the ELDATA electronic journal formats. Though the project played a positive role in attracting the attention of the international scientific community to core issues related to thermophysical data standardization, the final outcome has profound limitations related to its non-comprehensive and non-systematic nature.

In 1999, the Design Institute for Physical Property Data (DIPPR®) under the auspices of the American Institute of Chemical Engineers (AIChE) initiated Project 991 to develop a thermophysical property data exchange standard focusing primarily on the industrial application of the extended version of the CAPE Physical Property Data eXchange neutral file format (PPDX) [6], and later developed its XML version PPDXML.

In 2002 IUPAC approved the project 2002-055-3-024, "XML-based IUPAC Standard for Experimental and Critically Evaluated Thermodynamic Property Data Storage and Capture," and established a Task Group [7] to create standardized mechanisms for thermodynamic data communications with XML (Extensible Markup Language) technology. This project is an activity of the Committee on Printed and Electronic Publications [8]. The recommendations provided here are outcome of this project. XML technology [9], fully developed within the last 5 years, provides significant advantages for the development of standards for data exchange, such as its native interoperability based on ASCII code, its modular nature, and transparent readability by both humans and computers. From a practical standpoint, it is also critical that this technology is currently supported by both the software and hardware industries. The Task Group approved the name "ThermoML" for the emerging IUPAC standard [10] and authorized the establishment of the corresponding namespace on the IUPAC Web site [11]. Among other X-markup languages, CML (XML for chemistry) [12] and MatML (XML for

primarily mechanical properties of the materials) [13] are most closely related to the ThermoML.

The Task Group conducted three meetings. The first meeting held in London (UK) in January 2004 resulted in approval of the overall framework for ThermoML, including its application to experimental thermodynamic property data and representation of uncertainties. The second meeting held in Beijing (China) in August 2004 led to the approval of the description of predicted thermodynamic property data, critically evaluated thermodynamic property data, and fitting equations. At the third meeting held in Sesimbra (Portugal) in April 2005, the present recommendations were discussed and received preliminary approval.

The IUPAC standard (ThermoML) for thermodynamic data storage and exchange has been developed and is described below. A new global thermodynamic data communication process has been established on the basis of ThermoML that involves major journals in the field of thermodynamics and various industrial organizations. The software infrastructure has been developed to provide support for full realization of this process. The ThermoML standard is described completely in the following sections. Components of the global communication process for thermodynamic data are described at the end of this article.

Much of the material given below was published previous in a series of three articles describing the original formulation of ThermoML for representation of experimental data [14], extensions to the schema for representation of uncertainties [15], and further extensions for representation of predicted data, critically evaluated data, and fitting equations [16]. Every effort was made to ensure that information represented with the formats described in these earlier articles would remain valid within the new IUPAC standard version of ThermoML. Several minor changes were made to improve consistency in tag names, and to eliminate unnecessary elements. These changes might invalidate files created with the earlier version of ThermoML, and could require minor adjustment in the file structure to bring it into compliance with the new schema definitions. Details of these minor changes are provided as Supplementary Information to this article.

Several new extensions are described in the present article. These additions provide for representation of properties of ions and polymers, and provide for more comprehensive compound identification through implementation of the IUPAC International Chemical Identifier (InChI) [17] and more-extensive sample characterization options.

Basic Principles

Schema Structure. The ThermoML structure represents a balanced combination of hierarchical and relational elements. The schema structure explicitly incorporates structural elements related to basic principles of phenomenological thermodynamics: thermophysical, thermochemical, and transport properties, state variables, system constraints, phases, and units. Meta- and numerical data records are grouped into 'nested blocks' of information corresponding to data sets. Metadata records precede numerical data information providing a robust foundation for generating 'header' records for any relational database where ThermoML-formatted files might be incorporated. The structural features of the ThermoML metadata records ensure unambiguous interpretation of numerical data and allow data-quality control based on the Gibbs Phase Rule. Implementation of the Gibbs Phase Rule is a reflection of long-standing traditions and practices at NIST for assuring the highest quality in data, and would provide users with an indication of thermodynamic data inconsistencies before the data are deposited into a data-storage facility [18]. Moreover, some detailed information included in the metadata records could serve as a background for independent assessment of uncertainties, which could be propagated into uncertainties of physical parameters for reaction streams, and consequently, provide an opportunity for numerical characterization of the quality of a chemical process design [19].

Tagging. Commonly accepted IUPAC-based terminology is used as a foundation for metadata and numerical data tagging. ThermoML capitalizes on the fact that XML files are essentially textual files and can, in principle, be interpreted without customized software. In addition, the self-explanatory approach and very limited use of abbreviations minimizes the time necessary for users to understand the schema and to convert the ThermoML formatted data with customized software or commercial XML parsers.

Modularity. ThermoML is designed to take advantage of the modular nature of XML schemas. Structurally, it can be expanded easily into areas that are currently beyond its scope.

Units. By design, there is only one unit selected for each property covered by ThermoML. These units are SI-based, however, for a number of properties the selected units are multiples of SI units to ease interpretation of numerical values. Unit tagging is explicitly propagated to every numerical data point in a ThermoML file as a part of each property name, thus minimizing the possibility of unit misinterpretation.

Data Representation. Various methods of numerical data representation commonly used in publication of experimental property data (*e.g.*, direct, difference with values in a reference state, ratio of the value to that in a reference state, *etc*.) are incorporated into ThermoML.

Scope

ThermoML covers essentially all experimentally determined thermodynamic and transport property data (more than 120 properties) for pure compounds, multicomponent mixtures, and chemical reactions (including change-of-state and equilibrium). ThermoML allows storage and exchange of property data with full allowance for data provenance. Full specification of the data source (bibliographic information), method of property generation (experimental, predicted, critically evaluated), and multiple uncertainty assessments (with assessors specified) are included. Expansion of ThermoML to include properties of polymers and ionic systems was not included previously, but is included with release of this IUPAC standard. Common properties, which do not have unambiguous thermodynamic definitions, such as decomposition temperature, flammability limit, octane number, *etc.*, are not included. The list of all properties within the scope of ThermoML is provided with the complete schema description.

Conventions for Names of Elements in the ThermoML Schema

The names (or "*tags*") include special characters related to the type of information to be stored. A name beginning with "e" indicates an *enumeration* element (with values selected from a predefined list), "s" designates *string* elements (text strings), "n" specifies *numerical* elements (integer or floating), "yr" designates elements characterizing the

year, "date" specifies *date* elements, and "url" indicates elements specifying addresses on the World Wide Web. Elements shown as dotted boxes in the figures are optional, while those shown as solid-lined boxes are mandatory. A "complex" element is an element that includes subelements. Complex elements illustrated without their internal structure are identified by "+" at the right-hand edge of the box. Multiple elements of the same type are often needed within the schema to specify such things as multiple authors for a given citation or multiple property values for a given data type. These multiple elements are identified in the figures by lower and upper limits listed below the relevant boxes. The only the limits used are "0...∞" for optional elements and "1...∞" for mandatory elements.

Description of the ThermoML Schema

ThermoML consists of four major blocks and a block for version specification, as shown in Figure 1.

(1) *Citation* (description of the source of the data).

(2) *Compound* (characterization of the chemical system). The compound description is linked to a description of the sample that includes its initial source and purity, purification methods used, and final purity with specification of the method(s) of purity determination.

(3) *PureOrMixtureData* (meta- and numerical property data for a pure compound or multicomponent mixture).

(4) *ReactionData* (meta- and numerical property data for a chemical reaction with thermodynamic state change or in a state of chemical equilibrium).

One additional element **Version** [complex] is mandatory and provides for storage of the ThermoML version designation. The subelements of **Version** [complex] are **nVersionMajor** [numerical, integer] and **nVersionMinor** [numerical, integer]. For example, if the version number of ThermoML were 2.1, the "major" element would store the value "2" and the "minor" element would store the value "1".

***1. Citation* Block.** The schema for the *Citation* block is shown in Figure 2, and full descriptions of the elements are given here. Simple elements (*i.e.*, those without internal structure) are listed first, followed by complex elements.

eType [enumeration] indicates the type of source document (book, journal, report, patent, thesis, conference proceedings, archived document, personal correspondence,

published translation, unspecified). [*Note:* The associated enumeration lists are provided in brackets immediately after introduction of an enumeration element in the text.] **eSourceType** [enumeration] provides information about the nature of the source of information (original article, Chemical Abstracts, other). **sDocumentOrigin** [string] characterizes the origin of the document, such as a company name, institution, or conference. **sAuthor** [string] stores the name of an author. The symbol "0...∞" indicates that any number of authors can be specified with each name in a separate **sAuthor** [string] element. **sPubName** [string] stores the name of the publication where the citation was published, such as the name of a journal or a book title. **yrPubYr** [year] represents the year of publication. **dateCit** [date] is the date of creation of the ThermoML file. **sTitle** [string] stores the title of the cited document. Typically, this is the title of a journal article. **sAbstract** [string] is the abstract for the document. **sKeyword** [string] stores keywords for the document. **sDOI** [string] allows explicit storage of the DOI (Digital Object Identifier [[20]]) for a particular document on the World Wide Web. **urlCit** [url] stores a url for the citation and is designed for information reported on the World Wide Web. **sCASCit** [string] allows storage of the Chemical Abstract Service citation code. **sIDNum** [string] represents a local or global reference identifier. **sLocation** [string] is a string element that can be used for information such as a conference location or publisher location. **sVol** [string] stores the volume number of the citation. **sPage** [string] is the page range for the citation.

TRCRefID, **book**, **journal**, and **thesis** are complex elements within the *Citation* block. Their structures are shown in the Figure 3. **TRCRefID**, the TRC reference identifier consists of **yrYrPub** [year] the year of publication, **sAuthor1** [string] the first three characters of the first author's last name, **sAuthor2** [string] the first three characters of the second author's last name, and **nAuthorn** [numerical, integer] a numerical value to assure uniqueness of each **TRCRefID**.

Books, journals, and theses are characterized with additional tags. For **book** [complex]**: sChapter** [string] contains the chapter identifier, **sEdition** [string] the edition identifier, **sEditor** [string] the editor name(s), **sISBN** [string] specifies the International Standard Book Number, and **sPublisher** [string] stores the identity of the publisher. For **journal** [complex] the following items are specified**: sISSN** [string] specifies the

International Standard Subscription Number, **sIssue** [string] the issue identifier, and **sCODEN** [string] the CODEN identification of the journal. (CODEN are unique, six-character codes that identify serial and nonserial publications produced worldwide.) For **thesis** [complex]: **sDeg** [string] the academic degree designation such as M.S., Ph.D., *etc*.), **sDegInst** [string] (the name of the institution granting the academic degree), and **sUMIPubNum** [string] the University Microfilm International Publication Number are designated.

***2. Compound* Block.** The schema for the *Compound* block is represented in Figure 4. **RegNum** [complex] is a compound registry number. The Chemical Abstract Service Registry Number **nCASNum** [numerical, integer] and an identification number assigned by a user organization **nOrgNum** [numerical, integer] are supported. The very recently developed *IUPAC International Chemical Identifier* [21] is supported in ThermoML and is stored in **sInChI** [string]. Compounds can be characterized with a variety of chemical names: **sCASName** [string] the Chemical Abstract Service name, **sIUPACName** [string] the name specified by IUPAC, and **sCommonName** [string], which allows any other name. Other elements in the *Compound* block are **sFormulaMolec** [string] the elemental molecular formula, and **sSmiles** [string] the SMILES notation (Simplified Molecular Input Line Entry System) that describes the chemical formula.

Two new elements are now included in the *Compound* block to accommodate information related directly to ions and polymers. **nCharge** [numerical, integer] stores the charge for an ion. **Polymer** [complex] includes a series of elements for storage of polymer characterization information: **nNumberAvgMolWt** [numerical, floating], the number average molecular weight; **nPeakAvgMolWt** [numerical, floating], the peak average molecular weight; **nViscosityAvgMolWt** [numerical, floating], the viscosity average molecular weight; **nWeightAvgMolWt** [numerical, floating], the weight average molecular weight; **nZAvgMolWt** [numerical, floating], the Z-average molecular weight**;** and **nPolydispersityIndex** [numerical, floating], the polydispersity index. Definitions of the various average molecular weights for polymers are available from reference 22.

The **Sample** [complex] element consists of four subelements, as shown in Figure 5. These are **nSampleNm** [numerical, integer] used to distinguish different samples of the same compound, **eSource** [enumeration] to indicate the original source of the sample

before purification (commercial source, synthesized by the authors, synthesized by others, isolated from a natural product, standard reference material (SRM), not stated in the document), **eStatus** [enumeration] to indicate the status of the sample description (unknown, not described, described in a previous document, no sample used), and the element **purity** [complex] to provide information related to the purity of the sample.

The element **purity** [complex] consists of **nStep** [numerical, integer] a sequential number corresponding to a purification stage, **ePurifMethod** [enumeration] the purification method applied at the specified step (Impurity adsorption, Vacuum degasification, Chemical reagent treatment, Crystallization from melt, Crystallization from solution, Liquid chromatography, Dried with chemical reagent, Dried in a desiccator, Dried by oven heating, Dried by vacuum heating, De-gassed by boiling or ultrasonically, De-gassed by evacuation, De-gassed by freezing and melting in vacuum, Fractional crystallization, Fractional distillation, Molecular sieve treatment, Unspecified, Preparative gas chromatography, Sublimation, Steam distillation, Solvent extraction, Salting out of solution, Zone refining, Other, None used) or **sPurifMethod** [string] the purification method, if it is not listed in the values for **ePurifMethod** [enumeration]. The element **purity** [complex] also includes **eAnalMethod** [enumeration] the analytical method used to determine the purity after a purification stage (Chemical analysis, Difference between bubble and dew points, Density, DSC, Estimation, Gas chromatography, Fraction melting in an adiabatic calorimeter, Mass spectrometry, NMR (proton), NMR (other), Not known, Spectroscopy, Thermal analysis using temperature-time measurement, Acid-base titration, Other types of titration, Mass loss on drying, Karl Fisher titration, HPLC, Ion chromatography, Ion-selective electrode, CO2 yield in combustion products, Estimated by the compiler, Stated by supplier, Other) or **sAnalMethod** [string] the analytical method used, if it is not listed as a value for **eAnalMethod** [enumeration].

Four elements are provided within **purity** [complex] (Fig 5) for specification of purity; **nPurityMol** [numerical, floating] the mole percent purity, **nPurityMass** [numerical, floating] the mass percent purity, **nPurityVol** [numerical, floating] the volume percent purity, and **nUnknownPerCent** [numerical, floating] the percent purity of unknown type. Additionally, three elements are provided for specific types of

impurities; **nWaterMassPerCent** [numerical, floating] the mass percent of water, **nHalideMolPercent** [numerical, floating] the mole percent of halide impurity, and **nHalideMassPercent** [numerical, floating] the mass percent of halide impurity. Halide impurities were added explicitly to accommodate sample characterization information for ionic liquids.

Numerical values of uncertainty are not provided for sample purities, but the number of digits is represented. The number of digits specified should correspond approximately to the number of significant digits, but there is no strict requirement for this correspondence. A comprehensive uncertainty-specification scheme for purities would add unjustifiable complexity to the schema. As shown in figure 5, an element for the number of digits associated with each type of purity representation is given. These elements are **nPurityMolDigits** [numerical, integer], **nPurityMassDigits** [numerical, integer], **nPurityVolDigits** [numerical, integer], **nUnknownPerCentDigits** [numerical, integer], **nWaterMassPerCentDigits** [numerical, integer], **nHalideMolPerCentDigits** [numerical, integer] **nHalideMassPerCentDigits** [numerical, integer].

***3. PureOrMixtureData* Block.** The schema for the *PureOrMixtureData* block is shown in Figure 6. This block contains non-bibliographic information about the source of the ThermoML file contents, identifies the experimental purpose, specifies meta- and numerical data, and specifies the compound (or mixture) and particular samples to which the data are related. The subelement **Equation** [complex] is used for representation of fitted equations and is described later in this document.

The element **nPureOrMixtureDataNumber** [numerical, integer] is a number that is unique for each instance of the *PureOrMixtureData* block in a ThermoML file and is used for external linking (if needed) in equation representation. The source of the ThermoML file is recorded through the following elements: **sCompiler** [string] the name of the person who compiled the data contained in the ThermoML file, **sContributor** [string] an identifier for a particular project, institution, or general source of the ThermoML file, and **dateDateAdded** [date] the date that the particular instance of the *PureOrMixtureData* block was created. The experimental-purpose element, **eExpPurpose** [enumeration] provides a general description of the purpose of the

experiment (Principal objective of the work, Secondary purpose (by-product of other objective), Determined for identification of a synthesized compound).

The compound or mixture associated with the property data is identified by the element **Component** [complex] (Fig. 7) consisting of **RegNum** [complex] and a sample number, **nSampleNm** [numerical, integer]. The **RegNum** [complex] structure was described earlier (Fig. 4). The identities of the phases present in equilibrium for the chemical system are represented with **PhaseID** [complex] (Fig 7) consisting of **ePhase** [enumeration] (crystal 4, crystal 3, crystal 2, crystal 1, crystal, crystal of unknown type, crystal of intercomponent compound 1, crystal of intercomponent compound 2, crystal of intercomponent compound 3, metastable crystal, glass, smectic liquid crystal, nematic liquid crystal, cholesteric liquid crystal, plastic crystal, liquid, liquid mixture 1, liquid mixture 2, fluid [supercritical or subcritical phases], ideal gas, gas, air at 1 atmosphere), eCrystalLatticeType [enumeration] (cubic, tetragonal, hexagonal, rhombohedral, orthorhombic, monoclinic, triclinic), and **RegNum** [complex] (Fig. 4), if needed.

Metadata are described by the four elements **Property** [complex], **Constraint** [complex], and **Variable** [complex]. **Property** [complex] (Fig. 8) is characterized by **Property-MethodID** [complex], which identifies the property and experimental method used, **PropPhaseID** [complex] indicates the phase associated with the property and has subelements analogous to those of **PhaseID** [complex] (Fig. 7) , **ePresentation** [enumeration] indicates the mathematical form used to report the data (Direct value,X; Difference between upper and lower temperature, $X(T_2)$-$X(T_1)$; Difference between upper and lower pressure, $X(p_2)$-$X(p_1)$; Mean between upper and lower temperature, $[X(T_2)+X(T_1)]/2$; Difference with the reference state, $X-X_{ref}$; Ratio with the reference state, X/X_{ref}; Ratio of difference with the reference state to the reference state, $[X-Xref]/X_{ref}$), **eRefStateType** [enumeration] describes the thermodynamic reference state if required (reference phase with the same composition at fixed temperature and pressure, reference phase with the same composition, temperature and pressure, reference phase at fixed temperature and the same pressure, reference phase at the same temperature and fixed pressure, phase in equilibrium with primary phase at the same temperature and pressure, pure components in the same proportion at the same temperature and pressure, pure solvent at the temperature of the same phase equilibrium, pure solvent at the same

temperature and pressure, pure solute at the same temperature and pressure), **nRefTemp** [numerical, floating] lists the value of a reference temperature, **nRefPressure** [numerical, floating] lists the value of a reference pressure, **RefPhaseID** [complex] indicates the reference phase for a particular data set, **Solvent** [complex] identifies the solvent used, **eStandardState** [enumeration] indicates the thermodynamic standard state if required by the property definition (pure compound, hypothetical pure solute, hypothetical unit molality solute, hypothetical unit molarity solute, infinite dilution solute), and **nPropNumber** [numerical, integer] is a sequential property number for the case in which multiple properties are listed as a function of the same variable values. Provision for **nPropNumber** is convenient for storage of tie-line phase equilibria data.

nRefTemp [numerical, floating] and **nRefPressure** [numerical, floating] represent the values of the reference temperature and reference pressure, if required. **RefPhaseID** [complex] (Fig. 8) consists of **RegNum**, which is necessary in cases where the reference phase is a pure compound phase and is used in the representation of mixture data, and **eRefPhase** [enumeration], specifies the reference phase. The subelements of **RefPhaseID** [complex] are analogous to those of **PhaseID** [complex] (Fig.7). The enumeration lists are the same.

There are five subelements of **Property** [complex] (Fig. 8) that are associated with representation of uncertainties. These are **CombinedUncertainty** [complex], **PropUncertainty** [complex], **PropRepeatability** [complex], **PropDeviceSpec** [complex], and **CurveDev** [complex]. These are described later in the section titled *Representation of Uncertainties*.

The element **PropertyMethodID** [complex] includes **PropertyGroup** [complex] and **RegNum** [complex] (Fig. 8). **RegNum** [complex] has the same structure as described earlier, but should be used for mixtures only if the property definition involves a specific component, such as mole fraction of a particular compound. The **PropertyGroup** [complex] element includes ten property groups: **Criticals**, **VaporPBoilingTAzeotropTandP** (an abbreviation of *"Vapor pressure, Boiling temperature, and Azeotropic temperature and pressure"*), *etc.*, as listed in the upper right of Fig. 8. Thermophysical properties are divided into these 10 groups to simplify the property-selection process for the ThermoML user. Each group is characterized by

ePropertyName [enumeration] and **eMethodName** [enumeration], and the elements **CriticalEvaluation** [complex] and **Prediction** [complex]. These are shown in expanded form for the **Criticals** group in Figure 8. Methods enumerated within **eMethodName** are *experimental* in nature. If the option "Other" is used as a value for **eMethodName**, **sMethodName** [string] can be used to identify the experimental method. Methods associated with property *prediction* and *critical evaluation* are represented separately to allow clear distinction between the three property sources; *experiment*, *critical evaluation*, and *prediction*.

The structure of the **Prediction** [complex] and **CriticalEvaluation** [complex] subelements are shown in Fig. 9. **Prediction** [complex] contains one mandatory element **ePredictionType** [enumeration] and three optional elements: **sPredictionMethodName** [string]**, sPredictionMethodDescription** [string], and **PredictionMethodRef** [complex] (Fig. 9). The **ePredictionType** [enumeration] element allows selection of one major type of prediction method (*ab initio*, molecular dynamics, semiempirical quantum methods, statistical mechanics, corresponding states, correlation, and group contribution). The element **sPredictionMethodName** [string] serves to identify the name of the prediction method. This is particularly helpful in identifying the method, if this method name is well established. **sPredictionMethodDescription** [string] could be used to describe the principal features of the method, its limitations, assumptions, *etc*. If the method used for the prediction has been described in the literature, the element **PredictionMethodRef** [complex] can be used to identify the original reference(s). **PredictionMethodRef** [complex] has the same structure as the major element **Citation** [complex] (Fig. 2).

The element **CriticalEvaluation** [complex] (Fig. 9) allows selection of one of three elements: **SingleProp** [complex], **MultiProp** [complex], or **EquationOfState** [complex]. **SingleProp** [complex] is designed to identify a critical evaluation method based on analysis for a single property only without consideration of inter-property consistency. The property is identified in the **ePropName** [enumeration] element (Fig. 8). **SingleProp** [complex] contains **sEvalSinglePropDescription** [string] and the multiple element **EvalSinglePropRef** [complex]. For example, the method used for critical evaluation of density data along the saturation line published recently [23] could be described in **sEvalSinglePropDescription** [string] as, "*Weighted least-square fitting to a* 4^{th}*-order*

polynomial at low temperatures and weighted-least-squares fitting to the Guggenheim equation at higher temperatures. The weights are based on the uncertainties of the selected experimental data." Information about reference 23 could be incorporated into the element **EvalSinglePropRef** [complex], which has the same structure as **Citation** [complex] (Fig. 2) and allows full specification of the reference.

MultiProp [complex] (Fig. 9) allows identification of a critical evaluation method as enforcing mutual consistency for a limited number of related properties. The structure of **MultiProp** [complex] is similar to the structure of **SingleProp** [complex] with **sEvalMultiPropDescription** [string] analogous to **sEvalSinglePropDescription** [string]**,** and **EvalMultiPropRef** [complex] is similar to **EvalSinglePropRef** [complex]. However, **MultiProp** [complex] has one additional subelement, **sEvalMultiPropList** [string], in comparison with **SingleProp** [complex]. **sEvalMultiPropList** [string] identifies the properties involved in the multiple property critical evaluation. For example, experimental saturated vapor pressure data are often evaluated together with calorimetric enthalpy-of-vaporization data, experimental heat capacity data in the liquid state, and heat capacity data in gas phase (commonly calculated by the method of statistical mechanics). In this case, the list of properties should be provided in the element **sEvalMultiPropList** [string].

Finally, **EquationOfState** [complex] (Fig. 9) is designed to allow identification of a particular equation of state used to enforce general thermodynamic consistency. The elements **sEvalEOSDescription** [string] and **EvalEOSRef** [complex] are completely analogous to elements in **SingleProp** [complex] and **MultiProp** [complex]. **EquationOfState** [complex] also includes the element **sEvalEOSName** [string] to identify the name of the equation of state used (if applicable).

The list of options for **ePropertyName** [enumeration] and **eMethodName** [enumeration] for each **PropertyGroup** [complex] is provided below together with the units for each property.

Criticals [complex]

ePropertyName [enumeration] (critical temperature, K; critical pressure, kPa; critical density, $kg \cdot m^{-3}$; critical molar volume, $m^3 \cdot mol^{-1}$; critical specific volume, $m^3 \cdot kg^{-1}$; critical compressibility; lower consolute temperature, K; upper consolute temperature, K).

eMethodName [enumeration] (visual observation in an unstirred cell, visual observation in a stirred cell, DSC/DTA, derived from PVT data, extrapolated vapor pressure, rectilinear diameter, appearance of two phases, disappearance of two phases, direct measurement, other).

VaporPBoilingTAzeotropTandP [complex]

ePropertyName [enumeration] (vapor of sublimation pressure, kPa; normal boiling temperature, K; boiling temperature at pressure P, K; azeotropic pressure, kPa; azeotropic temperature, K)

eMethodName [enumeration] (manometric method, closed cell – static method, diaphragm manometer, inclined piston gauge, isochoric PVT apparatus, isoteniscope, Knudsen effusion method, distillation, ebulliometric method – recirculating still, twin ebulliometer, transpiration method, rate of evaporation, azeotropic temperature or pressure determination when X = Y, azeotropic temperature or pressure determination by temperature or pressure extreme).

PhaseTransition [complex]

ePropertyName [enumeration] (triple point temperature, K; triple point pressure, kPa; normal melting temperature, K; enthalpy of transition or fusion, $kJ \cdot mol^{-1}$; cryoscopic constant, K^{-1}; enthalpy of vaporization or sublimation, $kJ \cdot mol^{-1}$; quadruple (quintuple) point temperature, K; quadruple (quintuple) point pressure, kPa; solid-liquid equilibrium temperature, K; liquid-liquid equilibrium temperature, K; eutectic temperature, K; Lattice energy at 0 K, $kJ \cdot mol^{-1}$).

eMethodName [enumeration] (visual observation, heating/cooling curves, DSC/DTA, adiabatic calorimetry, large-sample thermal analysis, drop calorimetry, drop ice or diphenyl ether calorimetry, obtained from cryoscopic constant, derived from phase diagram analysis, static calorimetry, flow calorimetry, derived by the Second Law, depression of a freezing point of a dilute solution, other).

CompositionAtPhaseEquilibrium [complex]

ePropertyName [enumeration] (azeotropic composition- mole fraction; azeotropic composition- mass fraction; eutectic composition- mole fraction; eutectic composition- mass fraction; eutectic composition- volume fraction; lower consolute composition- volume fraction; lower consolute composition- mole fraction; lower consolute

composition- mass fraction; mass per volume of solution, $kg \cdot m^{-3}$; mass ratio to solvent; molality, $mol \cdot kg^{-1}$; molarity, $mol \cdot dm^{-3}$; mole fraction; mole fraction in LLG critical state; mole ratio to solvent; moles per mass of solution, mol/kg; upper consolute composition- volume fraction; upper consolute composition- mole fraction; upper consolute composition- mass fraction; volume fraction; volume ratio to solvent; mass fraction; mass fraction in LLG critical state; Henry's Law constant for mole fraction, kPa; Henry's Law constant (molality), $kPa \cdot kg \cdot mol^{-1}$; Henry's Law constant (molarity), $kPa \cdot l \cdot mol^{-1}$; Bunsen coefficient; Oswald coefficient; partial pressure, kPa).

eMethodName [enumeration] (azeotropic composition determination when X = Y, azeotropic composition determination by temperature of pressure extreme, chromatography, spectrophotometry, determined by refractive index and/or density, calculated by Gibbs-Duhem equation, titration method, static method, dynamic method, phase equilibrium, derived from phase diagram analysis, appearance of two phases, disappearance of two phases, photoacoustic method, other).

ActivityFugacityOsmoticProp [complex]

ePropertyName [enumeration] (activity; activity coefficient; fugacity, kPa; fugacity coefficient; osmotic pressure, kPa; osmotic coefficient).

eMethodName [enumeration] (chromatography, spectroscopy, mass-spectrometry, NMR-spectrometry, static method, isopiestic method, other).

VolumetricProp [complex]

ePropertyName [enumeration] (specific density, $kg \cdot m^{-3}$; specific volume, $m^3 \cdot kg^{-1}$; molar density, $mol \cdot m^{-3}$; molar volume, $m^3 \cdot mol^{-1}$; second virial coefficient, $m^3 \cdot mol^{-1}$; second acoustic virial coefficient, $m^3 \cdot mol^{-1}$; third virial coefficient, $m^6 \cdot mol^{-2}$; third acoustic virial coefficient, $m^6 \cdot mol^{-2}$; third interaction virial coefficient C_{112}, $m^6 \cdot mol^{-2}$; third interaction virial coefficient C_{122}, $m^6 \cdot mol^{-2}$; excess virial coefficient, $m^3 \cdot mol^{-1}$; interaction virial coefficient, m^3/mol^{-1}; excess volume, $m^3 \cdot mol^{-1}$; partial molar volume, $m^3 \cdot mol^{-1}$; relative partial molar volume, $m^3 \cdot mol^{-1}$; apparent molar volume, $m^3 \cdot mol^{-1}$; adiabatic compressibility, kPa^{-1}; isothermal compressibility, kPa^{-1}; coefficient of expansion, K^{-1}; compressibility factor, thermal pressure coefficient, $kPa \cdot K^{-1}$).

eMethodName [enumeration] (pycnometric method, buoyancy method, vibrating tube method, isochoric *PVT* measurement, other *PVT* measurement, Burnett expansion

technique, constant-volume piesometry, direct dilatometry, derived analytically, derived graphically, calculated with densities of this investigation, calculated with a solvent density reported elsewhere, other).

HeatCapacityAndDerivedProp [complex]

ePropertyName [enumeration] (heat capacity at constant pressure, $J \cdot K^{-1} \cdot mol^{-1}$; heat capacity at vapor saturation pressure, $J \cdot K^{-1} \cdot mol^{-1}$; heat capacity at constant volume, $J \cdot K^{-1} \cdot mol^{-1}$; heat capacity at constant pressure per unit mass, $J \cdot K^{-1} \cdot kg^{-1}$; heat capacity at constant pressure per unit volume, $J \cdot K^{-1} \cdot m^{-3}$; heat capacity ratio C_p/C_v, standard entropy, *S(T)-S(0)*, $J \cdot K^{-1} \cdot mol^{-1}$; standard enthalpy, *H(T)-H(0)*, $kJ \cdot mol^{-1}$; enthalpy function, *{H(T)-H(0)}/T*, $J \cdot K^{-1} \cdot mol^{-1}$; Gibbs energy function, {*G(T)-H(0)}/T*, $J \cdot K^{-1} \cdot mol^{-1}$; Gibbs energy, *G(T)-H(0)*, $kJ \cdot mol^{-1}$; Helmholtz energy, *A(T)-E(0)*, $kJ \cdot mol^{-1}$; internal energy, *E(T)-E(0)*, $kJ \cdot mol^{-1}$; Joule-Thompson coefficient, $K \cdot kPa^{-1}$; pressure coefficient of enthalpy, $J \cdot mol^{-1} \cdot kPa^{-1}$).

eMethodName [enumeration] (vacuum adiabatic calorimetry, small sample (less than 1g) adiabatic calorimetry, flow calorimetry, large sample (1g) DSC, small sample (50mg) DSC, drop calorimetry, drop ice or diphenyl ether calorimetry, open cup calorimetry, closed cup calorimetry, differential flow calorimetry, extra sensitive DSC, twin closed cell calorimetry, derived from speed of sound, derived from equation of state, expansion technique, other).

ExcessPartialApparentEnergyProp [complex]

ePropertyName [enumeration] (apparent enthalpy, $kJ \cdot mol^{-1}$; apparent entropy, $J \cdot mol^{-1}K^{-1}$; apparent Gibbs energy, $kJ \cdot mol^{-1}$; apparent molar heat capacity, $J \cdot mol^{-1}K^{-1}$; enthalpy of mixing with binary solvent, $kJ \cdot mol^{-1}$; excess enthalpy [enthalpy of mixing], $kJ \cdot mol^{-1}$; excess entropy, $J \cdot mol^{-1}K^{-1}$; excess Gibbs energy, $kJ \cdot mol^{-1}$; excess heat capacity, $J \cdot mol^{-1}K^{-1}$; partial molar enthalpy, $J \cdot mol^{-1}$; partial molar entropy, $J \cdot mol^{-1}K^{-1}$; partial molar Gibbs energy, $kJ \cdot mol^{-1}$; partial molar heat capacity, $J \cdot mol^{-1}K^{-1}$; relative partial molar enthalpy, $kJ \cdot mol^{-1}$; relative partial entropy, $J \cdot mol^{-1}K^{-1}$; relative partial molar Gibbs energy, $kJ \cdot mol^{-1}$; relative partial molar heat capacity, $J \cdot mol^{-1}K^{-1}$; standard state enthalpy, $kJ \cdot mol^{-1}$; standard state entropy, $J \cdot mol^{-1}K^{-1}$; standard state Gibbs energy, $kJ \cdot mol^{-1}$; standard state heat capacity, $J \cdot mol^{-1}K^{-1}$).

eMethodName [enumeration] (vacuum adiabatic calorimetry, small (less than 1 g) adiabatic calorimetry, flow calorimetry, differential flow calorimetry, Calvet calorimetry, large sample (1 g) DSC, small sample (50 mg) DSC, extra sensitive DSC, twin closed calorimetry, other).

TransportProp [complex]

ePropertyName [enumeration] (viscosity, Pa·s; kinematic viscosity, $m^2 \cdot s^{-1}$; fluidity, $Pa^{-1} \cdot s^{-1}$; thermal conductivity, $W \cdot m^{-1} \cdot K^{-1}$; thermal diffusivity, $m^2 \cdot s^{-1}$; binary diffusion coefficient; $10^{-9} \cdot m^2 \cdot s^{-1}$; self diffusion coefficient, $10^{-9} \cdot m^2 \cdot s^{-1}$, tracer diffusion coefficient, $10^{-9} \cdot m^2 \cdot s^{-1}$).

eMethodName [enumeration] (capillary tube [Oswald, Ubellohde] method, cone and plate viscometry, concentric cylinders viscometry, falling or rolling sphere viscometry, oscillating disk viscometry, vibrating wire viscometry, parallel plate method, coaxial cylinder method, hot wire method, optical interferometry, dispersion, diaphragm cell, open capillary, closed capillary, Taylor dispersion method, NMR spin-echo technique, other).

RefractionSurfaceTensionSoundSpeed [complex]

ePropertyName [enumeration] (refractive index [*Na* D-line]; refractive index [other wavelength]; relative permittivity at zero frequency; relative permittivity at various frequencies; electrical conductivity, $S \cdot m^{-1}$; surface tension liquid-gas, $N \cdot m^{-1}$; interfacial tension, $N \cdot m^{-1}$; speed of sound, $m \cdot s^{-1}$).

eMethodName [enumeration] (standard abbe refractometry, precision abbe refractometry, dipping refractometry (monochromatic), interferometer, capillary rise, drop weight, drop volume, maximal bubble pressure, pendant drop shape, ring tensiometer, linear variable-path acoustic interferometer, sing-around technique in a fixed-path interferometer, annular interferometer, pulse-echo method, spherical resonator, light diffraction method, other).

The metadata elements **Constraint** [complex] and **Variable** [complex] have very similar structures, as shown in Fig. 10. **Constraint** [complex] has nine subelements: **nConstraintNumber** [numerical, integer] used to ease equation representation, **ConstraintID** [complex], **ConstraintPhaseID** [complex], **Solvent** [complex], **nConstraintValue** [numerical, floating], **nConstrDigits** [numerical, integer], and three

elements associated with specification of uncertainty: **ConstrUncertainty** [complex], **ConstrRepeatability** [complex], and **ConstrDeviceSpec** [complex]. All elements associated with uncertainty are described later in this document in the section titled *Representation of Uncertainties*. **ConstraintID** consists of **RegNum** [complex] and **ConstraintType** [complex]. **ConstraintType** [complex] includes five types of constraints: **eTemperature** [enumeration]**, ePressure** [enumeration], **eComponentComposition** [enumeration], **eSolventComposition** [enumeration], and **eMiscellaneous** [enumeration]. **RegNum** [complex] should be used for mixtures only if the constraint is a composition expressed in terms of the concentration of a particular compound.

The values of the enumerated elements for **eTemperature** [enumeration] and **ePressure** [enumeration] are (temperature, K; upper temperature, K; lower temperature, K) and (pressure, kPa; upper pressure, kPa; lower pressure, kPa), respectively. The 'upper' and 'lower' values are required in the specification of enthalpy-increment metadata. The element **eComponentComposition** [enumeration, always defined with **RegNum**] lists (mole fraction; mass fraction; molality, $mol \cdot kg^{-1}$; molarity, $mol \cdot dm^{-3}$; volume fraction; moles per mass of solution, $mol \cdot kg^{-1}$; mass per volume of solution, $kg \cdot m^{-3}$; mole ratio to solvent; mass ratio to solvent; volume ratio to solvent; activity; and activity coefficient. The element **eSolventComposition** [enumeration, always defined with **RegNum**] lists (solvent- mole fraction; solvent- mass fraction; solvent-molality, $mol \cdot kg^{-1}$; solvent- molarity, $mol \cdot dm^{-3}$; solvent- volume fraction; solvent- mole ratio to other component of a binary solvent; solvent- mass ratio to other component of a binary solvent; solvent- volume ratio to other component of a binary solvent). The element **eMiscellaneous** [enumeration] identifies various other types of constraints and includes the following enumerations (wavelength, nm; molar volume, $m^3 \cdot mol^{-1}$; specific volume, $m^3 \cdot kg^{-1}$; density, $kg \cdot m^{-3}$; molar density, $mol \cdot m^{-3}$; entropy, $J \cdot K^{-1} \cdot mol^{-1}$).

The structure of the element **ConstraintPhaseID** [complex] (Fig. 10) is analogous to the structure of **PhaseID** [complex] (Fig. 7) and **RefPhaseID** [complex] (Fig. 8). **Solvent** [complex] is identified through specification **RegNum** [complex] for each solvent component. **nConstraintValue** [numerical, floating] stores the numerical value of the

constraint with **nConstrDigits** [numerical, integer] representing the number of digits in the value.

The structure of the element **Variable** [complex] (Fig. 10) is analogous to that of **Constraint** [complex]; however, **Variable** [complex] does not include elements corresponding to **nConstraintValue** [numerical, floating] and **nConstrDigits** [numerical, integer]and includes the additional element **nVarNumber** [numerical, integer]. **nVarNumber** [numerical, integer] designates the sequential variable number for the list of variables. This ensures correct association of numerical values with variables.

The schema element **NumValues** [complex] (Fig. 7) consists of **VariableValue** [complex], which represents numerical values of variables, and **PropertyValue** [complex], which represents numerical values of properties. Each contains a sequential identifier for the variable or property: **nVarNumber** [numerical, integer] for **VariableValue** [complex] and **nPropNumber** [numerical, integer] for **PropertyValue** [complex].

The element **VariableValue** [complex] (Fig. 7) has six subelements, including the sequential identifier **nVarNumber** [numerical, integer], the numerical value of the variable **nVarValue** [numerical, floating] and the associated number of digits **nVarDigits** [numerical, integer]. The remaining subelements are associated with the representation of uncertainty and are described later: **VarUncertainty** [complex], **VarRepeatability** [complex], and **nVarDeviceSpecValue** [numerical, floating].

The element **PropertyValue** [complex] (Fig. 7) has seven subelements, including the sequential identifier **nPropNumber** [numerical, integer], and five subelements associated with representation of uncertainty that will be described in later: **Combined Uncertainty** [complex], **PropUncertainty** [complex], **PropRepeatability** [complex], **nPropDeviceSpecValue** [numerical, floating], and **CurveDev** [complex]. The “switch” symbol as a subelement for **PropertyValue** [complex] allows representation of the property as a particular value or as an upper or lower limit. Particular values are stored in **nPropValue** [numerical, floating] with a specified number of digits **nPropDigits** [numerical, integer]. Property limits are stored in **PropLimit** [complex] with subelements **nPropUpperLimitValue** [numerical, floating] or **nPropLowerLimitValue** [numerical,

floating] and the number of numerical digits in the property value **nPropLimitDigits** [numerical, integer].

***4. ReactionData* Block.** The *ReactionData* block is for storage of data for chemical reactions, and is shown in Fig. 11. This block includes a number of elements **sExpPurpose** [string], **sCompiler** [string], **sContributor** [string], **dateDateAdded** [date], and **NumValues** [complex] that are identical to those used in the *PureOrMixtureData* block, and were described earlier. The element **Participant** [complex] (Fig. 12) stores information about a participant in a chemical reaction. This element includes **RegNum** [complex], **nSampleNum** [numerical, integer], **nStoichiometricCoef** [numerical, floating] to store stoichiometric coefficients (negative values for reactants and positive values for products), **ePhase** [enumeration] and **eCrystalLatticeType** [enumeration] with values the same as in the **PhaseID** [complex] (Fig. 7) element of the *PureOrMixtureData* block, **eCompositionRepresentation** [enumeration], and **nNumericalComposition** [numerical, floating]. **Variable** [complex] and **Constraint** [complex] differ from those in the *PureOrMixtureData* block by the absence of the **Solvent** [complex] and **PhaseID** [complex] elements (Fig. 13).

Within the **Participant** [complex] (Fig. 12) of the *ReactionData* block **eCompositionRepresentation** [enumeration] stores the composition representation for a participant (mole ratio of solvent to participant; molality - moles of participant per kilogram of solvent, $mol \cdot kg^{-1}$; moles of participant per kilogram of solution, $mol \cdot kg^{-1}$; molarity - moles of participant per liter of solution; mole ratio - mole of participant per mole of solvent; mass ratio - mass of participant per mass of solvent; volume ratio - volume of participant per volume of solvent; mass of participant per volume of solution, $kg \cdot m^{-3}$). **nNumericalComposition** [numerical, floating] (Fig. 12) indicates the numerical value of the composition representation. **ECompositionRepresentation** [enumeration] and **nNumericalComposition** [numerical, floating] are used for change-of-state reactions only.

The element **eReactionType** [enumeration] (Fig. 11) stores a description of the general type of chemical reaction. The complete enumeration list includes the following: (combustion with oxygen, addition of various compounds to unsaturated compounds, addition of water to a liquid or solid to produce a hydrate, atomization (or formation from

atoms), combustion with other elements or compounds, esterification, exchange of alkyl groups, exchange of hydrogen (atoms) with other groups, formation of a compound from elements in their stable state, halogenation (addition of or replacement by a halogen), hydrogenation (addition of H_2 to unsaturated compounds), hydrohalogenation, hydrolysis of ions, other reactions with water, ion exchange, neutralization (reaction of an acid with a base), oxidation with oxidizing agents other than oxygen, oxidation with oxygen (not complete), polymerization (all other types), homonuclear dimerization, solvolyis (solvents other than water), stereoisomerism, structural isomerization, other reactions).

The element **Property** [complex] (Fig. 14) is similar in structure to that used in the *PureOrMixtureData.* However, instead of the 10 property groups used in the *PureOrMixtureData* structure, the **Property** [complex] block has only two: **ReactionStateChangeProp** [complex], for representation of thermochemical properties for change-of-state reactions such as combustion with oxygen, and **ReactionEquilibriumProp** [complex] for properties of reactions in equilibrium. Both property groups are characterized with **ePropName** [enumeration] for identification of properties and **eMethodName** [enumeration], which specifies the experimental methods used. Analogous to the *PureOrMixtureData* block, the **Property-Method** [complex] subelement of **Property** [complex] (Fig. 14) includes **Prediction** [complex] and **CriticalEvaluation** [complex] subelements. Subelements of **Prediction** [complex] and **CriticalEvaluation** [complex] are shown in Fig. 9, and were described earlier.

ReactionStateChangeProp *[complex]*

The element **ePropName** [enumeration] includes the following properties: (enthalpy of reaction, $kJ \cdot mol^{-1}$; internal energy, $J \cdot g^{-1}$; internal energy of reaction - mole basis, $kJ \cdot mol^{-1}$; Gibbs energy of reaction, $kJ \cdot mol^{-1}$; entropy of reaction, $J \cdot K^{-1} mol^{-1}$).

eMethodName [enumeration] includes the following experimental methods: (static bomb calorimetry, rotating bomb calorimetry, micro-bomb calorimetry, flame calorimetry, e.m.f. measurement, other).

ReactionEquilibriumProp *[complex]*

ePropName [enumeration] includes the following properties: (thermodynamic equilibrium constant; apparent equilibrium constant in terms of molality, $(mol \cdot kg^{-1})^n$;

apparent equilibrium constant in terms of molarity, $(mol \cdot dm^{-3})^n$; apparent equilibrium constant, in terms of partial pressure, $(kPa)^n$; apparent constant in terms of mole fraction).

eMethodName [enumeration] includes the following experimental methods: (static equilibration, dynamic equilibration, chromatography, IR spectrometry, UV spectroscopy, NMR spectrometry, titration, other).

Solvent [complex] and **Catalyst** [complex] (Fig. 14) have essentially identical structures both characterized with **ePhase** [enumeration] and **RegNum** [complex], as described earlier. The list of options for **eStandardState** [enumeration] was given earlier. The temperature and pressure associated with the reaction property are stored in **nTemperature-K** [numerical, floating] and **nPressure-kPa** [numerical, floating] together with their respective numbers of digits: **nTemperatureDigits** [numerical, integer] and **nPressureDigits** [numerical, integer]. The remaining five elements within the **Property** [complex] element of the *ReactionData* block are associated with representation of uncertainties: **CombinedUncertainty** [complex], **PropUncertainty** [complex], **PropRepeatability** [complex], **PropDeviceSpec** [complex], and **CurveDev** [complex].

Representation of Uncertainties

Basic Principles and Definitions. The expression of uncertainty requires clear definition of a variety of quantities and terms. Definitions and descriptions of all quantities related to the expression of uncertainty in ThermoML conform to the *Guide to the Expression of Uncertainty in Measurement*, ISO (International Organization for Standardization), October, 1993.[24] These ISO recommendations were adopted with minor editorial changes as the *U. S. Guide to the Expression of Uncertainty in Measurement*.[25] Reference 24 is commonly referred to by its abbreviation; the *GUM*. Reference 25 is assumed equivalent to reference 24 and is referred to as the *Guide* in this paper. The historical development of these recommendations beginning in 1977 is described in the *Guide*. The recommendations have been summarized in *Guidelines for the Evaluation and Expression of Uncertainty in NIST Measurement Results*,[26] which is available by free download from the Internet (http://physics.nist.gov/cuu/).

In the second article [15] in the series [14-16] describing the development of ThermoML, application of the recommendations of the *Guide* to particular aspects of

experimental thermodynamic-property data were discussed, and additions to the ThermoML schema for representation of uncertainties were described. In the present paper, the basic principles are reviewed briefly, but the reader is referred to the earlier papers [15, 24-26] for comprehensive discussions of the internationally accepted quantities and terms for describing uncertainty, including *standard uncertainty*, *combined standard uncertainty*, *coverage factor*, *expanded uncertainty*, *combined expanded uncertainty*, and *level of confidence*. Table 1 shows the general mathematical relationships between the quantities used for the expression of uncertainty, which are used explicitly in ThermoML. Uncertainties are represented for *variables*, *constraints*, and *properties*.

As noted in the footnote of Table 1, the standard uncertainties associated with state functions are defined to be independent, and do not include uncertainty components associated with propagation of uncertainty from one state function to another. For example, if density of a single-component gas (the property) is reported as a function of temperature and pressure (the variables), it is important to avoid including the effect of uncertainty in temperature upon the uncertainty in the pressure. This is to avoid overestimation (or 'double counting') of uncertainties, when they are propagated to the designated property in a subsequent step.

In Table 1, the standard uncertainty is listed independently for variables, constraints, *and* properties. The appropriateness of this type of reporting is shown using typical results for vapor-liquid equilibrium (*VLE*) experiments. In the reporting of *VLE* results, pressures p, temperatures T, and phase-compositions x (liquid) and y (vapor) are commonly reported. Uncertainties associated with each quantity (p, T, x, and y) are often reported independently. Furthermore, the 'property' is often not specified explicitly because the designation is arbitrary. To accommodate this type of reporting, ThermoML includes representation of standard uncertainties for the variables, constraints, *and* the property. A second uncertainty (the *combined* uncertainty) is defined only for the property and includes propagation of uncertainties from the variables and constraints to the property. The combined uncertainty is represented separately, as shown in the bottom of Table 1. In a broader sense, the standard uncertainties u_x could be considered "*combined*" in that they combine uncertainties from various sources. In ThermoML;

however, the term *combined standard uncertainty* is reserved for uncertainties derived by propagation of uncertainties from variables and constraints to those for the designated property.

Recently, practices in the expression of uncertainty in the experimental literature for thermodynamic property measurements were reviewed with determinations of critical temperature T_c for pure compounds used as a case study [27]. In that article it was shown that although gradual and continuous progress has been made in the reporting of uncertainty information, comprehensive uncertainty analyses remain rare, particularly with regard to consideration of contributions arising from sample impurities. Examples were provided of dramatic underreporting of uncertainty magnitudes due to failure to consider this important component. In the time period since 1990, approximately 42 percent of the articles reporting experimental T_c values listed only some type of precision information rather than a comprehensive combined uncertainty. This limited information provides only a lower bound for the combined uncertainty, and is of low value to data evaluators and application engineers. Nonetheless, as this is often the only type of information related to uncertainties available from many reports, it was decided to represent this type of information in ThermoML.

The following series of definitions describe terms that are commonly reported in the literature as repeatabilities, deviations from a fitted curve, or device specifications. These uncertainty-assessment components are referred to collectively here as *precisions*. These represent components of an uncertainty assessment, but do not meet the criteria for the *uncertainty of measurement*, which includes all sources of uncertainty. It was decided to include some precisions in ThermoML because certain of these quantities can be well defined and may be useful to data evaluators in subsequent assessments. The *International Vocabulary of Basic and General Terms in Metrology* [28] (commonly abbreviated VIM) does not give a definition for precision because of the many definitions that exist for this word. This is consistent with the usage here.

Definitions for *repeatability* [15,28], *curve deviation* [15], and *device specification* [15] were given previously, and are the only representations of precision included in ThermoML. It must be recognized clearly by the user of this information that these are components in an array of information that can be used in estimating the *uncertainty of*

measurement. The terms *precision* and *accuracy* continue to be widely used in the scientific literature, even though neither has a quantitative meaning as noted in the following quotes from authoritative sources.

> ***"Precision***: As noted earlier, this term has numerous conflicting meanings, and is not expressed quantitatively in the authoritative literature." [25]

This term is not represented in ThermoML.

> ***"Accuracy (of Measurement)*** [VIM 3.5]: Closeness of agreement between the result of a measurement and the measurand."[25] "Accuracy is a qualitative concept." [25]

Accuracy cannot be represented quantitatively and is not represented in ThermoML.

Implementation of Uncertainty Definitions in ThermoML

The *Guide* [25] provides specific recommendations for the reporting of uncertainties (Chapter 7: Reporting Uncertainty), which are accommodated fully in ThermoML. Specifically, section 7.2.3 of the *Guide* lists recommendations for reporting of the *expanded uncertainty U*. The recommendations are:

1. Give a full description of how the *measurand Y* is defined.
2. State the result of the measurement as $Y = y \pm U$ and give the units of y and U.
3. Include the *relative expanded uncertainty* $U / |y|$, $|y| \neq 0$, when appropriate.
4. Give the value of k used to obtain U [or, for the convenience of the user, give both k and u].
5. Give the approximate *level of confidence* associated with the interval $y \pm U$ and state how it was determined.
6. Give the information outlined in 7.2.7 of the *Guide* or refer to a published document that contains it. (Note: Section 7.2.7 makes specific recommendations related to reporting of the origins of all uncertainty estimates in the document text.)

Recommendation 1 is implemented in ThermoML through the complete definition of variables, constraints, and properties based on the established laws of phenomenological thermodynamics. Recommendations 2, 4, and 5 are addressed explicitly in ThermoML. Although not included explicitly, the *relative expanded uncertainty* (recommendation 3) can be derived simply from the other values. Recommendation 6 involves detailed reporting suggestions that would be impractical to implement fully in ThermoML. These

include listing the source of all uncertainty estimates used to estimate any standard uncertainty and providing partial derivatives or sensitivity coefficients related to key uncertainty components. Nonetheless, in all cases a text schema element is provided, which can be used for descriptions of the uncertainty estimation methods.

General Structure of Schema Elements for Expression of Uncertainties and Precisions

Schema elements for expressions of uncertainty are summarized in Table 2 and those for expressions of precision are in Table 3. This separation is done to emphasize the conceptual difference between these quantities. Column 1 in Tables 2 and 3 indicates the element name used in ThermoML. Column 2 clarifies the meaning of the abbreviations used in column 1. Column 3 specifies the type of measurand to which the names apply. Columns 4 and 5 specify the general location of the element in the ThermoML schema in terms of whether the element is associated with a *data set* or with each *data point*. In the following sections, each element is defined and some specific guidance related to its usage is given.

The elements listed in Table 2 are separated into those for representation of defined *uncertainties*, which are given for variables, constraints, and properties, and those for *combined uncertainties*, which are given for properties only. Elements for representation of the defined *uncertainties* are described first.

ThermoML elements for the expression of uncertainties

***Uncertainty** [complex]*

This complex element includes the subelements associated with expression of the *expanded uncertainty* values U_x for variables, constraints, and properties. The symbol * indicates that this element is present in the schema for variables (***** = **Var**), constraints (***** = **Constr**), and properties (***** = **Prop**). The subelements of ***Uncertainty** [complex] are described in the following paragraphs.

nUncertAssessNum [numerical, integer]; the *uncertainty assessment number* is an integer used to identity a particular assessment of the uncertainty. ThermoML can accommodate multiple uncertainty assessments for the same data. For variables and properties, this number also serves to link uncertainty elements associated with individual

data points (e.g., the *expanded uncertainty value*) with an element associated with the entire data set (e.g., the *coverage factor*). The *uncertainty assessment number* for a given assessment is associated with all of the subelements within ***Uncertainty** [complex], as listed in the upper section of Table 2.

sUncertEvaluator [string]; the *uncertainty evaluator* string is used to identify the individual or institution responsible for the assessment. For information reported explicitly in journal articles, the evaluator is the author(s). Multiple evaluations can be stored simultaneously with ThermoML.

sUncertEvalMethod [string]; the *uncertainty evaluation method* element allows storage of descriptive information related to the uncertainty assessment, such as the sources of key information. In this way, this element can be used to accommodate item number 6 of the recommendations for the reporting of uncertainties given in the *Guide* (*Guide* section 7.2.3).

nStdUncertValue [numerical, floating] is for storage of the numerical *standard uncertainty* value u_x and shown in column 3 of Table 1. By definition, the value u_x represents one standard deviation.

AsymStdUncertainty [complex] is for storage of the *standard uncertainty* for an uncertainty that is asymmetrical about the property value. This element is included only for properties and not for variables, constraints, or defined precisions. A symmetrical uncertainty can by represented as $y_p \pm u_p$, where y_p is the measured value and u_p is the *standard uncertainty*. The values spanned by this range lie between ($y_p + u_p$) and ($y_p - u_p$). In contrast, the values spanned by an unsymmetrical range lie between ($y_p + u_{p+}$) and ($y_p - u_{p-}$), where $u_{p+} \neq u_{p-}$. Subelements of **AsymStdUncertainty** [complex] are **nPositiveValue** [numerical, floating] for storage of u_{p+} and **nNegativeValue** [numerical, floating] for storage of u_{p-}.

nCoverageFactor [numerical, floating] is used for storage of the *coverage factor* k_x shown in column 4 of Table 1.

nExpandUncertValue [numerical, floating] is used for storage of the *expanded uncertainty value* U_x shown in column 5 of Table 1. It is recognized that simultaneous storage of **nStdUncertValue, nCoverageFactor,** and **nExpandUncertValue** is

redundant because of the simple relationship $u_x \cdot k_x = U_x$. It is a recommendation of the *Guide* that all three be given to avoid any ambiguity.

AsymExpandUncertainty [complex] is for storage of the *expanded uncertainty* for an uncertainty that is asymmetrical about the property value. This element is included only for properties. The values spanned by the unsymmetrical range lie between (y_p + U_{P+}) and (y_p – U_{P-}), where $U_{P+} \neq U_{P-}$ Subelements of **AsymExpandUncertainty** [complex] are **nPositiveValue** [numerical, floating] for storage of U_{P+} and **nNegativeValue** [numerical, floating] for storage of U_{P-}.

nUncertLevOfConfid [numerical, floating] is used for storage of the *Level of Confidence* L_x associated with U_x. The level of confidence is always expressed in ThermoML as a percentage.

CombinedUncertainty *[complex]*

This complex element includes the subelements associated with expression of the *combined expanded uncertainty* values U_{comb}. The *combined expanded uncertainty* is stored for properties only, and includes propagation of uncertainties in the variables and constraints to that of the property, as discussed earlier, and as indicated in the lower section of Table 1. Nearly all subelements are analogous to those for **Uncertainty** [complex]. One additional element provides an enumeration list for description of the method of assessment.

nCombUncertAssessNum [numerical, integer]; the *combined uncertainty assessment number* is an integer used to identity a particular assessment of the combined uncertainty. Its use is as described for **nUncertAssessNum** [numerical, integer].

sCombUncertEvaluator [string]; the *combined uncertainty evaluator* string is for identification of the individual or institution responsible for the assessment of the *combined uncertainty*.

eCombUncertEvalMethod [enumeration] provides an enumeration list for specification of the general method for evaluation of the *combined uncertainty*. The enumerations are: (propagation of estimated standard uncertainties, comparison with reference property values). In the field of experimental thermodynamics, it is common to test the accuracy of an apparatus through measurements performed on reference materials with property values having well-established uncertainties. This approach is an

alternative to that of determining uncertainties for all possible components and propagating these to the uncertainty in the property. The source of the reference property values should be provided in the corresponding string element, which follows.

sCombUncertEvalMethod [string]; the *combined uncertainty evaluation method* element allows storage of descriptive information related to the uncertainty assessment. This element can be used to accommodate item number 6 of the recommendations for the reporting of uncertainties given in the *Guide* [25]. In addition, this element can be used for storage of information about the source of reference property values, if *comparison with reference property values* is chosen in **eCombUncertEvalMethod** [enumeration].

nCombStdUncertValue [numerical, floating] is used for storage of the combined standard uncertainty value u_{comb} shown in column 3 of the lower section of Table 1. The value u_{comb} represents one standard deviation by definition.

AsymCombStdUncertainty [complex] is for storage of the *combined standard uncertainty* for an uncertainty that is asymmetrical about the property value. This element is included only for properties. The values spanned by the unsymmetrical range lie between $(y_p + u_{comb+})$ and $(y_p - u_{comb-})$, where $u_{comb+} \neq u_{comb-}$. Subelements of **AsymCombStdUncertainty** [complex] are **nPositiveValue** [numerical, floating] for storage of u_{comb+} and **nNegativeValue** [numerical, floating] for storage of u_{comb-}.

nCombCoverageFactor [numerical, floating] stores the *coverage factor* k_{comb} used to obtain $U_{comb} = k_{comb} \cdot u_{comb}$.

nCombExpandUncertValue [numerical, floating] stores the *combined expanded uncertainty* value U_{comb}.

AsymCombExpandUncertainty [complex] is for storage of the *combined expanded uncertainty* for an uncertainty that is asymmetrical about the property value. This element is included only for properties. The values spanned by the unsymmetrical range lie between $(y_p + U_{comb+})$ and $(y_p - U_{comb-})$, where $U_{comb+} \neq U_{comb-}$. Subelements of **AsymCombExpandUncertainty** [complex] are **nPositiveValue** [numerical, floating] for storage of U_{comb+} and **nNegativeValue** [numerical, floating] for storage of U_{comb-}.

nCombUncertLevOfConfid [numerical, floating] stores the *Level of Confidence* L_{comb} associated with U_{comb}. The level of confidence is stored as a percentage.

PropLimit *[complex]*

This complex element is a subelement of **PropertyValue** [complex] and is for storage of property values reported as upper or lower limits for the measurand. This type of element is included for properties only and is not included for variables, constraints, or precisions. The subelements of **PropLimit** [complex] are **nPropUpperLimitValue** [numerical, floating] and **nPropLowerLimitValue** [numerical, floating] for storage of either numerical limiting value, and **nPropLimitDigits** [numerical, integer] for storage of the number of digits in the value.

ThermoML elements for the expression of precisions

The elements listed in Table 3 are for the expression of precisions. The measurand types to which these may be associated are listed in column three of Table 3. These quantities are completely independent of those specified in Table 2. The term *combined* is not used in these elements. This term is applicable only to uncertainties, which include contributions from *all* sources. *Expanded* precisions are also not included in ThermoML. All of the following elements are optional in the schema, as is true for all elements associated with the specification of uncertainty. The general locations of the elements listed in Table 3 are indicated in columns 4 and 5. Detailed locations are provided later in this paper.

****Repeatability*** [complex]

The numerical quantity, *repeatability*, was defined earlier. The symbol * indicates that this element is present in the schema for variables (***** = **Var**), constraints (***** = **Constr**), and properties (***** = **Prop**). The following elements are used for specification of this quantity for variables, constraints, and properties.

sRepeatEvaluator [string]; the *repeatability evaluator* string is used to identify the individual or institution responsible for the assessment of the repeatability. In most cases, this will be the author of the original publication.

eRepeatMethod [enumeration] provides an enumeration list for specification of the statistical definition of the *repeatability value*. The four enumerations are: (Standard deviation of a single value (biased), Standard deviation of a single value (unbiased), Standard deviation of the mean, and Other). These terms are defined in most common texts in the field of statistics (*cf.* reference 29). Selection of the enumeration ‘Other,’

should be accompanied by a description of the method in the string element **sRepeatMethod** [string].

Three representations of repeatability are included in ThermoML. The standard deviation of a single value (unbiased) $\sigma_{unbiased}$, the standard deviation of a single value (biased) σ_{biased}, and the standard deviation of the mean σ_{mean}. Mathematical definitions were given previously [15].A discussion of the application of these formulae to particular experimental conditions is beyond the scope of this paper. The reader is referred to any common text in statistics for this information. The standard deviation of the mean is often applied in the analysis of results obtained with combustion bomb calorimetry [**Error! Bookmark not defined.**].

sRepeatMethod [string]; the *repeatability assessment method* can be used for storage of details related to the determination, such as the particular type of statistics used in determining the repeatability value. This element should always be used, when 'Other' is selected in **eRepeatMethod** [enumeration].

nRepeatValue [numerical, floating] is used for storage of the *repeatability value*. The units match those of the quantity being repeated.

nRepetitions [numerical, integer] is used for storage of the number of *measurement repetitions n* used in the calculation of the repeatability value.

****DeviceSpec*** [complex]

This complex element includes subelements for storage of components of uncertainty obtained as *device specifications* from manufacturers or certificates of calibration. The symbol ∗ indicates that this element is present in the schema for variables (***** = **Var**), constraints (***** = **Constr**), and properties (***** = **Prop**). These quantities are often used and reported as part of an uncertainty assessment by experimentalists, and may be of value to subsequent data evaluators.

sDeviceSpecEvaluator [string]; the *device specification evaluator* string is used to identify the individual or institution responsible for assessment of the device specification. In most cases, this will be a manufacturing company or an institute or company providing calibration services.

eDeviceSpecMethod [enumeration] provides an enumeration list for identification of the *device specification method*. The three enumerations are: (specified by the

manufacturer, calibrated by the experimentalist, calibrated or certified by a third party). Details related to the specification can be described in **sDeviceSpecMethod** [string].

sDeviceSpecMethod [string]; this element is used for storage of details related to the enumeration selected in **eDeviceSpecMethod** [enumeration]. Details might include particulars of the calibration method, identities and sources of reference materials, literature references to standard values, *etc.*

nDeviceSpecValue [numerical, floating] is used for storage of the numerical value of the uncertainty component arising from the device specification. The units match those of the state function being determined with the device.

nDeviceSpecLevOfConfid [numerical, floating] is used to store the *level of confidence* (percent) associated with **nDeviceSpecValue** [numerical, floating].

CurveDev *[complex]*

This element allows storage of uncertainty information derived from fitting of curves to experimental property data. By definition, these quantities are associated with properties only. The information stored is the root-mean-square deviation of the experimental values from the fitted curve (for a *data set*) and the deviations from the fitted curve for each numerical value (*i.e.*, for each *data point*), as indicated in Table 3.

nCurveDevAssessNum [numerical, integer]; the *curve deviation assessment number* is an integer used to identity a particular assessment. Its use is as described for **nUncertAssessNum** [numerical, integer]. The assessment number is needed to allow storage of results for fits with various equations. Identification of the particular equation is stored in **sCurveSpec** [string], as described below.

sCurveDevEvaluator [string]; the *curve deviation evaluator* element is used to identify the individual or institution responsible for the assessment.

sCurveSpec [string]; the *curve specification* element is use for storage of text that describes the fitted curve. The description might include a particular equation name (e.g., Antoine or Wagner for vapor pressures), an equation form (*e.g.*, $C_{p,m}$ = a + bT, for heat capacities of a liquid), or special conditions, such as specification of fixed parameters.

nCurveRmsDevValue [numerical, floating]; the *curve rms deviation value* is stored in this element, and has the same units as the associated property. This value is associated with the data set as a whole. The numerical value δ_{rms} is defined by the equation δ_{rms} =

$[\sum(x_i - x_{curve})^2 / n]^{0.5}$. The summation is over all data points, and the symbols represent the number of data points *n* and the deviation ($x_i - x_{curve}$) of data point *i* from the fitted curve.

nCurveRmsRelativeDevValue [numerical, floating]; the *curve rms relative deviation value* is stored in this element as a percentage. This value is also associated with the data set as a whole. Calculation of this value is as for **nCurveRmsDevValue** [numerical, floating], but with the deviations expressed as $[100 \cdot (x_i - x_{curve}) / x_{curve}]$ rather than $(x_i - x_{curve})$.

nCurveDevValue [numerical, floating]; the *curve deviation value* $(x_i - x_{curve})$ is stored in this element, and is the deviation of a particular numerical value (a *data point*) from the specified fitted curve. This value is associated always with an individual data point. The units are those of the property. Percentage values are not represented explicitly because they can be easily derived from the values provided.

Locations in ThermoML of elements for the expression of uncertainties and precisions

Locations in the PureOrMixtureData block. Element locations for uncertainties and precisions in this block are shown separately for constraints, variables, and properties. Elements for the expression of uncertainties and precisions for constraints are expanded in Fig15. All numerical values for constraints are associated with data sets rather than individual data points. Consequently, the element **nConstraintValue** [numerical, floating] is included within **Constraint** [complex], as seen in the figure. All elements for specification of the constraint uncertainty are within the element **ConstrUncertainty** [complex], as seen in Fig. 15. In contrast, it will be seen that elements for the expression of uncertainty for variables and properties must be split between the location associated with the data set and that associated with the individual data points.

Elements for the expression of uncertainties and precisions for variables in the *PureOrMixtureData* block are shown in Figs. 16 and 17. Fig. 16 shows those elements associated with the data set as a whole, while Fig. 17 shows the elements associated with the individual numerical data points. The element **nVarDeviceSpecValue** [numerical, floating] is associated with the individual data points because device specifications are

sometimes given as a function of the size of the measured value (*e.g.*, as a percentage), and are not constant for the entire data set. In addition, different devices may be used for measurements within a single data set. Generally, if different devices are used, it is preferable to identify a separate data set with each separate device.

Elements for the expression of uncertainties and precisions for properties in the *PureOrMixtureData* block are shown in Figs. 18 and 19. Fig. 18 shows those elements associated with the data set as a whole, while Fig. 19 shows the elements associated with the individual numerical data points. The additional elements for properties associated with representation of *combined uncertainties* and *deviations from fitted curves* are apparent in the figures.

Locations in the ReactionData block. Extensions to this block are also shown separately for constraints, variables, and properties. Detailed locations for all of the new elements for the expression of uncertainties and precisions for *constraints* in the *ReactionData* block are shown in Fig. 20. The analogous structure is shown in Fig. 15 for the *PureOrMixtureData* block.

Elements for the expression of uncertainties and precisions for *variables* in the *ReactionData* block are shown in Figs. 17 and 21. The schema structure for the representation of numerical values for variables (Fig. 17) is the same in the *PureOrMixtureData* block and the *ReactionData* block. Fig. 21 shows those elements associated with the data set as a whole, while Fig. 17 shows the elements associated with the individual numerical data points.

Elements for the expression of uncertainties and precisions for *properties* in the *ReactionData* block are shown in Figs. 18 and 22. The structure of the schema for the representation of numerical values for properties (Fig. 18) is the same in the *PureOrMixtureData* block and the *ReactionData* block. Fig. 22 shows those elements associated with the data set as a whole, while Fig. 18 shows the elements associated with the individual numerical data points.

Equation Representation of Property Data in ThermoML

Elements are described here that allow representation of property data as mathematical equations. The modular nature of XML is exploited here, and allows published MathML formats [30] to be used in conjunction with ThermoML, and

eliminates the need to develop here the complex structures needed to represent mathematical expressions. The Mathematical Markup Language (MathML) is a low-level specification for describing mathematics as a basis for machine-to-machine communication in terms of both content (mathematical meaning) and presentation (format). MathML (version 2.0) is a W3C (World Wide Web Consortium)[31] Recommendation, and was released February 21, 2001. It will be shown that equations can be defined by any user of ThermoML through use of the *ThermoML-EquationDefinition* schema. By linking the *ThermoML-EquationDefinition* schema to MathML, it is possible to take advantage of the full scope of elements developed for MathML in construction of the equation definition. Schema elements in ThermoML for equation representation provide for storage of the various equation components required for the specific equation definition. The nature or scope of the equations is not restricted in any way.

Some equation templates formulated with MathML are provided here for a variety of common equation types, such as the Antoine equation for vapor pressures, or a polynomial equation in terms of different variable powers commonly used for representation of a wide variety of properties, such as heat capacities or densities at saturation or constant pressure over relatively short temperature intervals. No attempt has been made (or will be made) to make this collection of templates comprehensive, because the variety of possible equation representations is infinite. The selection of the initial collection of equations was based on their use in existing data collections for chemical engineering. The provided templates serve as a convenience to ThermoML users who require the given specific formulations, and also, as examples for their general methods of construction. The structure of MathML syntax is not described in this paper; however, full descriptions are readily available [30].

Elements in ThermoML for Equation Representation of Property Data in ThermoML

The general locations of the **Equation** [complex] elements described here for equation representation of property data are indicated in Figs. 6 and 11. Subelements of **Equation** [complex] are identical for the *PureOrMixtureData* and *ReactionData* blocks, so only one description of the subelements is necessary.

The structure of the element **Equation** [complex] for the *PureOrMixtureData* and *ReactionData* blocks is shown in Fig. 23. Complex subelements of **Equation** [complex] are shown in Figs. 24 through 29. The first pair of elements within **Equation** [complex], **eEqName** [enumeration] and **sEqName** [string], allow specification of an equation name. The element **eEqName** [enumeration] allows selection from a list of the equation names provided within the ThermoML library of equation representations in ThermoMLEquation format. This list will grow as new representations are added. The element **sEqName** [string] is used to name an equation that is not part of the ThermoML library of equations. The element **urlMathSource** [Web address] is the location on the World Wide Web where the XML representation of the specified equation is stored. Five subelements of **Equation** [complex]; **EqProperty** [complex], **EqConstraint** [complex], **EqVariable** [complex], **EqParameter** [complex], and **EqConstant** [complex], are used to represent the components of an equation. *Indexes* are represented within these complex subelements to allow vector or matrix representation. **Covariance** [complex] stores the covariance for each of the equation parameter pairs, and **nCovarianceLevOfConfid** [numerical, floating] stores the level of confidence (in percent) associated with uncertainties calculated with the covariance values.

The names used for equations within the ThermoML library of equation representations follow the convention: *'ThermoML.(equation name).(property)'*. An example of this convention is *'ThermoML.Wagner.VaporPressure.'* It is recommended that other ThermoMLEquation representations created for use with ThermoML conform to this format, but this is not required. Duplicate equation names are not a problem because uniqueness is enforced through the particular **urlMathSource** [Web address] specified for the equation definition, and not through the equation name.

The structure of the subelement **EqProperty** [complex] within **Equation** [complex] is shown in Fig. 24. The elements **nPureOrMixtureDataNumber** [numerical, integer] and **nReactionDataNumber** [numerical integer] are numbers that are unique for each instance of the *PureOrMixtureData* block or *ReactionData* block, respectively. These elements are used for correct linking of properties (if needed) in equation representations, and are also included in **EqConstraint** [complex] and **EqVariable** [complex]. Within ThermoML it is possible to include any number of properties within the element

Property [complex] for a given chemical sample, mixture, or chemical reaction, as was shown in Figs. 8 and 14. **nPropNumber** [numerical, integer] within **EqProperty** [complex] (Fig. 24) provides the mechanism to identify a specific property from amongst those defined within **Property** [complex]. **sEqSymbol** [string] is used to map a symbol used in the equation definition (a ThermoMLEquation file) to a particular property defined in the ThermoML file. **nEqPropIndex** [numerical, integer] is used to map the property to a particular index in the equation definition. **sOtherPropUnit** [string] allows the user to define any unit preferred for the property. Because there are an infinite number of possible units, no attempt was made to create an enumeration list for these. Provision of the **sOtherPropUnit** [string] element was included to facilitate, for example, intra-company or interpersonal communications. All equations provided in the NIST/TRC library of representations include only SI units as defined earlier for the various properties. **nEqPropRangeMin** [numerical, floating] and **nEqPropRangeMax** [numerical, floating] define the range within which the equation is valid for the particular property.

The structure of the subelement **EqConstraint** [complex] within **Equation** [complex] is shown in Fig. 25. The subelements of **EqConstraint** [complex]: **nPureOrMixtureDataNumber** [numerical, integer], **nReactionDataNumber** [numerical, integer], **nConstraintNumber** [numerical, integer], **sEqSymbol** [string]; **nEqConstraintIndex** [numerical, integer], **sOtherConstraintUnit** [string], **nEqConstraintRangeMin** [numerical, floating], and **nEqConstraintRangeMin** [numerical, floating], are analogous to those of **EqProperty** [complex] described above. The term *constraint* used in the context of **EqConstraint** [complex] refers to a *constraint* defined as an immediate subelement of the *PureOrMixtureData* or *ReactionData* blocks. It is not necessarily a constraint for the equation. For example, an equation for which pressure is not constrained might be used to represent an experimental data set in which it is.

The structure of the subelement **EqVariable** [complex] within **Equation** [complex] is shown in Figure 26. The subelements of **EqVariable** [complex]: **nPureOrMixtureDataNumber** [numerical, integer], **nReactionDataNumber** [numerical, integer], **nVarNumber** [numerical, integer], **sEqSymbol** [string];

nEqVariableIndex [numerical, integer], **sOtherVarUnit** [string], **nEqVarRangeMin** [numerical, floating], and **nEqVarRangeMax** [numerical, floating], are analogous to those of **EqProperty** [complex] described above.

The structure of the subelement **EqParameter** [complex] within **Equation** [complex] (Fig. 23) is shown in Fig. 27. **nEqParNumber** [numerical, integer] associates the parameter to a particular row and column in the covariance matrix. Parameters without **nEqParNumber** [numerical, integer] values do not contribute to the covariance, if it is established independent of the represented vapor pressure values. **sEqParSymbol** [string] is used to map a symbol used in the equation definition (located on the World Wide Web) to a particular parameter defined in the ThermoML file. **nEqParIndex** [numerical, integer] is used to map the parameter to a particular index in the equation definition. One index is used for a vector of parameters and two indexes for a matrix element (*e.g.*, binary interaction parameters g represented as g_{ij} have indexes i and j). **nEqParValue** [numerical, floating] stores the numerical value of the parameter. **nEqParDigits** [numerical, integer] stores the total number of digits in **nEqParValue** [numerical, floating].

The structure of the subelement **EqConstant** [complex] within **Equation** [complex] is shown in Fig. 28. **sEqConstantSymbol** [string] is used to map a symbol used in the equation definition to a particular constant defined in the ThermoML file. **nEqConstantIndex** [numerical, integer] is used to map the constant to a particular index in the equation definition. **nEqConstantValue** [numerical, floating] stores the numerical value of the constant. **nEqConstantDigits** [numerical, integer] stores the total number of digits in **nEqConstantValue** [numerical, floating].

The structure of the subelement **Covariance** [complex] within **Equation** [complex] is shown in Figure 29. This subelement is used to store the elements of the covariance matrix. **nEqParNumber1** [numerical, integer] is used to specify a particular equation parameter, **nEqParNumber** [numerical, integer] specified within **EqParameter** [complex]. **nEqParNumber2** [numerical, integer] is used to specify a second parameter defined within **EqParameter** [complex]. **nCovarianceValue** [numerical, floating] stores the covariance value for the two defined parameters.

The ThermoMLEquation Schema: Equation definitions

The structure of the schema (ThermoMLEquation) for equation definitions is shown in Fig. 30. This schema can be used by anyone to create an equation representation for use with ThermoML. The subelements of the ThermoMLEquation schema for equation definition are defined in the following paragraphs. Indexes are not included explicitly in the ThermoMLEquation schema, but they are used for correct association of vector or matrix elements with numerical values in the ThermoML file, as demonstrated in the Supporting Information for this article.

The element **Version** [complex] is mandatory and provides for storage of the ThermoMLEquation version designation for the ThermoMLEquation schema. The subelements of **Version** [complex] are **nVersionMajor** [numerical, integer] and **nVersionMinor** [numerical, integer]. For example, if the version number of ThermoMLEquation were 1.2, the "major" element would store the value "1" and the "minor" element would store the value "2".

sEqName [string] stores the name of the equation, such as *ThermoML.Wagner.VaporPressure*. (This is in accord with the naming convention suggested earlier.) In the case of the NIST/TRC library of equation representations, the names correspond to those provided in the element **eEqName** [enumeration] shown in Fig. 23. The current enumeration list is provided in the text of the ThermoML schema included as Supporting Information. **sEqAltName** [string] stores alternative names for the same equation. **sEqDescription** [string] stores a general description of the equation in text format. The element **EqReference** [complex] provides the means to store locations of information related to the equation description. **EqReference** [complex] has the same substructure as **Citation** [complex] shown in Fig. 2.

Within any particular ThermoMLEquation representation, *properties*, *constraints*, and *variables*, as defined in ThermoML, are not distinguished. In the context of mathematics, all of these quantities are 'variables,' and are so named in a ThermoMLEquation file as **EqVariable** [complex] as shown in Fig. 30. Association of a particular *property*, *constraint*, or *variable* in a ThermoML data file with a 'variable' in a ThermoMLEquation file is made through the assigned symbol and indexes (if needed) specified in **EqProperty** [complex] (Figure 24), **EqConstraint** [complex] (Figure 25), or

EqVariable [complex] (Figure 26). This is why the element tag **sEqSymbol** [string] is the same for each of these complex elements (Figs. 24-26).

The subelements of **EqVariable** [complex] in the ThermoMLEquation schema all provide additional information related to the particular variable. **sEqSymbol** [string] is the symbol used in the representation of the equation formulated with MathML. The element **sEqVarComment** [string] stores a text description of the particular variable. For example, **sEqSymbol** [string] might contain 'T,' while **sEqVarComment** [string] contains 'temperature.' This is a simple way to provide some description of symbols within the text of a ThermoMLEquation file. The element **IUPACSymbol** [complex] is provided to take advantage of the 'presentation' elements of MathML, and allows standard IUPAC symbols [32] to be associated with particular quantities in the equation.

The subelement of **IUPACSymbol** [complex], **mml:math**, indicates that this portion of the ThermoMLEquation schema allows importation of elements from MathML. In this case, the elements would be MathML presentation elements for representation of an IUPAC symbol.

The elements **EqParameter** [complex] and **EqConstant** [complex] with ThermoMLEquation have the same substructure as **EqVariable** [complex]. The subelements are analogous to those for **EqVariable** [complex]. A parameter in a ThermoML data file is associated with a particular equation parameter through use of the **sEqParSymbol** [string] element shown in Fig. 27. Similarly, constants are associated with equation constants through the **sEqConstantSymbol** [string] (Fig. 28).

The representation of the equation in MathML format is stored in the complex elements **EqMathContent** [complex] and **EqMathPresentation** [complex]. **EqMathContent** [complex] is used to store the mathematical meaning of the equation, and **EqMathPresentation** [complex] is used to store information for display. Examples of the representation of several equations with MathML are included in the Supporting Information.

Fig. 31 shows a schematic representation of linking of *property*, *constraint*, *variable*, *parameter*, and *constant* information both within a ThermoML file and between a ThermoML and a ThermoMLEquation file. Particular schema elements through which the linking is accomplished are indicated on the figure.

Schema Validation: Extent and Strategy

The developed schema was validated extensively with data records from the SOURCE Data System [33]. Validation covered essentially all properties within the scope of ThermoML including pure compounds, multicomponent mixtures, and chemical reactions. More than 5,000 data sets from more than 3,000 publications were used at the TRC Data Entry Facility to validate the schema. In addition, validation included data files submitted to TRC by authors of upcoming publications submitted through the Editorial Board of the *Journal of Chemical and Engineering Data*, *The Journal of Chemical Thermodynamics*, *Thermochimica Acta*, and *Fluid Phase Equilibria*, as well as data files submitted to TRC by its data collection contractors.

Global Thermodynamic Data Communication Processes Based on ThermoML

Background. As discussed in the *Introduction*, there is a great demand for the establishment of efficient global data delivery processes. Until very recently, such a process did not exist in the field of thermodynamics. In fact, there are only two well known processes of this nature outside the field of thermodynamics: submission and retrieval of protein structures from the Protein Data Bank (PTB) [34] and submission and retrieval of crystal structures for smaller molecules from the Cambridge Structural Database (CSD) [35]. Establishing a global data delivery process for thermodynamic properties is yet more complex than that for the PTB and CSD because of the necessity to communicate information related to the numerous (>100) thermophysical, thermochemical, and transport properties commonly reported. Moreover, communicating such data is further complicated by the extensive system of thermodynamic metadata (variables, constraints, phases, methods, uncertainties) required. This complexity necessitated development of a software infrastructure to support the global delivery process for thermodynamic data. In order to address this need, several software applications have been developed.

Guided Data Capture (GDC) software. Guided Data Capture (GDC) software was developed at TRC [36,37], and serves as a data-capture expert by guiding extraction of information from the literature, assuring the completeness of the information extracted, validating the information through data definition, range checks, *etc*. A key feature of the

GDC software is the capture of information in close accord with customary original-document formats. The compiler's main interactions with GDC involve a navigation tree, which provides a visual representation in accord with the hierarchical structure of a typical source document. Operation of GDC including deletion, addition, and editing of all captured information is accomplished through interactions with the navigation tree. Numerical values are not shown explicitly in the tree, but may be accessed through the property-specification nodes. Lists of established field values (journal title abbreviations, compound identifiers, properties, units, phases, experimental methods, *etc.*) are stored in a local database, which is part of the GDC software. Selection of field values by the data compiler is, generally, achieved through pre-defined lists, which prevents many common errors. Most numerical values can be captured through electronic means (PDF files, spreadsheets, *etc.*) and rarely require manual input. All other input is accomplished through pre-defined menus, check boxes, or other controlled selection processes. GDC also provides simple ("one-click") graphical representation of the numerical data. This is a powerful tool for detection of typographical errors with data sets for thermophysical properties that can include data points numbering in the thousands. A version of GDC has been developed to output files in ThermoML format, and will be released for free download from the World Wide Web in 2005. Software designed to convert ThermoML formatted files to a simple spreadsheet format is under development at NIST/TRC.

As part of cooperation between TRC and major journals in the field, Guided Data Capture software and ThermoML are at the core of a global submission and dissemination process for thermodynamic data. Following the peer-review process, authors are requested by the journal editors to download and use the GDC software to capture the experimental property data that have been accepted for publication. The output of the GDC software is an electronic data file (plain text file), which is submitted directly to TRC. The electronic data file is checked and converted into ThermoML format with software developed at TRC. A key part of the checking process involves comparison of new property values submitted by authors against those generated by the ThermoData Engine (TDE) software [38, 39] developed at TRC. TDE is the first software implementation of the dynamic data evaluation concept for thermophysical property data. This concept requires the development of large electronic databases capable of storing

essentially all experimental data known to date with detailed descriptions of relevant metadata and uncertainties. The combination of these electronic databases with expert-system software, designed to automatically generate recommended data based on available experimental data, leads to the ability to produce critically evaluated data dynamically or 'to order'. The first release of TDE (version 1.0) is for pure compounds. Comparison of the submitted property values with those generated by TDE provides a powerful check for typographical and compound-identification errors.

Upon release of the manuscript for publication, the ThermoML files are posted on the public-domain TRC Web site [40] with unrestricted public access. This procedure was initiated with the *Journal of Chemical and Engineering Data* in January of 2003 [41], and now, also includes *The Journal of Chemical Thermodynamics* [42], Fluid Phase Equilbria [43], Thermochimica Acta [44], and soon, the *International Journal of Thermophysics* [45].

Direct communications between software applications. The ThermoData Engine software (TDE) also provides an example of direct data communication between applications. Properties generated by TDE are output in a traditional text file and in ThermoML format. Any software, such as a process simulation engine, that includes a ThermoML reader can communicate directly with TDE.

Fig. 32 illustrates the data delivery process from data suppliers (thermodynamicists reporting results of measurements of thermophysical and thermochemical property data *via* major journals in the field) to data users (chemical engineers *via* engineering software applications including chemical process design). The GDC software represents a key support element for data submission and ThermoML serves as the media to assure interoperability for propagation of the data across different platforms. ThermoML software 'readers' have been developed by a number of organizations in cooperation with NIST/TRC to transfer data from the ThermoML format to customized formats suitable for application software and databases [46]. This process is supported by standardization efforts with the participation of industry (DIPPR®), IUPAC, and the International Association of Chemical Thermodynamics (IACT, [47]).

Use Cases and ThermoML Schema Text

The complete text of the ThermoML schema and ThermoMLEquation schema are included as Supporting Information and are available on the Web [48] or through direct request to the authors. A variety of use cases including properties for pure compounds, mixtures, and chemical reactions are also included as Supporting Information together with examples of equation representation.

In development of the present IUPAC standard, several small changes were made relative to the previous ThermoML schema (version 3.0), which can invalidate some files generated with that earlier format. Changes were made typically to improve the clarity of meanings for tag names and to allow expansion of the scope of ThermoML. These changes are detailed in the Supporting Information.

Future Extensions of ThermoML

Properties involving multiple chemical systems are not included in the present version of ThermoML. For example, property changes involving the mixing of a two-component chemical system (compound A + compound B) with a second system (compound A + compound C) to form a third chemical system (compound A + compound B + compound C) are not represented in the current version of ThermoML. Such measurements are common, particularly in the fields of ionic solutions and biothermodynamics. Extensions to accommodate this these data types are planned.

Recommendations

The structure and description of ThermoML as a new IUPAC (XML-based IUPAC Standard for Experimental and Critically Evaluated Thermodynamic Property Data Storage and Capture) standard has been discussed and approved by the Task Group for IUPAC Project 2002-055-3-024 at its meeting on April 8th in Sesimbra, Portugal.

Acknowledgments

The authors express their appreciation to Drs. T. L. Teague (ePlantData, Houston, USA), D. L. Embry (ConocoPhillips, Ponca City, USA), A. K. Dewan (Shell, Houston, USA), S. Watanasiri (AspenTech, Cambridge, USA), M. Satyro (Virtual Materials Group, Calgary, Canada), A. I. Johns (National Engineering Laboratory, Glasgow, UK), M. Schmidt (Fiz-

Chemie, Berlin, Germany), A. R. H. Goodwin (Schlumberger Product Center, Sugarland, USA), J. W. Magee (NIST, Boulder, USA), M. Thijssen (Elsevier, Amsterdam, Netherlands), R. Craven (ESDU, London, UK), D. Lide (CRC Press, Gaithersburg, USA), W. M. Haynes (NIST, Boulder, USA), whose advice was very valuable for the development of ThermoML and its application to the global delivery process for thermophysical and thermochemical property data. In addition, the authors thank Drs. D. G. Friend, R. A. Perkins, M. L. Huber (NIST, Boulder, USA), G. J. Rosasco, P. Linstrom, G. W. Mallard (NIST, Gaithersburg, USA), for their practical suggestions related to the preparation of the published materials describing ThermoML. We wish to express our appreciation to Dr. A. N. Davies of Creon Lab Control AG (Frechen, Germany), the Secretary of the IUPAC Committee on Printed and Electronic Publications, for his unwavering support of the project. The authors wish to acknowledge the late Dr. Randolph Wilhoit of Texas A & M University (College Station, USA) who was an inspiration for the development of the standard.

REFERENCES

1. M. Frenkel. “Global Communications and Expert Systems in Thermodynamics: Connecting Property Measurement and Chemical Process Design”, *Pure Appl. Chem.*, **77** (2005). In press.
2. M. Frenkel, R. D. Chirico, V. V. Diky, Q. Dong, S. Frenkel, P. R. Franchois, D. L. Embry, T. L. Teague, K. N. Marsh, R. C. Wilhoit. “ThermoML-An XML-Based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data. 1. Experimental Data”, *J. Chem. Eng. Data*, **48**, 2-13 (2003).
3. R. C. Wilhoit, K. N. Marsh, *COdataSTAndardThermodynamics. Rules for Preparing COSTAT Message for Transmitting Thermodynamic Data*; Report to CODATA Task Group on Geothermodynamic Data and Chemical Thermodynamic Tables: Paris, 1987.
4. www-i5.informatik.rwth-aachen.de/lehrstuhl/projects/gco/
5. www.fiz-karlsruhe.de/dataexplorer/test/iucosped/dataexplorer.html
6. A. K. Dewan, D. L. Embry, T. J. Willman. “DIPPR/AIChE Project 991 – Thermophysical Property Data Exchange”, *Book of Abstracts of the 14-th Symposium on Thermophysical Properties*, Boulder, Colorado, 2000, p. 169.
7. http://www.iupac.org/projects/2002/2002-055-3-024.html
8. http://www.iupac.org/standing/cpep.html
9. C. Finkelstein, P. Aiken. *Building Corporate Portals with XML*. McGraw-Hill, New York (1999).
10. “XML-based IUPAC Standard for Experimental and Critically Evaluated Thermodynamic Property Data Storage and Capture”, *Chem. Int*. **26**(4), 26 (2004).
11. http://www.iupac.org/namespaces/ThermoML/index.html
12. P. Murray-Rust, H. S. Rzepa. “Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles”, *J. Chem. Inform. Comp. Sci.*, **39**, 938-942 (1999).
13. http://www.matml.org/
14. Frenkel, M.; Chirico, R. D.; Diky, V. V.; Dong, Q.; Frenkel, S.; Franchois, P. R.; Embry, D. L.; Teague, T. L.; Marsh, K. N.; Wilhoit, R. C. ThermoML - An XML-Based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data. 1. Experimental Data. *J. Chem. Eng. Data 48*, 2-13 (2003).
15. Chirico, R. D.; Frenkel, M.; Diky, V. V.; Marsh, K. N.; Wilhoit, R. C. ThermoML - An XML-Based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data. 2. Uncertainties. *J. Chem. Eng. Data 48*, 1344-1359 (2003).

16. Frenkel, M.; Chirico, R. D.; Diky, V. V.; Marsh, K. N.; Dymond, J. H.; Wakeham, W. A. ThermoML - An XML-Based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data. 3. Critically Evaluated Data, Predicted Data, and Equation Representation. *J. Chem. Eng. Data 49*, 381-393 (2004).

17. http://www.iupac.org/projects/2000/2000-025-1-800.html.

18. Dong, Q.; Yan, X.; Wilhoit, R. C.; Hong, X.; Chirico, R. D.; Diky, V. V.; Frenkel, M. Data Quality Assurance for Thermophysical Property DatabasessApplications to the TRC SOURCE Data System *J. Chem. Inf. Comput. Sci. 42*, 473-480 (2002).

19. (a) Whiting, W. B. Effects of Uncertainties in Thermodynamic Data and Models on Process Calculations. *J. Chem. Eng. Data 41*, 935-941 (1996). (b) Vasquez, V. R.; Whiting, W. B. Uncertainty and Sensitivity Analysis of Thermodynamic Models Using Equal Probability Sampling (EPS).*Computers Chem. Eng. 23*(11/12), 1825-1838 (2000).

20. See http://www.doi.org/

21. See http://www.iupac.org/projects/2000/2000-025-1-800.html

22. See http://www.ampolymer.com/

23. Frenkel, M.; Hong, X.; Dong, Q.; Yan, X.; Chirico, R. D. *Densities of Halohydrocarbons.* Landolt-Börnstein Series (Volume IV/8J), Springer-Verlag: Berlin, 2003**.**

24. *Guide to the Expression of Uncertainty in Measurement* (International Organization for Standardization, Geneva, Switzerland, 1993). This *Guide* was prepared by ISO Technical Advisory Group 4 (TAG 4), Working Group 3 (WG 3). ISO/TAG 4 has as its sponsors the BIPM, IEC, IFCC (International Federation of Clinical Chemistry), ISO, IUPAC (International Union of Pure and Applied Chemistry), IUPAP (International Union of Pure and Applied Physics), and OIML. Although the individual members of WG 3 were nominated by the BIPM, IEC, ISO, or OIML, the *Guide* is published by ISO in the name of all seven organizations.

25. *U. S. Guide to the Expression of Uncertainty in Measurement*, ANSI/NCSL Z540-2-1997, ISBN 1-58464-005-7. NCSL International: Boulder, Colorado. 1997.

26. Taylor, B. N.; Kuyatt, C. E. *Guidelines for the Evaluation and Expression of Uncertainty in NIST Measurement Results*, NIST Technical Note 1297; NIST: Gaithersburg, Maryland. 1994.

27. Dong, Q.; Chirico, R. D.; Yan, X.; Hong, X.; and Frenkel, M. Uncertainty Reporting for Experimental Thermodynamic Properties. *J. Chem. Eng. Data* **2005**. In press.

28. *International Vocabulary of Basic and General Terms in Metrology*, second edition (International Organization for Standardization, Geneva, Switzerland, 1993). This document (abbreviated VIM) was prepared by ISO Technical Advisory Group 4

(TAG 4), Working Group 1 (WG 1). ISO/TAG 4 has as its sponsors the BIPM, IEC, IFCC (International Federation of Clinical Chemistry), ISO, IUPAC (International Union of Pure and Applied Chemistry), IUPAP (International Union of Pure and Applied Physics), and OIML. The individual members of WG 1 were nominated by BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, or OIML, and the document is published by ISO in the name of all seven organizations.

29. Ostle, B. *Statistics in Research: Basic Concepts and Techniques for Research Workers*; Iowa State University Press: Ames, 1958.
30. Sandhu, P. *The MathML Handbook*; Charles River Media, Inc.: Hingham, Massachusetts, U.S.A., 2003. See also: www.w3.org/Math/
31. See http://www.w3.org/.
32. Mills, I.; Cvitas, T.; Homann, K.; Kallay, N. and Kuchitsu, K. *Quantities, Units and Symbols in Physical Chemistry (The Green Book)*; Blackwell Science: Oxford, 1993.
33. Frenkel, M.; Dong, Q.; Wilhoit, R. C.; Hall, K. R. TRC SOURCE Database: A Unique Tool for Automatic Production of Data Compilations. *Int. J. Thermophys. 22*, 215-226 (2001).
34. See http://www.rcsb.org/pdb/
35. See. http:/www.ccdc.cam.ac.uk/products/csd/
36. Diky, V. V.; Chirico, R. D.; Wilhoit, R. C.; Dong, Q.; Frenkel, M. Windows-Based Guided Data Capture Software for Mass-Scale Thermophysical and Thermochemical Property Data Collection. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 15-24.
37. See http://www.trc.nist.gov/GDC.html
38. Frenkel, M.; Chirico, R. D.; Diky, V.; Yan, X.; Dong, Q.; Muzny, C. ThermoData Engine (TDE): Software Implementation of the Dynamic Data Evaluation Concept. *J. Chem. Inf. Mod.* Submitted.
39. See http://www.nist.gov/srd/nist103.htm
40. See http://www.trc.nist.gov/ThermoML.html
41. Marsh, K. N. New Process for Data Submission and Dissemination. *J. Chem. Eng. Data 48*, 1 (2003).
42. Electronic Data Submission to the NIST Thermodynamics Research Center. *J. Chem. Thermodynamics 36*, iv (2004)
43. Electronic data submission to NIST Thermodynamics Research Center. *Fluid Phase Equilib. 226*, v (2004)
44. Electronic data submission to NIST Thermodynamics Research Center. *Thermochim. Acta 421*, 241 (2004)
45. *Int. J. Thermophysics* (2005). In press.

46. Names of commercial products and/or commercial entities are provided for complete scientific description and as a service to the reader of this publication. Such identification neither constitutes nor implies endorsement of such products or companies by NIST or by the U. S. Government. Other products or services may be found to be just as good.
47. See http://www.iactweb.org/Projects/projects.htm
48. http://www.iupac.org/dhtml_home.html

Table 1. Relationships between quantities used for the expression of uncertainty in ThermoML

State functions ***(Measurand, Y)***	**Measurement Result, *y***	**Standard Uncertainty (1σ)** [a] ***u***	**Coverage Factor** [b] ***k***	**Expanded Uncertainty** ***U***	**Level of Confidence** [b] ***L (percent)***
Variable(s), Y_V	y_V	$u_V = f(v_1, v_2, v_3, ...)$	k_V	$U_V = u_V \cdot k_V$	$L_V = f(k_V)$
Constraint(s), Y_C	y_C	$u_C = f(c_1, c_2, c_3, ...)$	k_C	$U_C = u_C \cdot k_C$	$L_C = f(k_C)$
Property, Y_P	y_P	$u_P = f(p_1, p_2, p_3, ...)$	k_P	$U_P = u_P \cdot k_P$	$L_P = f(k_P)$
For *Properties* Y_P (only)		**Combined Standard Uncertainty (1σ)** [c]	**Combined Coverage Factor**	**Combined Expanded Uncertainty**	**Combined Level of Confidence**
Property, Y_P	y_P	$u_{comb} = f(u_V, u_C, u_P)$	k_{comb}	$U_{comb} = u_{comb} \cdot k_{comb}$	$L_{comb} = f(k_{comb})$

[a] All components of uncertainty are included except those of other state functions.

[b] For many practical situations with assumed normal distributions, a coverage factor *k* near 2 corresponds to a level of confidence *L* near 95 percent.

[c] Standard uncertainties of variables and constraints are propagated to the uncertainty of the property.

Table 2. Names and locations of ThermoML elements for the expression of uncertainties

Abbreviated Element Name	Full Element Name	*Measurand* Type [a]	Location [b] Set	Value
**Uncertainty* [complex] [c]	*Uncertainty*			
nUncertAssessNum [numerical, integer]	Uncertainty assessment number	P, V, C	●	●
sUncertEvaluator [string]	Uncertainty evaluator	P, V, C	●	
sUncertEvalMethod [string]	Uncertainty evaluation method	P, V, C	●	
nStdUncertValue [numerical, floating]	Standard uncertainty value u_x	P, V, C		●
AsymStdUncertainty [complex] [d]	Coverage factor k_x used to obtain $U_x = k_x \cdot u_x$	P only	●	
nCoverageFactor [numerical, floating]	Coverage factor k_x used to obtain $U_x = k_x \cdot u_x$	P, V, C	●	
nExpandUncertValue [numerical, floating]	Expanded uncertainty value U_x	P, V, C		●
AsymExpandUncertainty [complex] [d]	Asymmetrical uncertainty	P only	●	
nUncertLevelOfConfid [numerical, floating]	Level of Confidence (%) L_x associated with U_x	P, V, C	●	
CombinedUncertainty	*Combined Uncertainty*	P only		
nCombUncertAssessNum [numerical, integer]	Combined uncertainty assessment number	P only	●	●
sCombUncertEvaluator [string]	Combined uncertainty evaluator	P only	●	
eCombUncertEvalMethod [enumeration]	Combined uncertainty evaluation method	P only	●	
sCombUncertEvalMethod [string]	Combined uncertainty evaluation method	P only	●	
nCombStdUncertValue [numerical, floating]	Combined standard uncertainty value u_{comb}	P only		●
AsymCombStdUncertainty [complex] [d]	Combined standard uncertainty value u_{comb}	P only		●
nCombCoverageFactor [numerical, floating]	Coverage factor k_{comb} to obtain $U_{comb} = k_{comb} \cdot u_{comb}$	P only	●	
nCombExpandUncertValue [numerical, floating]	Combined expanded uncertainty value U_{comb}	P only		●
AsymCombExpandUncertainty[complex] [d]	Combined expanded uncertainty value U_{comb}	P only		●
nCombUncertLevelOfConfid [numerical, floating]	Level of Confidence (%) L_{comb} associated with U_{comb}	P only	●	

[a] P = property, V = variable, C = constraint.

[b] *Location* indicates the location of the element in the ThermoML schema. *Set* specifies elements associated with an entire data set, and *Value* specifies elements associated with individual numerical values.

[c] The name of this complex element is different for a property (* = *Prop*), the variable (* = *Var*), and constraint (* = *Constr*).

[d] Subelements of this complex element represent the positive and negative values of the asymmetrical uncertainty.

Table 3. Names and locations of ThermoML elements for the expression of precisions

Abbreviated Element Name [1]	Full Element Name	Measurand Type [2]	Location [3] Set	Location [3] Value
n*Digits [numerical, integer]	Number of digits	P, V, C		●
**Repeatability* [complex]	*Repeatability*			
sRepeatEvaluator [string]	Repeatability evaluator	P, V, C	●	
eRepeatMethod [enumeration]	Repeatability method	P, V, C	●	
sRepeatMethod [string]	Repeatability method	P, V, C	●	
nRepeatValue [numerical, floating]	Repeatability value	P, V, C		●
nRepetitions [numerical, integer]	Number of repetitions	P, V, C	●	
**DeviceSpec* [complex]	*Device Specification*			
sDeviceSpecEvaluator [string]	Device specification evaluator	P, V, C	●	
eDeviceSpecMethod [enumeration]	Device specification method	P, V, C	●	
sDeviceSpecMethod [string]	Device specification method	P, V, C	●	
nDeviceSpecValue [numerical, floating]	Uncertainty based on device specification only	P, V, C		●
nDeviceSpecLevOfConfid [numerical, floating]	Level of confidence for nDeviceSpec Value	P, V, C	●	
CurveRmsDev [complex]	*RMS deviation from a fitted curve*			
nCurveDevAssessNum [numerical, integer]	Fitted curve assessment number	P only	●	●
sCurveDevEvaluator [string]	Fitted curve evaluator	P only	●	
sCurveSpec [string]	Fitted curve specification	P only	●	
nCurveRmsDevValue [numerical, floating]	Fitted curve RMS-deviation value	P only	●	
nCurveRmsRelativeDevValue [numerical, floating]	Fitted curve relative RMS-deviation value (percent)	P only	●	
nCurveDevValue	Deviations from the fitted curve	P only		●

[1] The names of some elements are different for a property (* = *Prop*), the variable (* = *Var*), and constraint (* = *Constr*).
[2] P = property, V = variable, C = constraint.
[3] *Location* indicates the location of the element in the ThermoML schema. *Set* specifies elements associated with a data set as a whole, and *Value* specifies elements associated with individual numerical values.

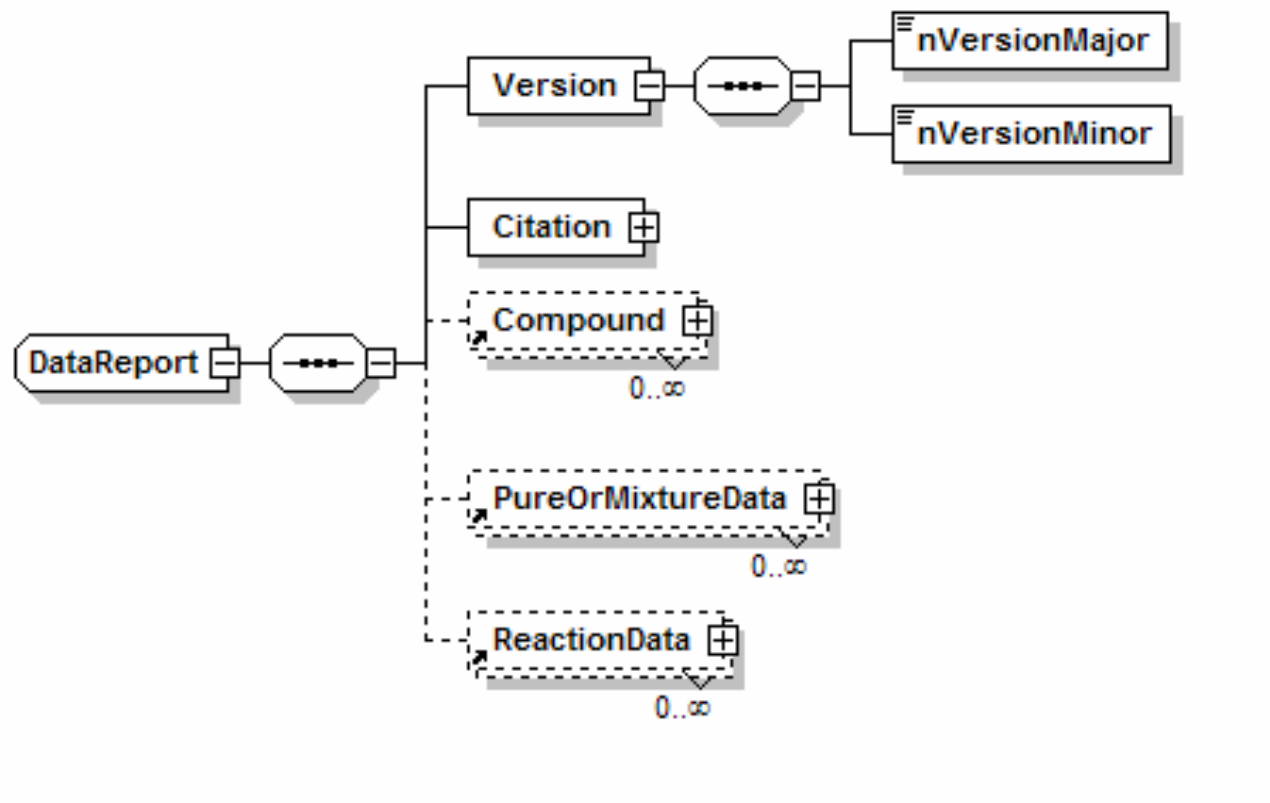

Fig. 1. Major components of the ThermoML schema.

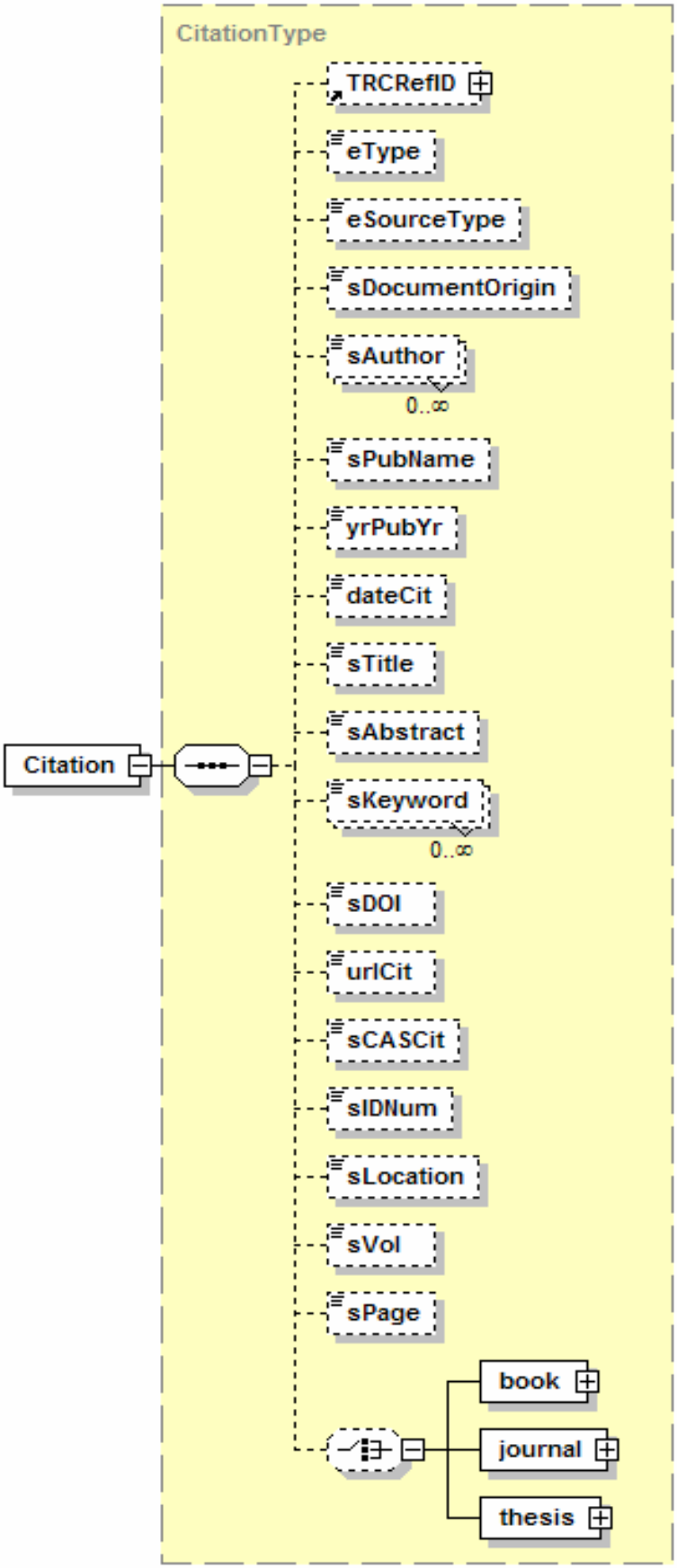

Fig. 2. Structure of the *Citation* block.

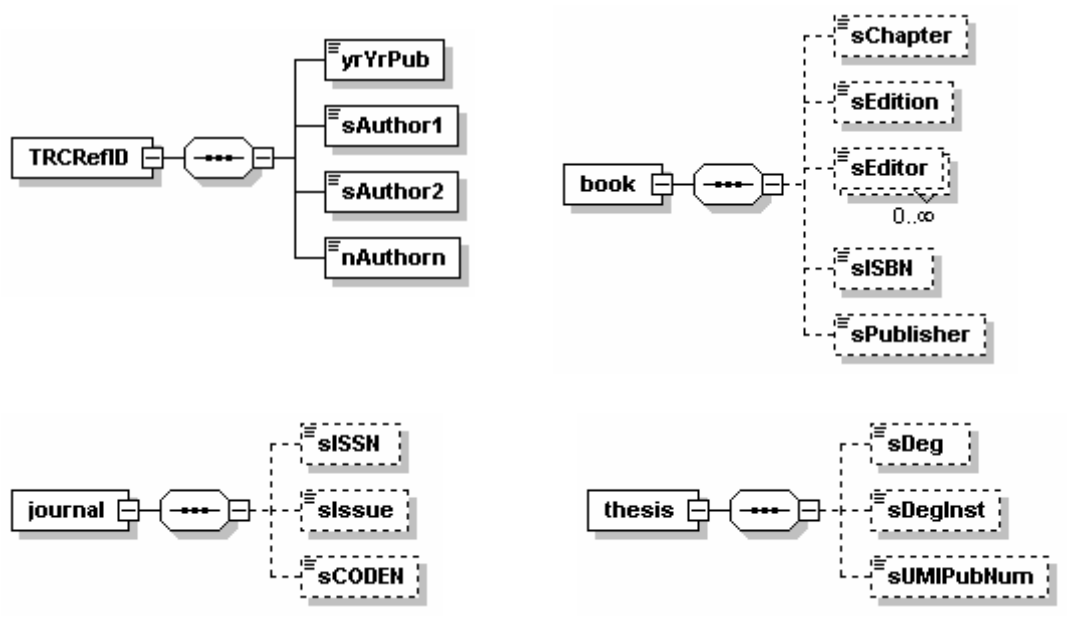

Fig. 3. Structures of the **TRCRefID**, **book**, **journa**l, and **thesis** complex elements of the *Citation* block.

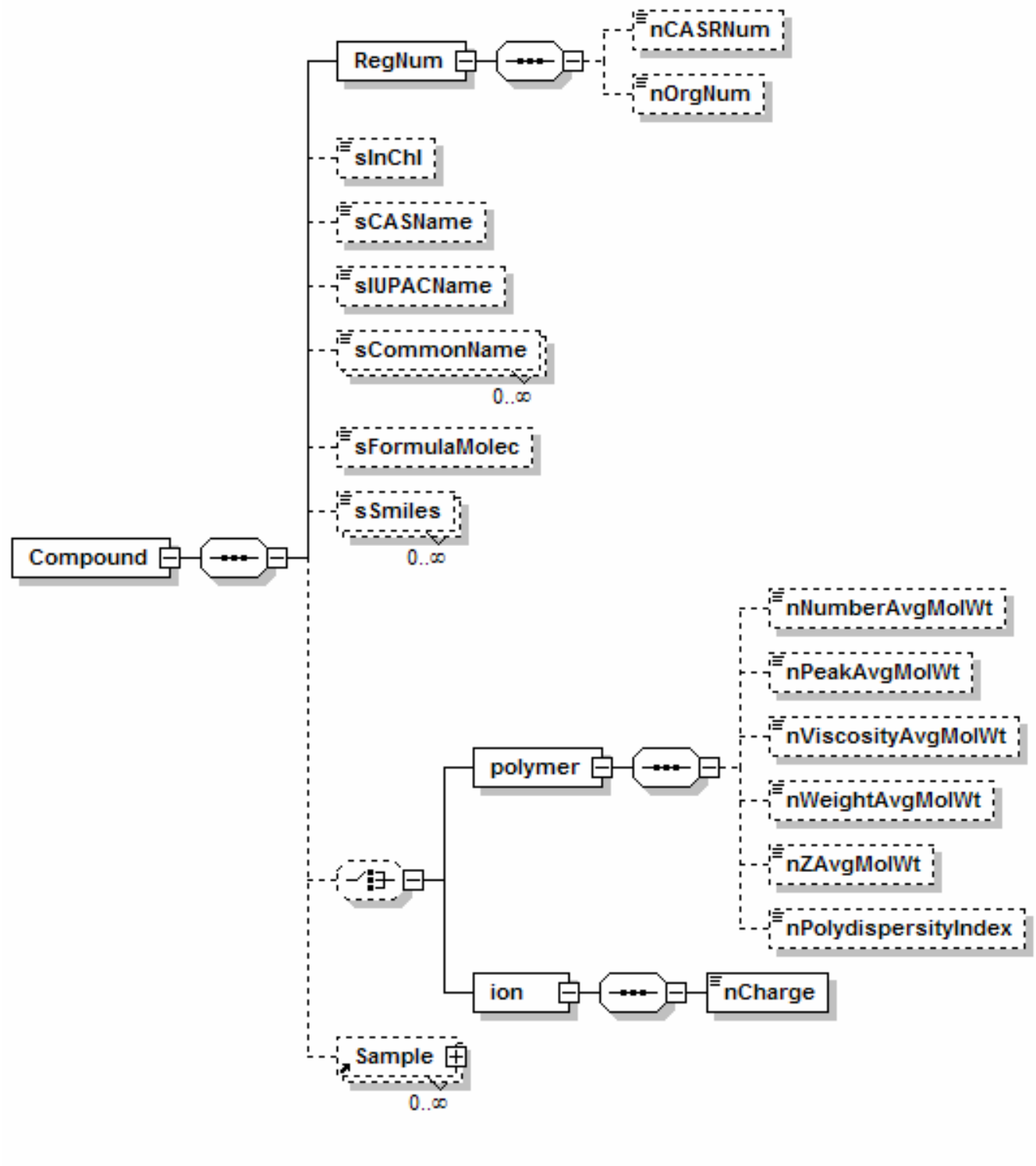

Fig. 4. Structure of the *Compound* block.

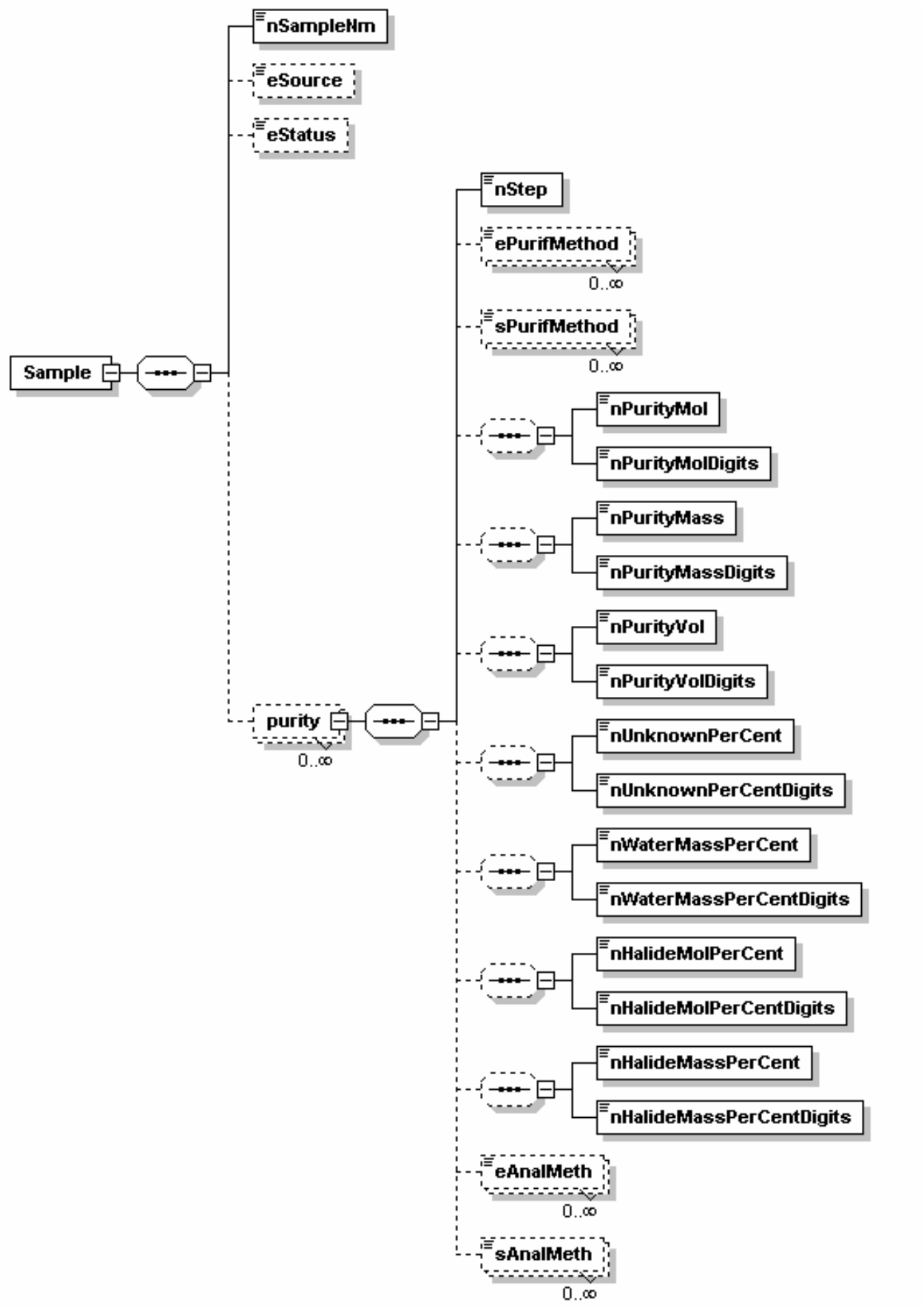

Fig. 5. Structure of the **Sample** [complex] element of the *Compound* block.

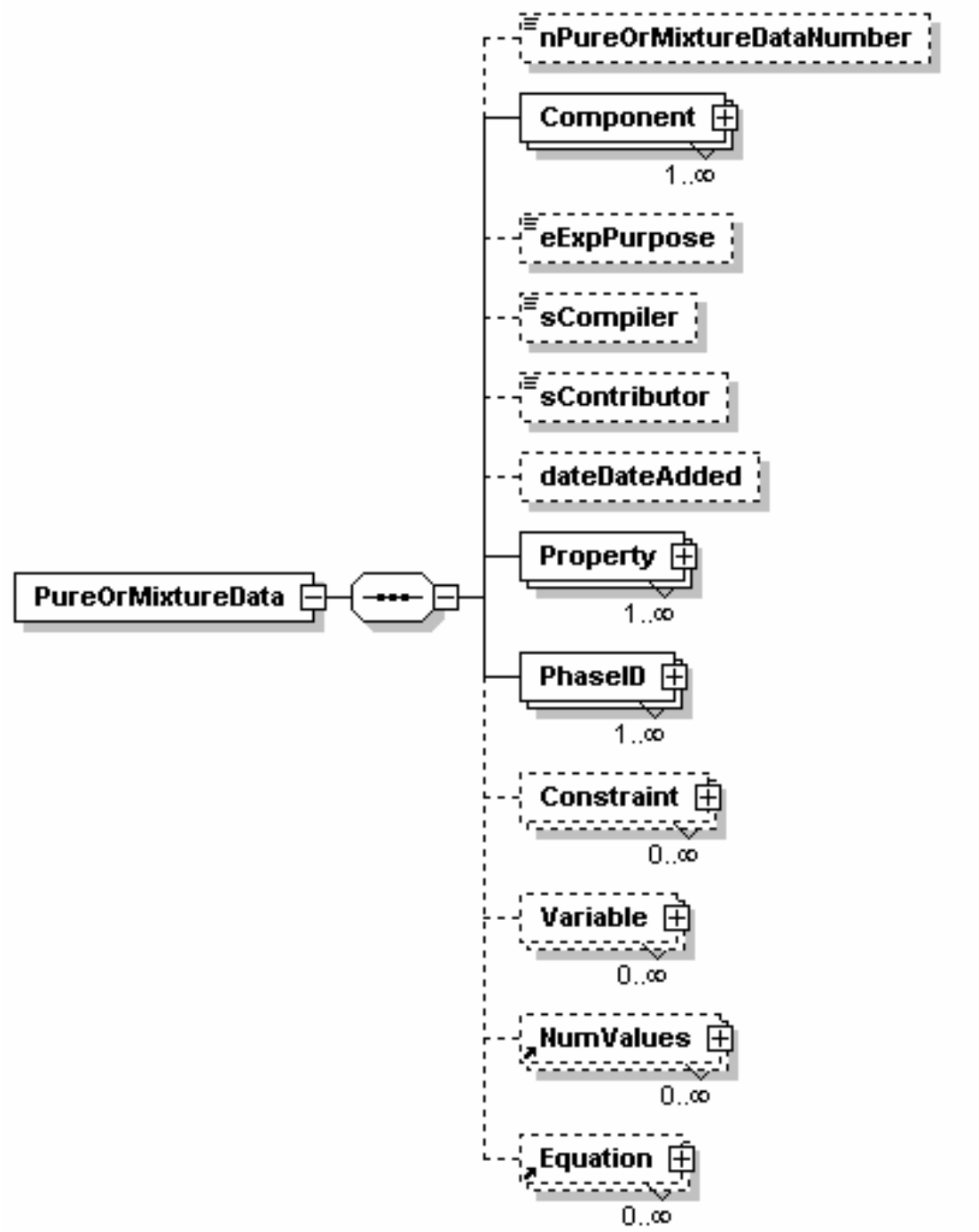

Fig. 6. Structure of the *PureOrMixtureData* block.

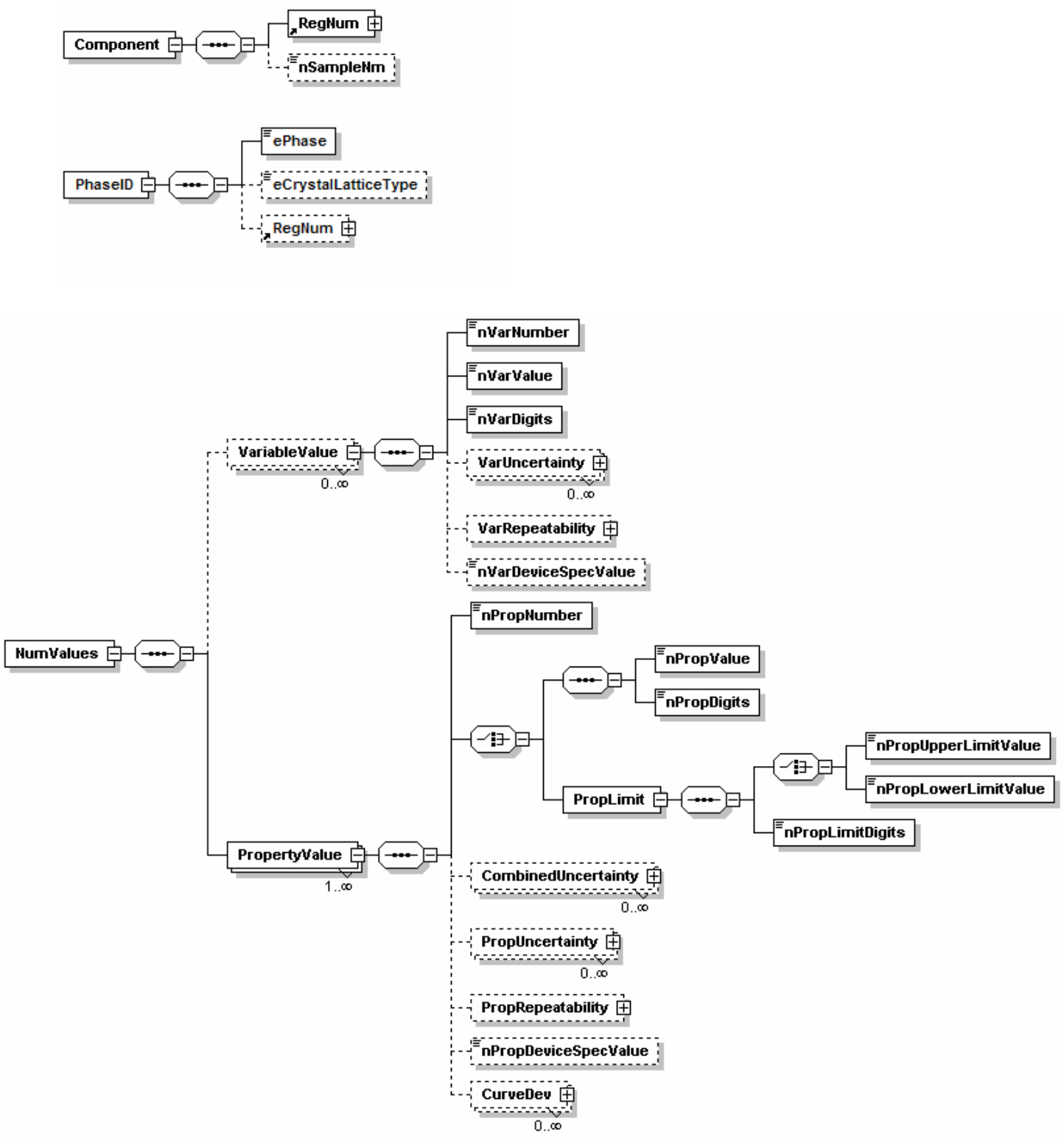

Fig. 7. Structures of the **Component** [complex], **PhaseID** [complex] and **NumValues** [complex] elements of the *Pure or Mixture Data* block.

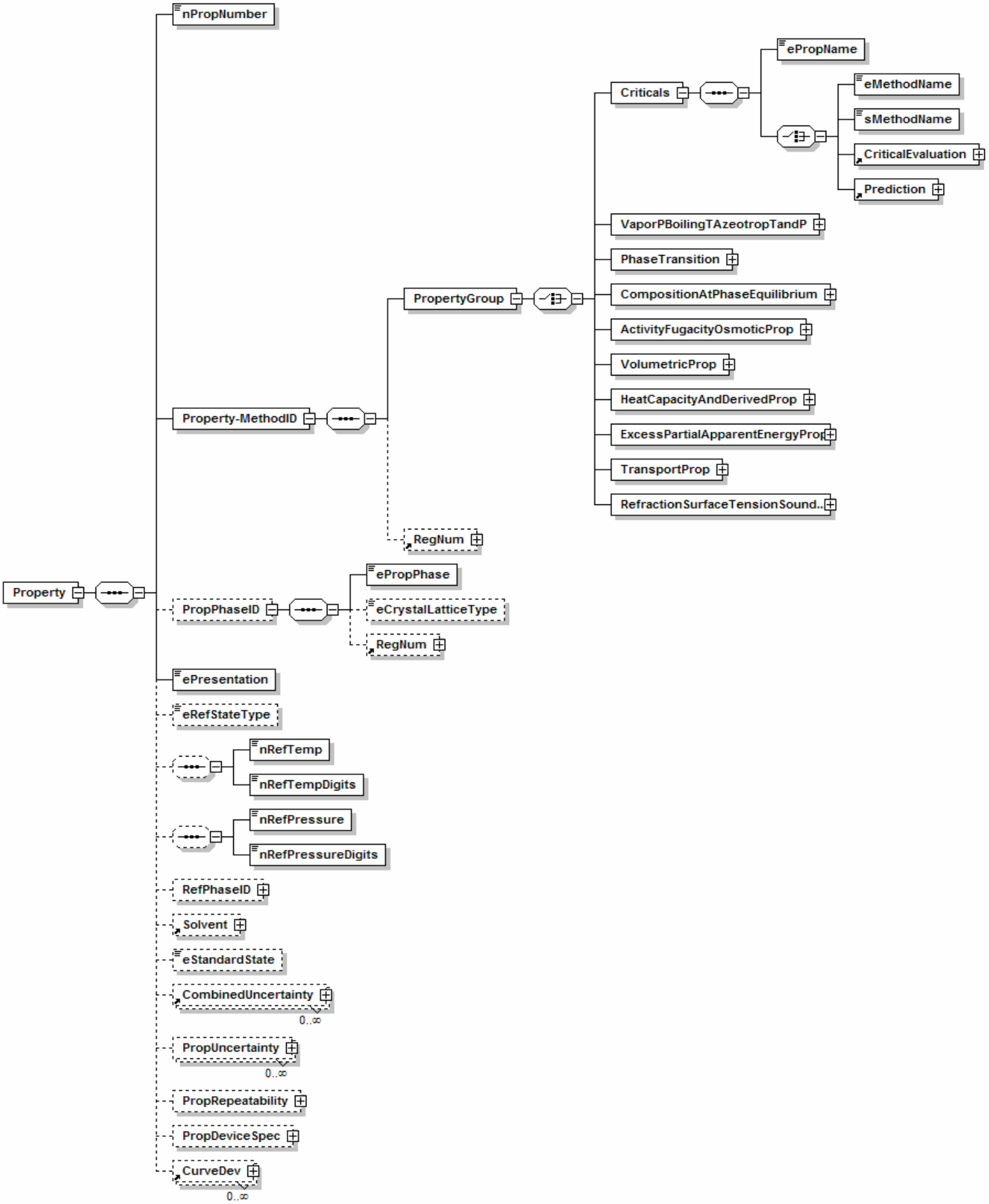

Fig. 8. Structure of the element **Property** [complex] of the *PureOrMixtureData* block.

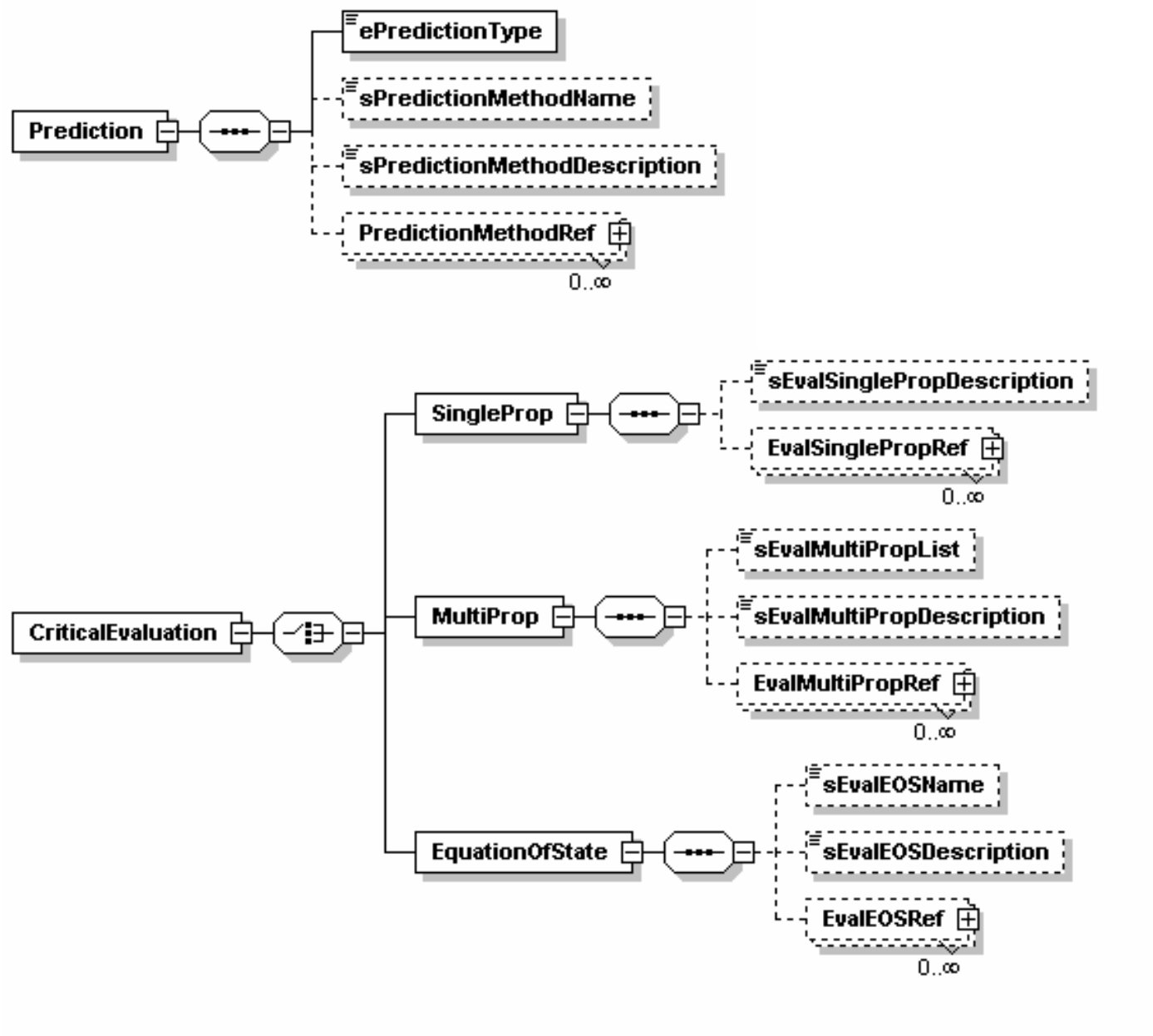

Fig 9. Structure of the **Prediction** [complex] and **CriticalEvaluation** [complex] subelements.

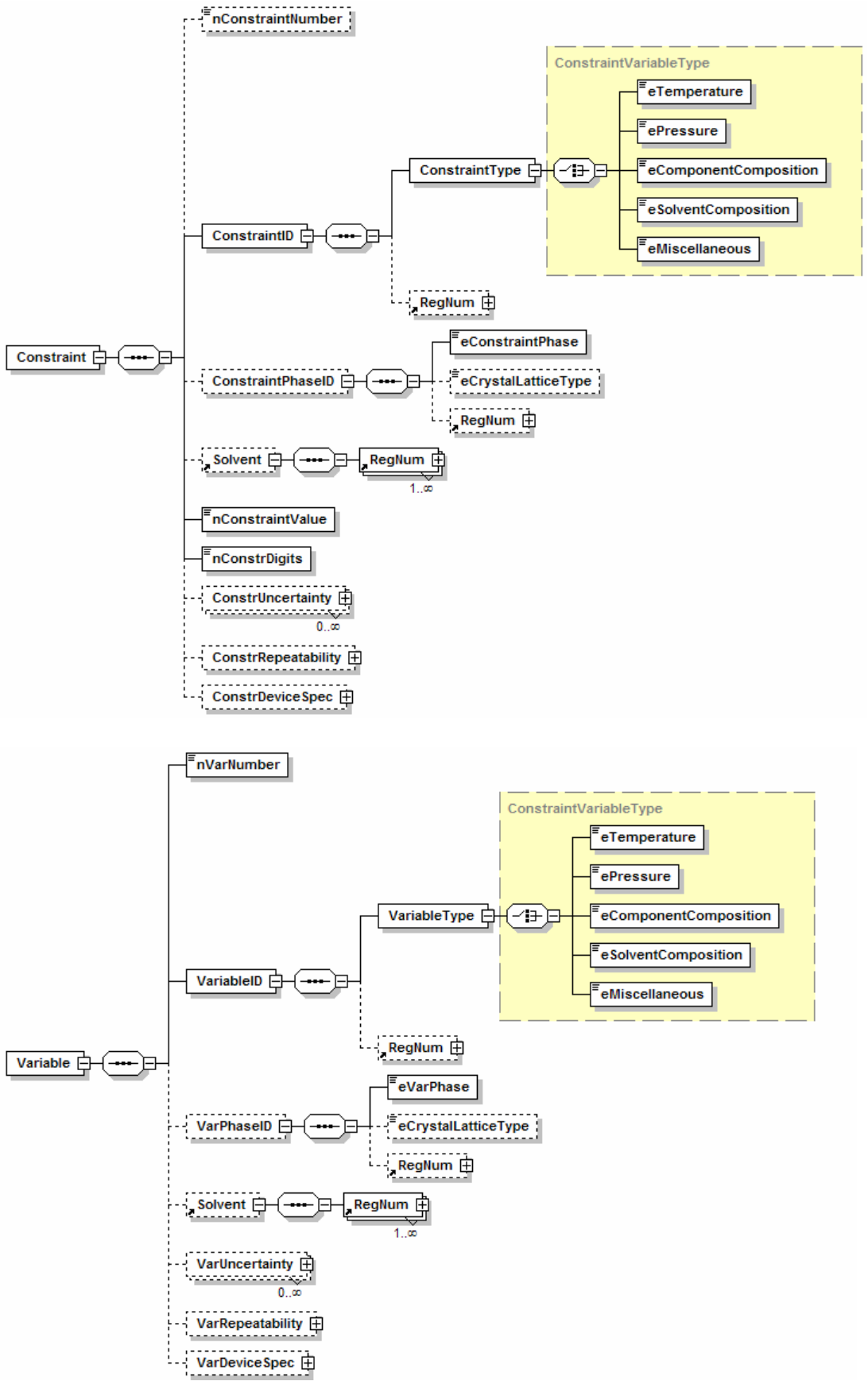

Fig. 10. Structures of the **Constraint** [complex] and **Variable** [complex] elements of the *PureOrMixtureData* block.

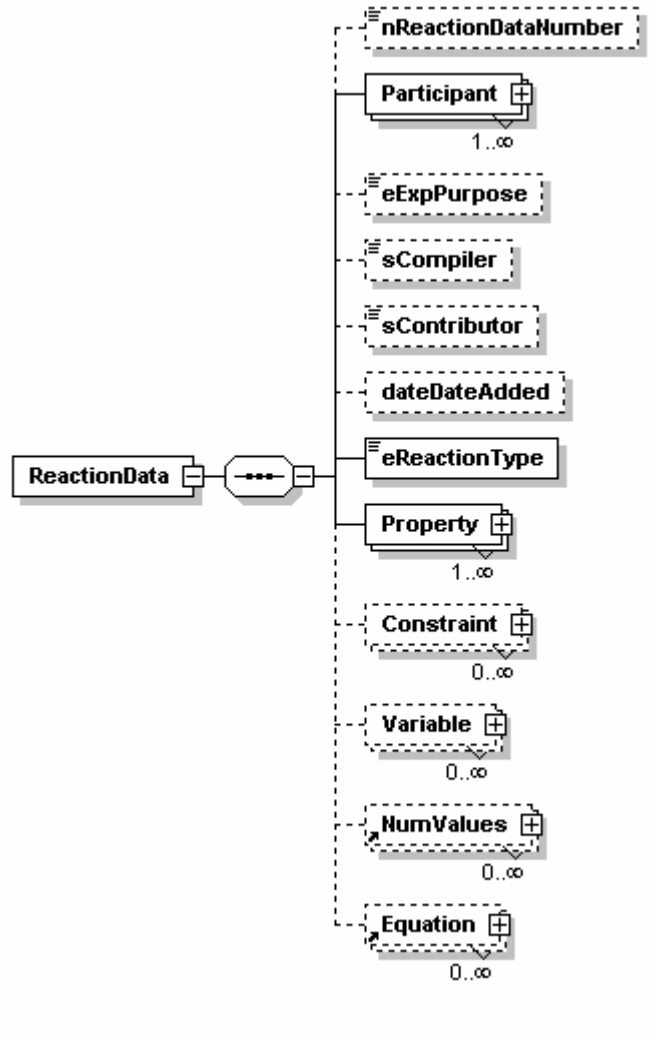

Fig. 11. Structure of the *ReactionData* block.

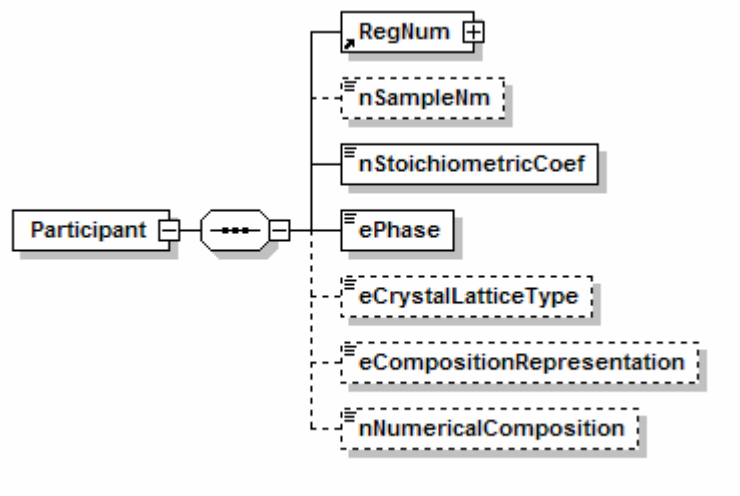

Fig. 12. Structure of the **Participant** [complex] element of the *ReactionData* block.

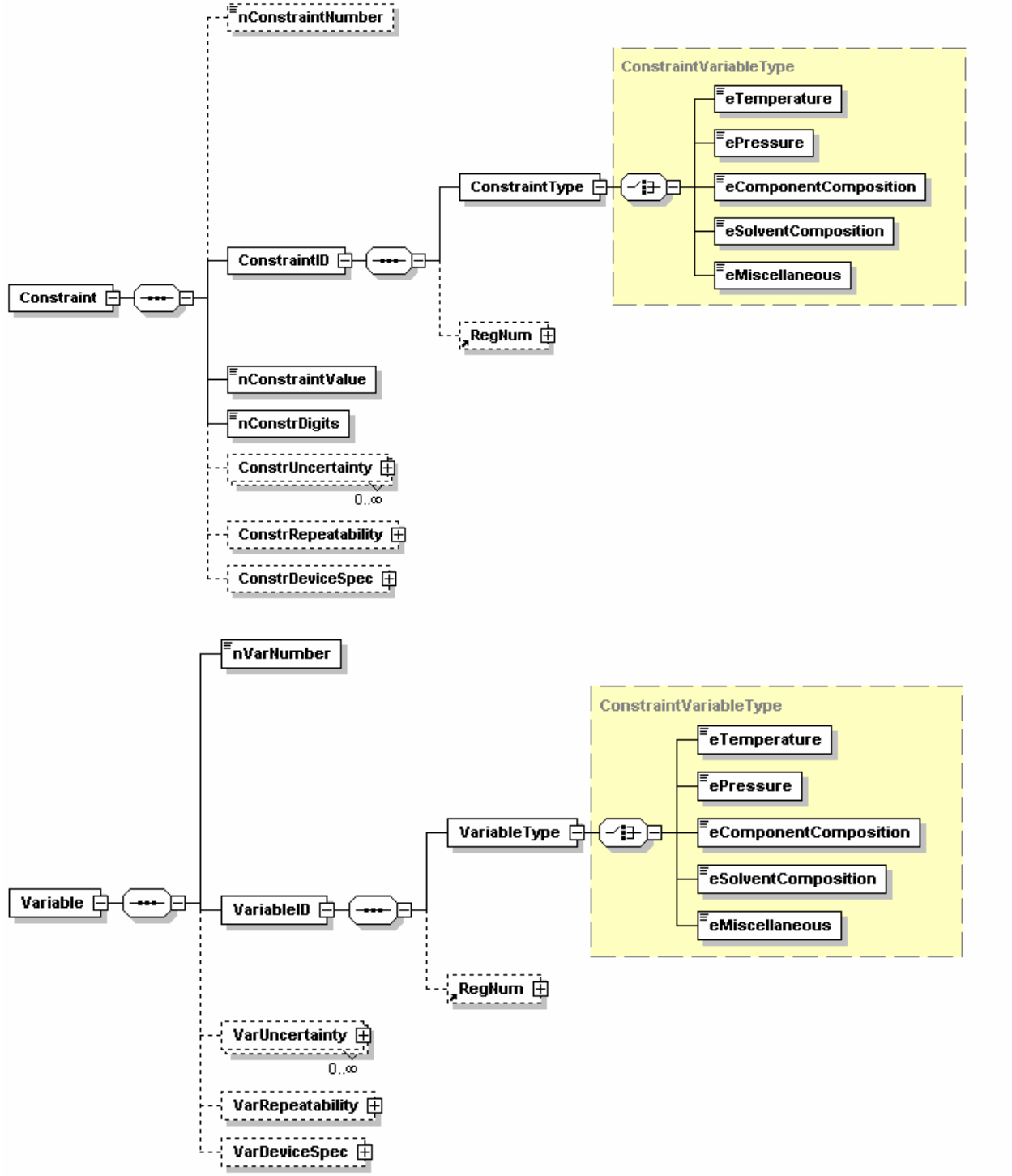

Fig. 13. Structures of the **Constraint** [complex] **Property** [complex] element of the *ReactionData* block.

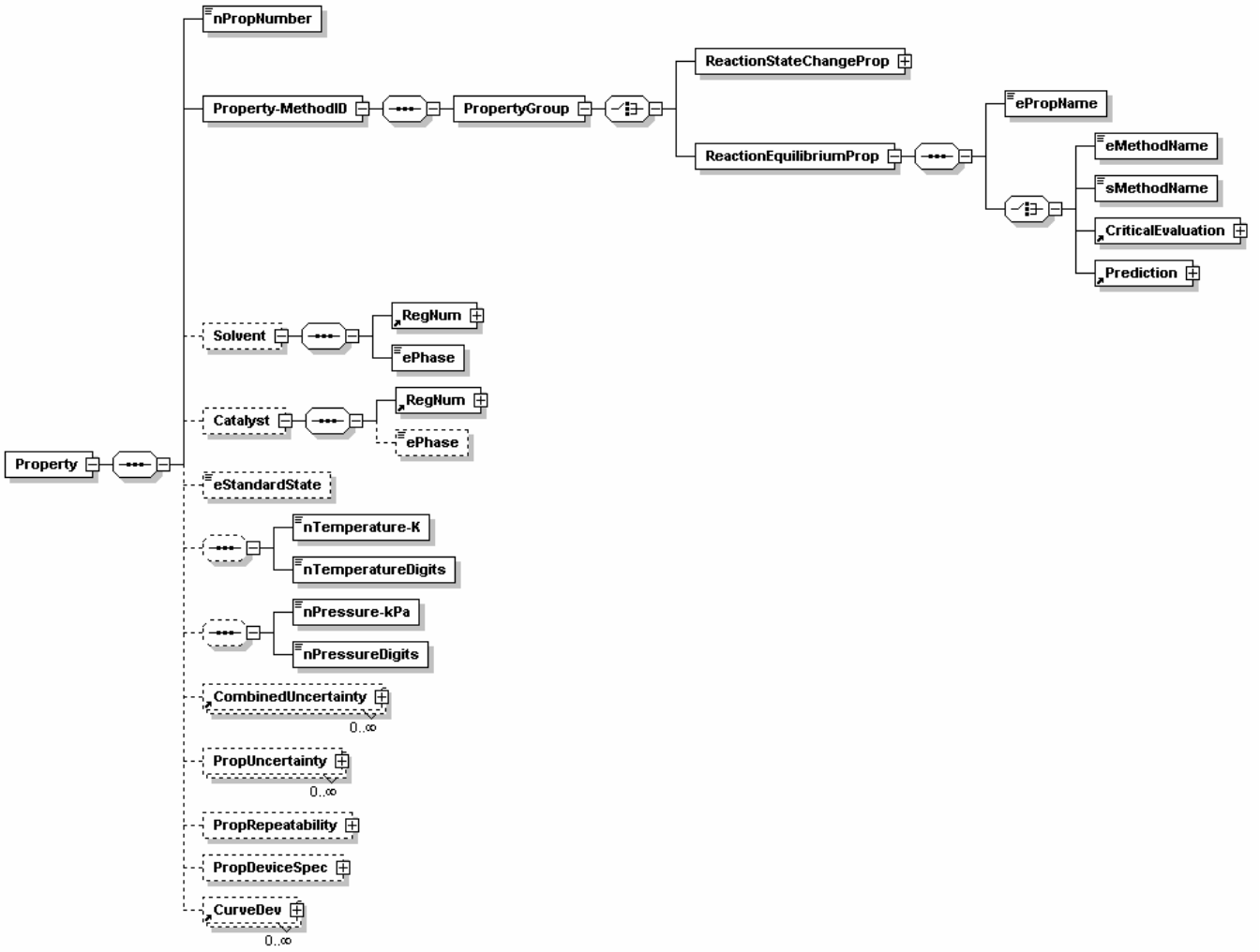

Fig. 14. Structure of the **Property** [complex] element of the *ReactionData* block.

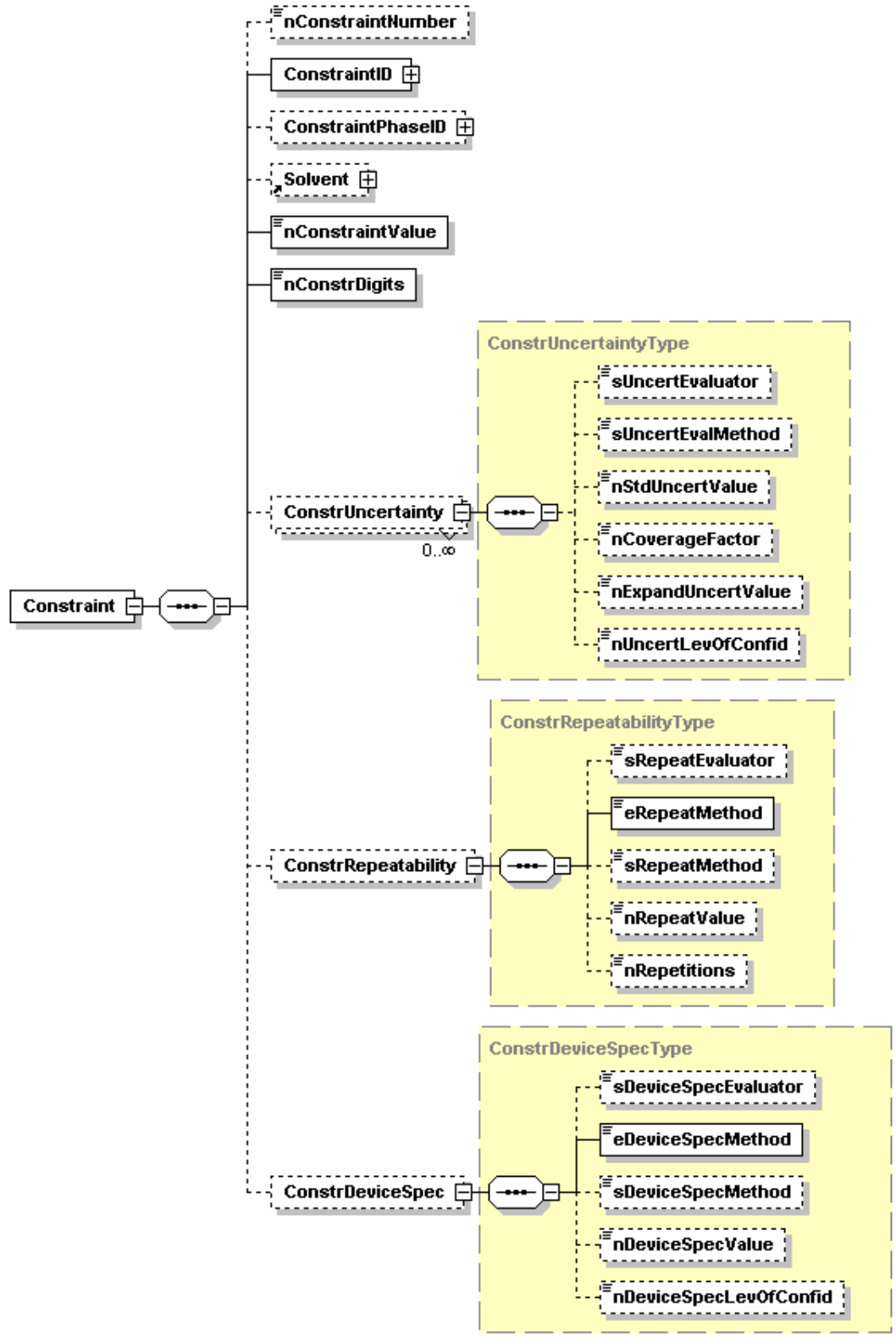

Fig. 15. Structure of the **Constraint** [complex] element in the *PureOrMixtureData* block with elements for expression of uncertainties (**ConstrUncertainty** [complex]) and precisions (**ConstrRepeatability** [complex] and **ConstrDeviceSpec** [complex] expanded.

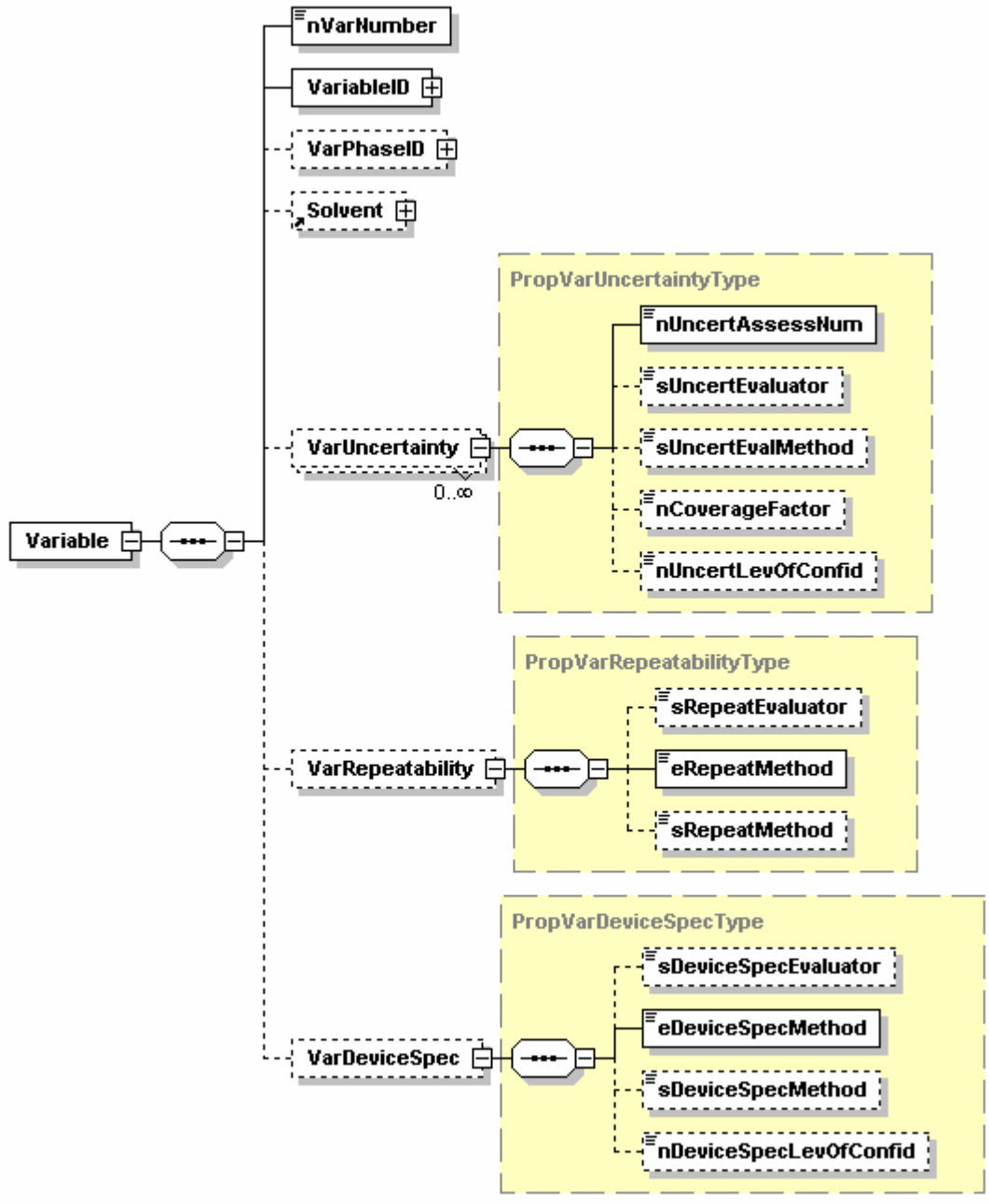

Fig. 16. Structure of the **Variable** [complex] element in the *PureOrMixtureData* block with elements for expression of uncertainties (**VarUncertainty** [complex]) and precisions (**VarRepeatability** [complex] and **VarDeviceSpec** [complex]) expanded.

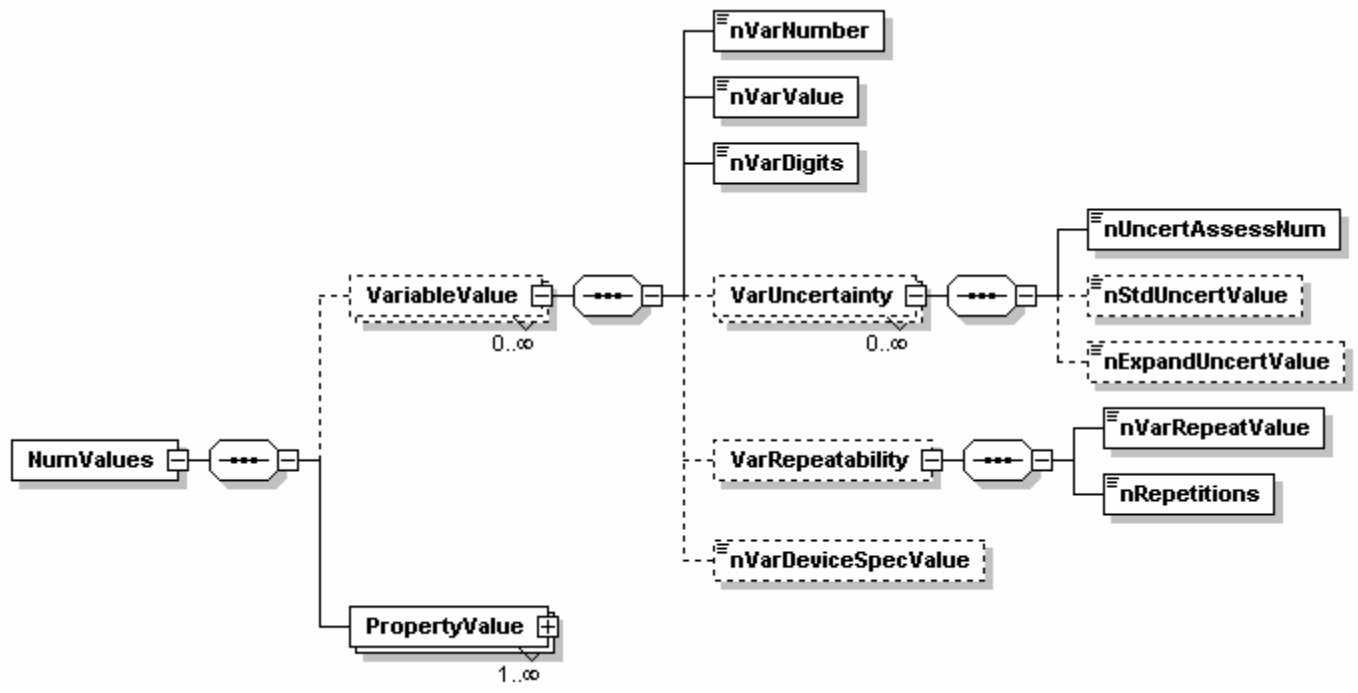

Fig. 17. Structure of the **VariableValue** [complex] subelement within the **NumValues** [complex] element in the *PureOrMixtureData* block and in the *ReactionData* block with elements for expression of uncertainties (**VarUncertainty** [complex]) and precisions (**VarRepeatability** [complex] and **nVarDeviceSpecValue** [numerical, floating]) expanded.

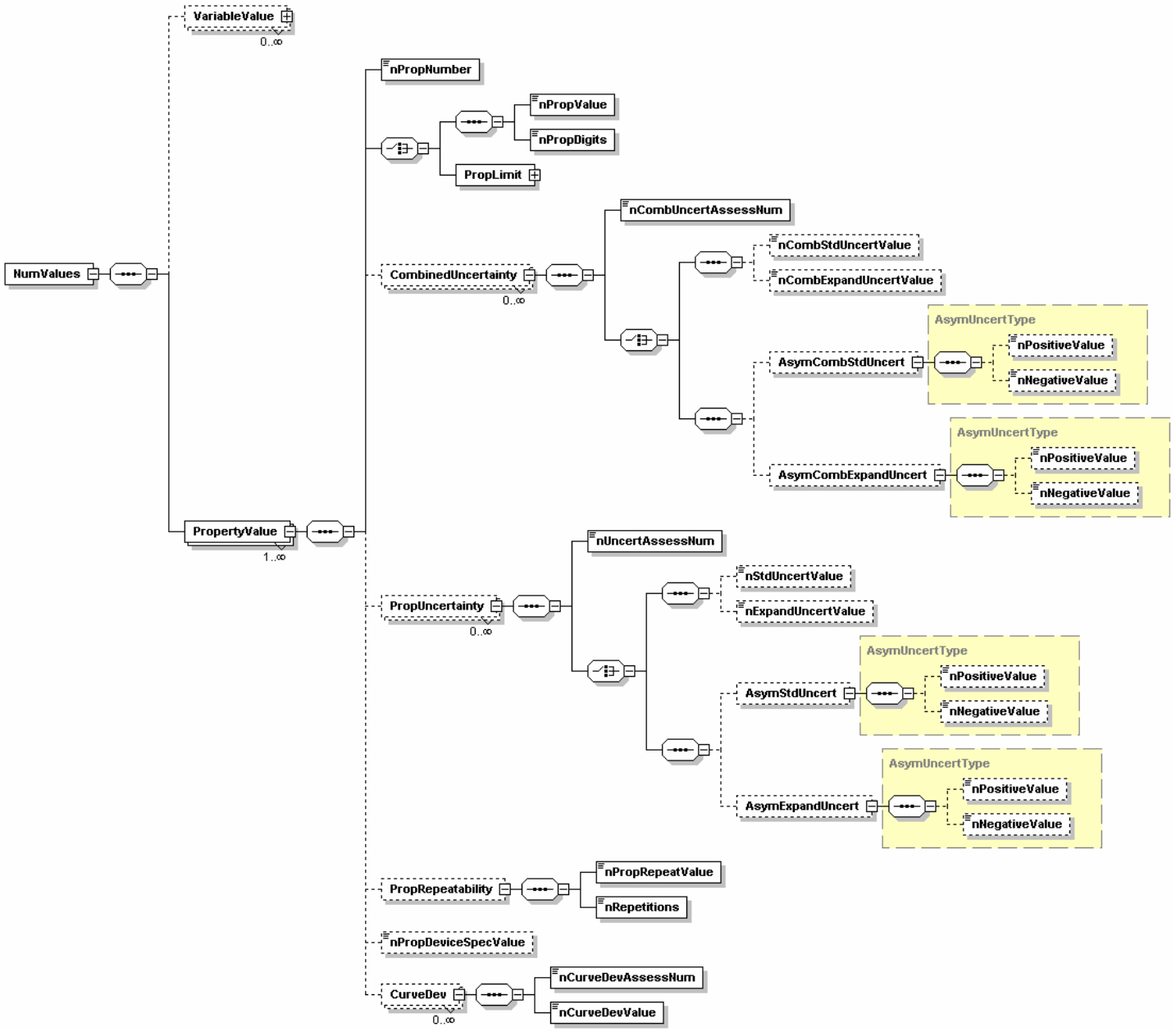

Fig. 18. Structure of the **PropertyValue** [complex] subelement within the **NumValues** [complex] element in the *PureOrMixtureData* block and in the *ReactionData* block with elements for expression of uncertainties (**CombinedUncertainty** [complex] and **PropUncertainty** [complex]) and precisions (**PropRepeatability** [complex], **nPropDeviceSpecValue** [numerical, floating], and **CurveDev** [complex]) expanded.

Fig. 19. Structure of the **Property** [complex] element in the *PureOrMixtureData* block with expression of uncertainties (**CombinedUncertainty** [complex] and **PropUncertainty** [complex]) and precisions (**PropRepeatability** [complex], **PropDeviceSpec** [complex], and **CurveDev** [complex]) expanded.

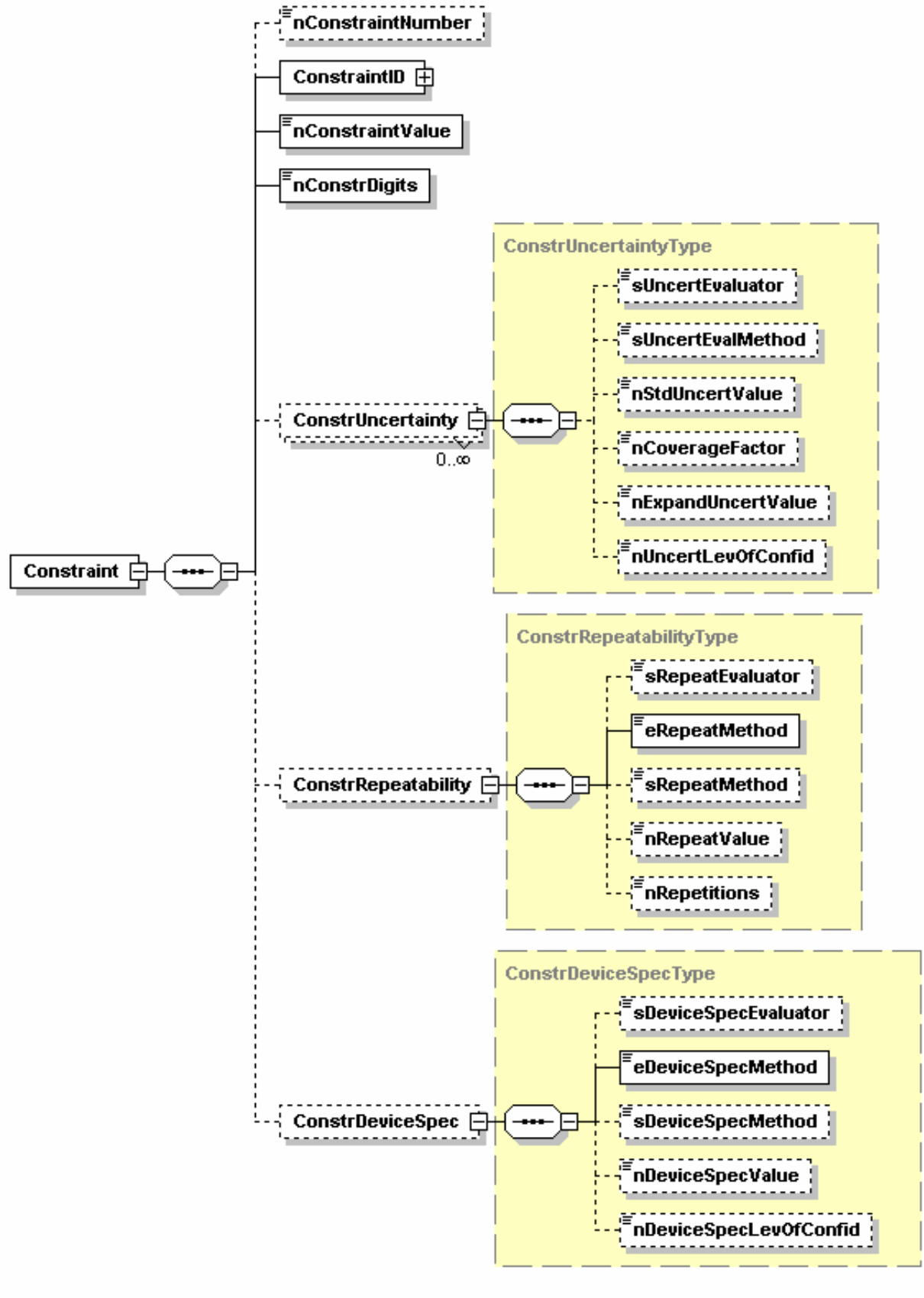

Fig. 20. Structure of the **Constraint** [complex] element in the *ReactionData* block with elements for expression of uncertainties (**ConstrUncertainty** [complex]) and precisions (**ConstrRepeatability** [complex] and **ConstrDeviceSpec** [complex] expanded.

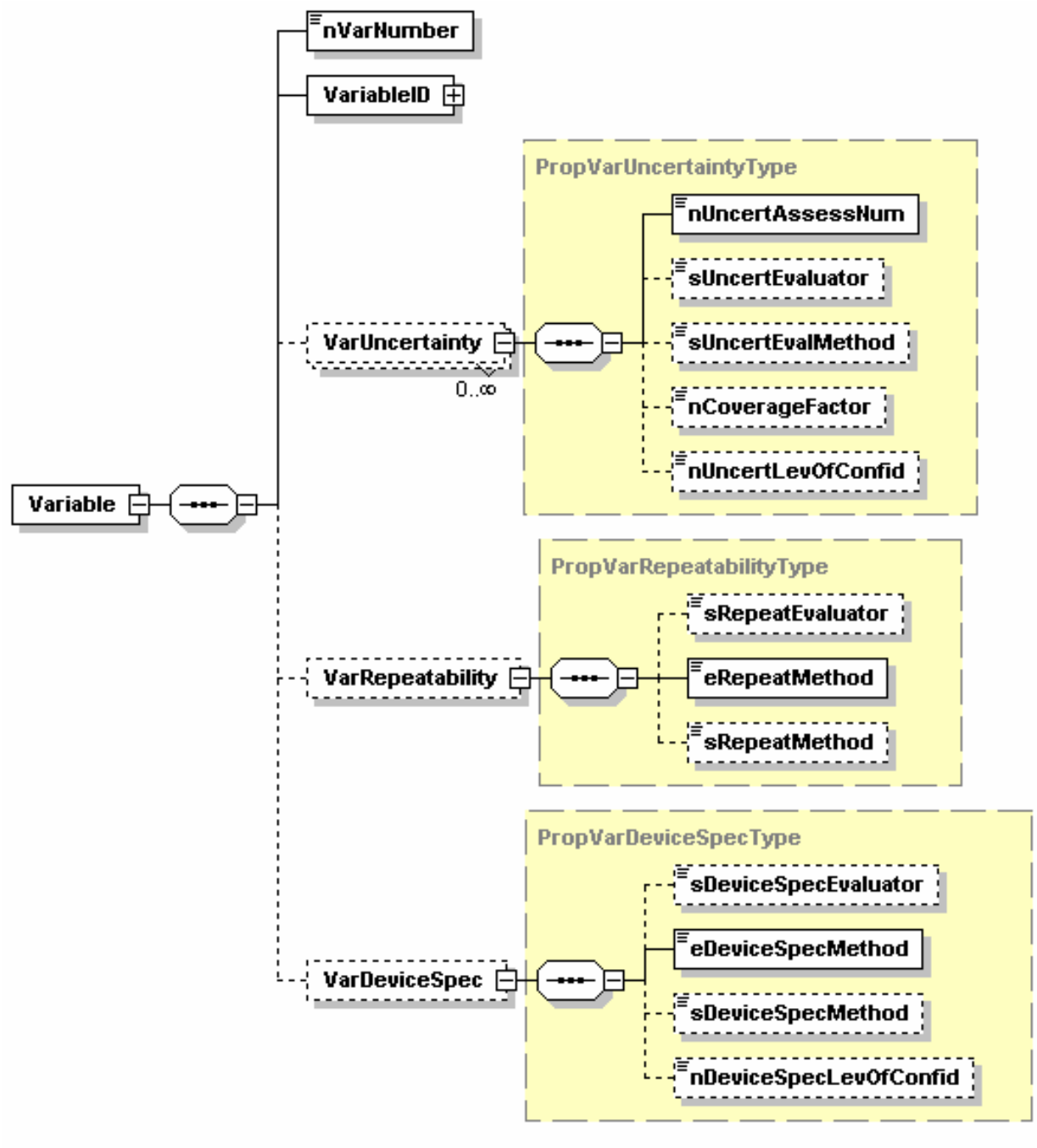

Fig. 21. Structure of the **Variable** [complex] element in the *ReactionData* block with elements for expression of uncertainties (**VarUncertainty** [complex]) and precisions (**VarRepeatability** [complex] and **VarDeviceSpec** [complex]) expanded.

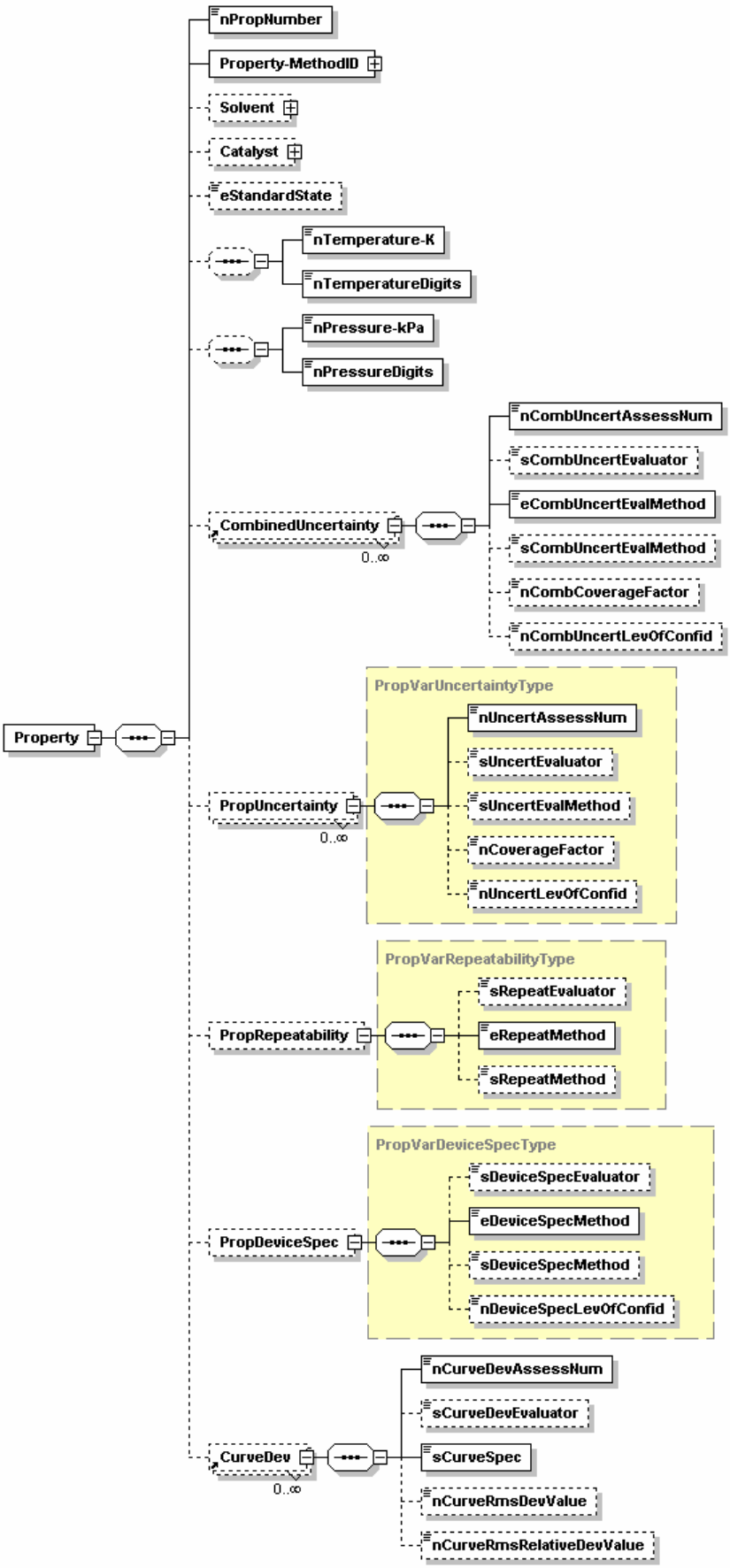

Fig. 22. Structure of the **Property** [complex] element in the *ReactionData* block with expression of uncertainties (**CombinedUncertainty** [complex] and **PropUncertainty** [complex]) and precisions (**PropRepeatability** [complex], **PropDeviceSpec** [complex], and **CurveDev** [complex]) expanded.

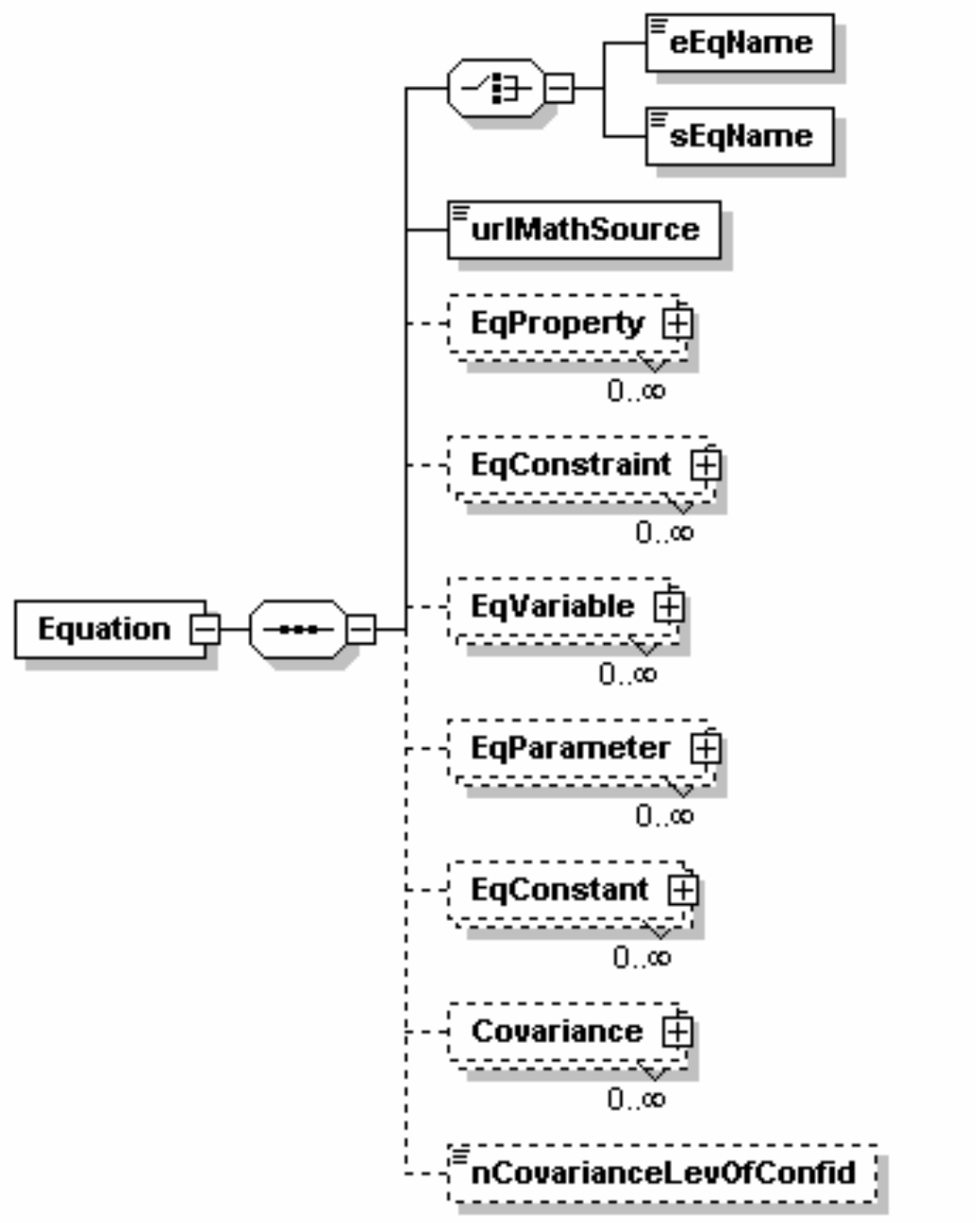

Fig. 23. Structure of the **Equation** [complex] element for equation representation. This element occurs in the *PureOrMixtureData* block and the *ReactionData* block.

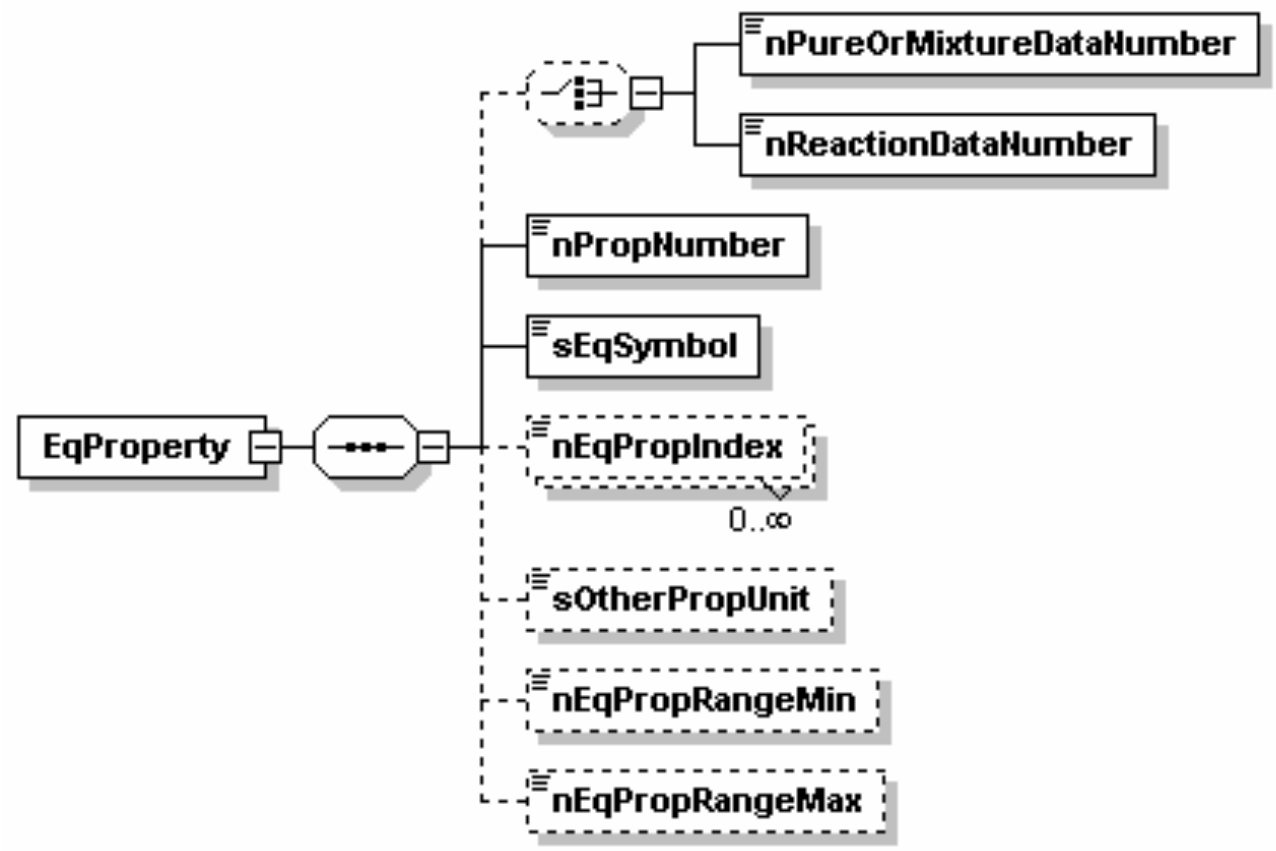

Fig. 24. Structure of the element **EqProperty** [complex] for equation representation. **EqProperty** is a subelement of **Equation** [complex] (Fig. 23).

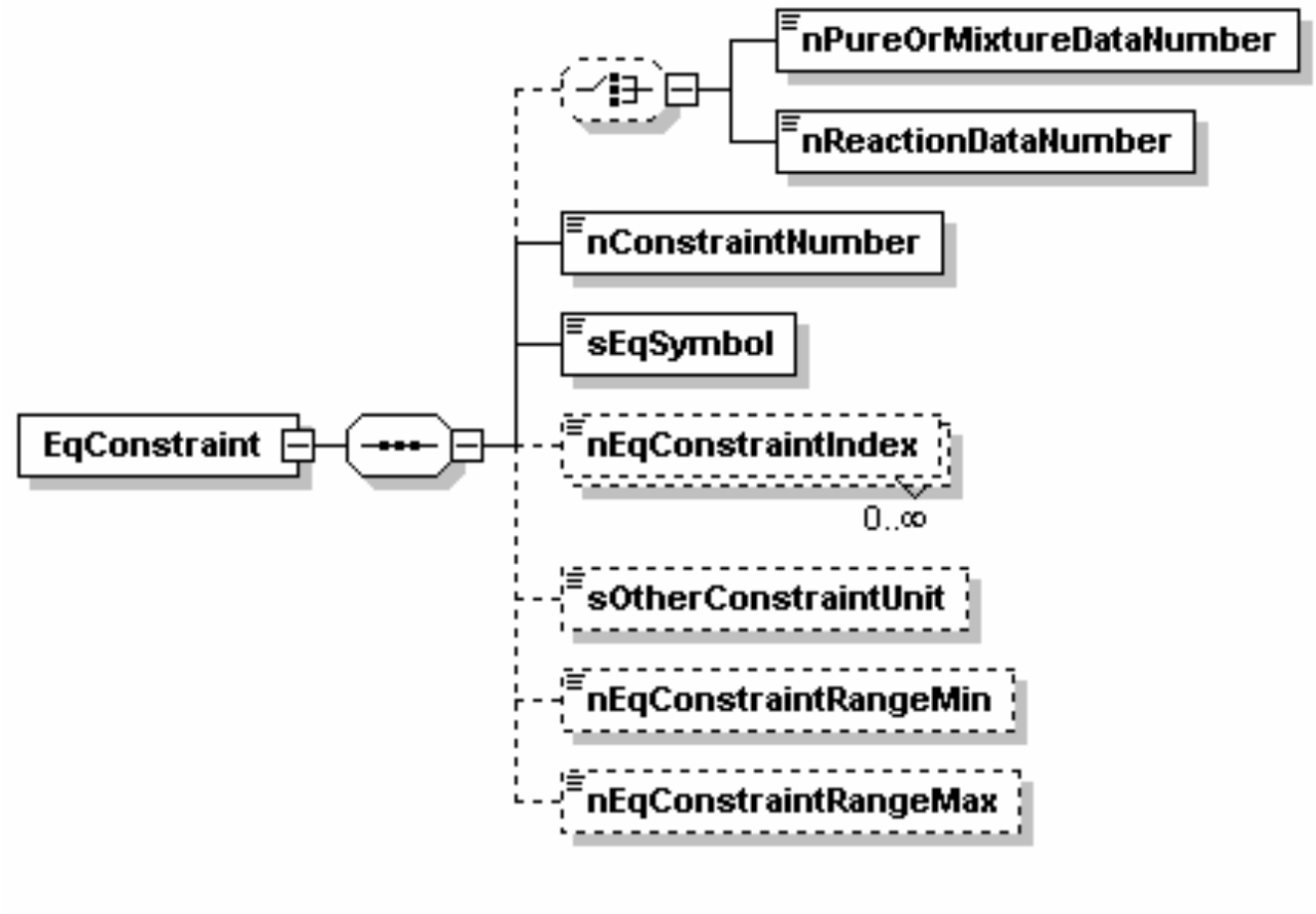

Fig. 25. Structure of the element **EqConstraint** for equation representation. **EqConstraint** is a subelement of **Equation** [complex] (Fig. 23).

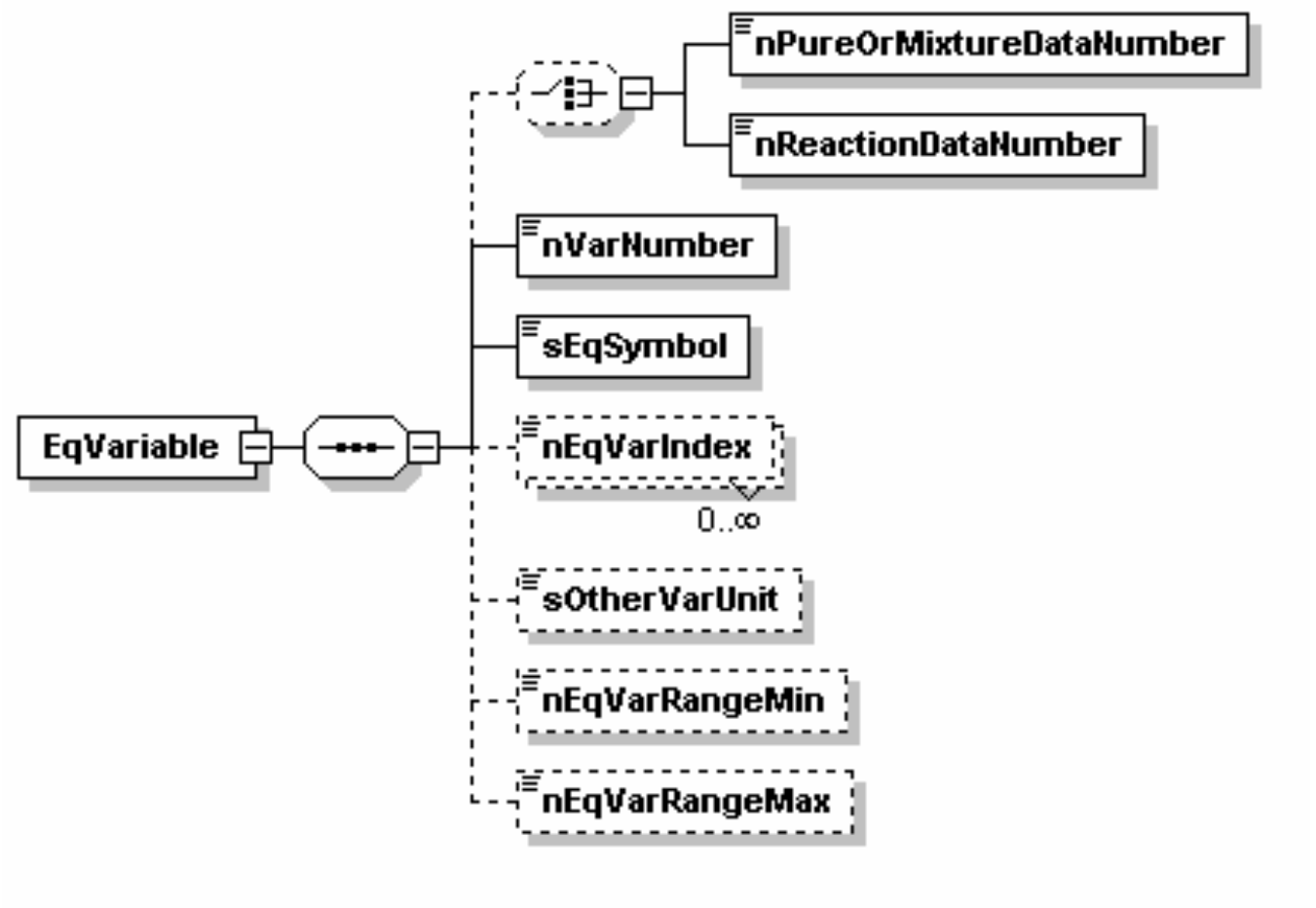

Fig. 26. Structure of the element **EqVariable** [complex] for equation representation. **EqVariable** is a subelement of **Equation** [complex] (Fig. 23).

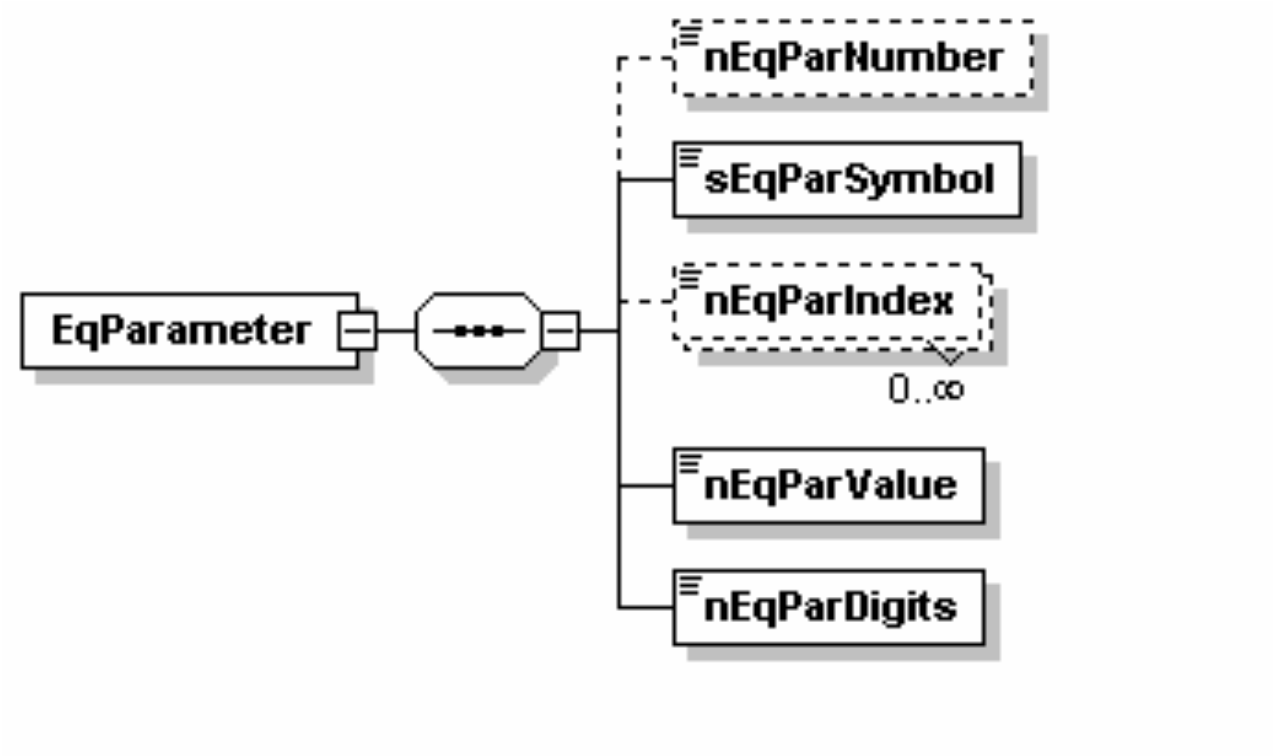

Fig 27. Structure of the element **EqParameter** [complex] for equation representation. **EqParameter** is a subelement of **Equation** [complex] (Fig. 23).

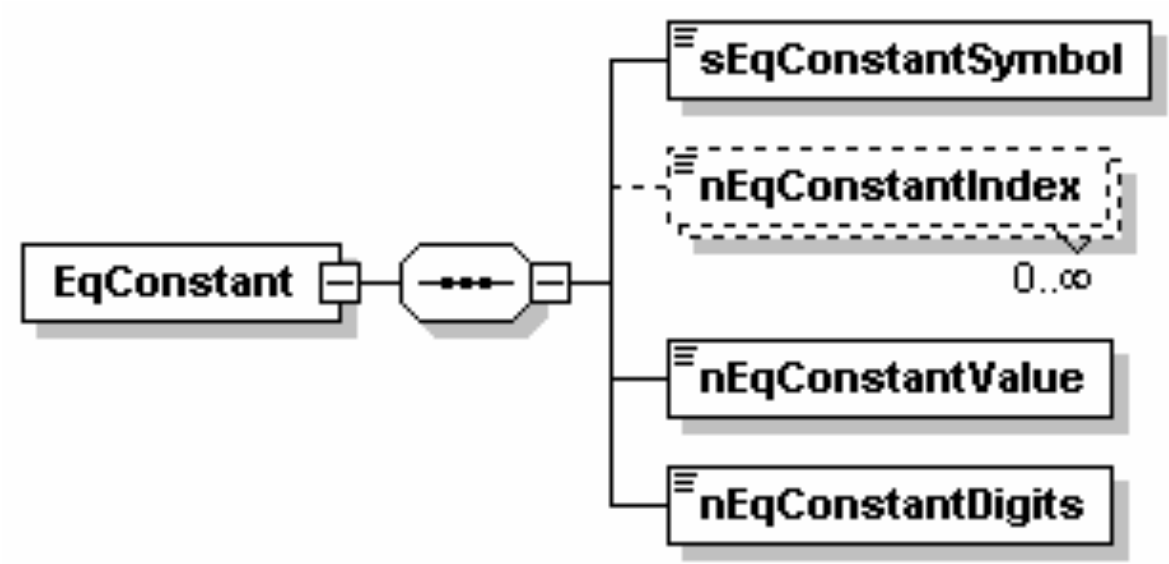

Fig. 28. Structure of the element **EqConstant** for equation representation. **EqConstant** is a subelement of **Equation** [complex]. (Fig. 23)

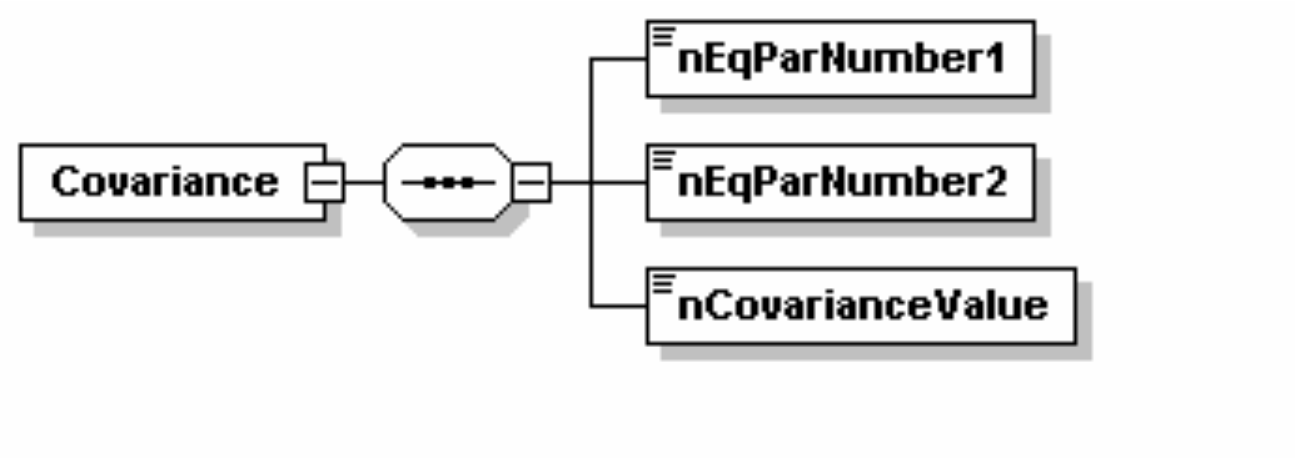

Fig. 29. Structure of the element **Covariance** for equation representation. **Covariance** is a subelement of **Equation** [complex]. (Fig. 23)

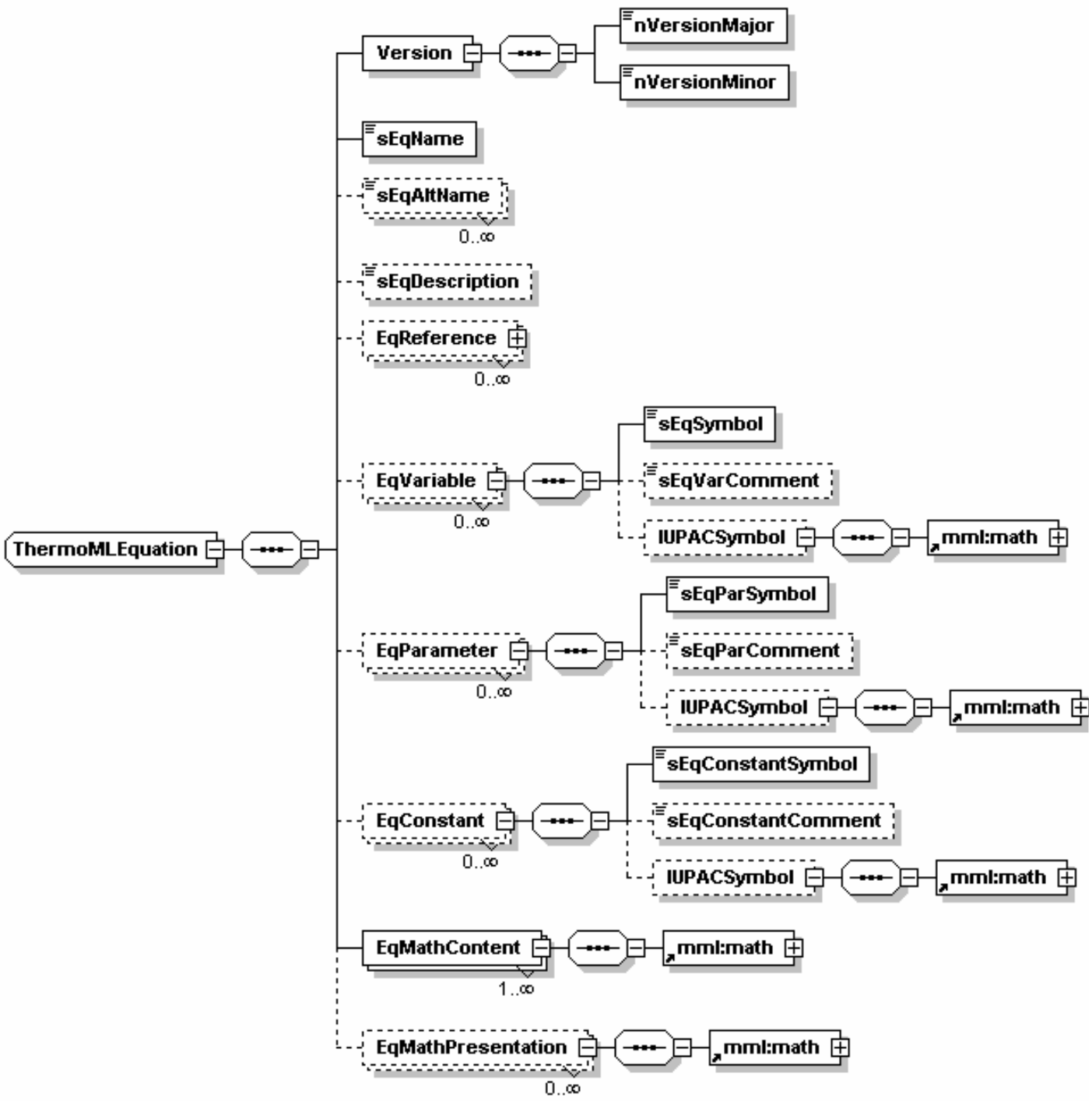

Fig. 30. Structure of the ThermoMLEquation schema for definition of equation representations.

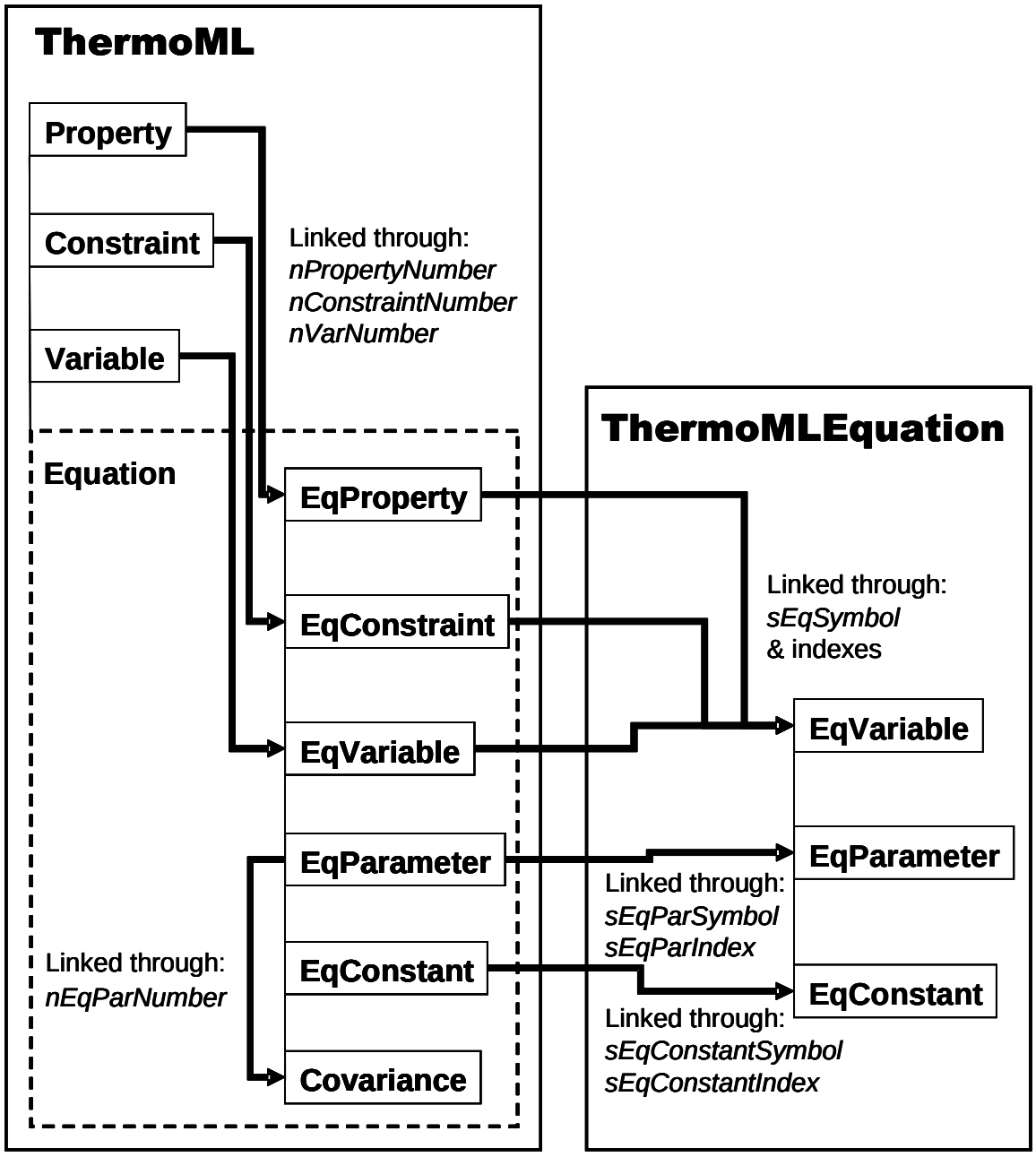

Fig. 31. Linking of *property*, *constraint*, *variable*, *parameter*, and *constant* information both within a ThermoML file and between a ThermoML and ThermoMLEquation file. Particular schema elements through which the linking is accomplished are indicated on the figure. Within ThermoML, the elements **Property** [complex], **Constraint** [complex], **Variable** [complex], and **Equation** [complex] (shown in the figure) are immediate sub-elements of the major blocks *PureOrMixtureData* and *ReactionData*. The dotted box encloses the sub-elements of **Equation** [complex] involved in linking of equation information.

87

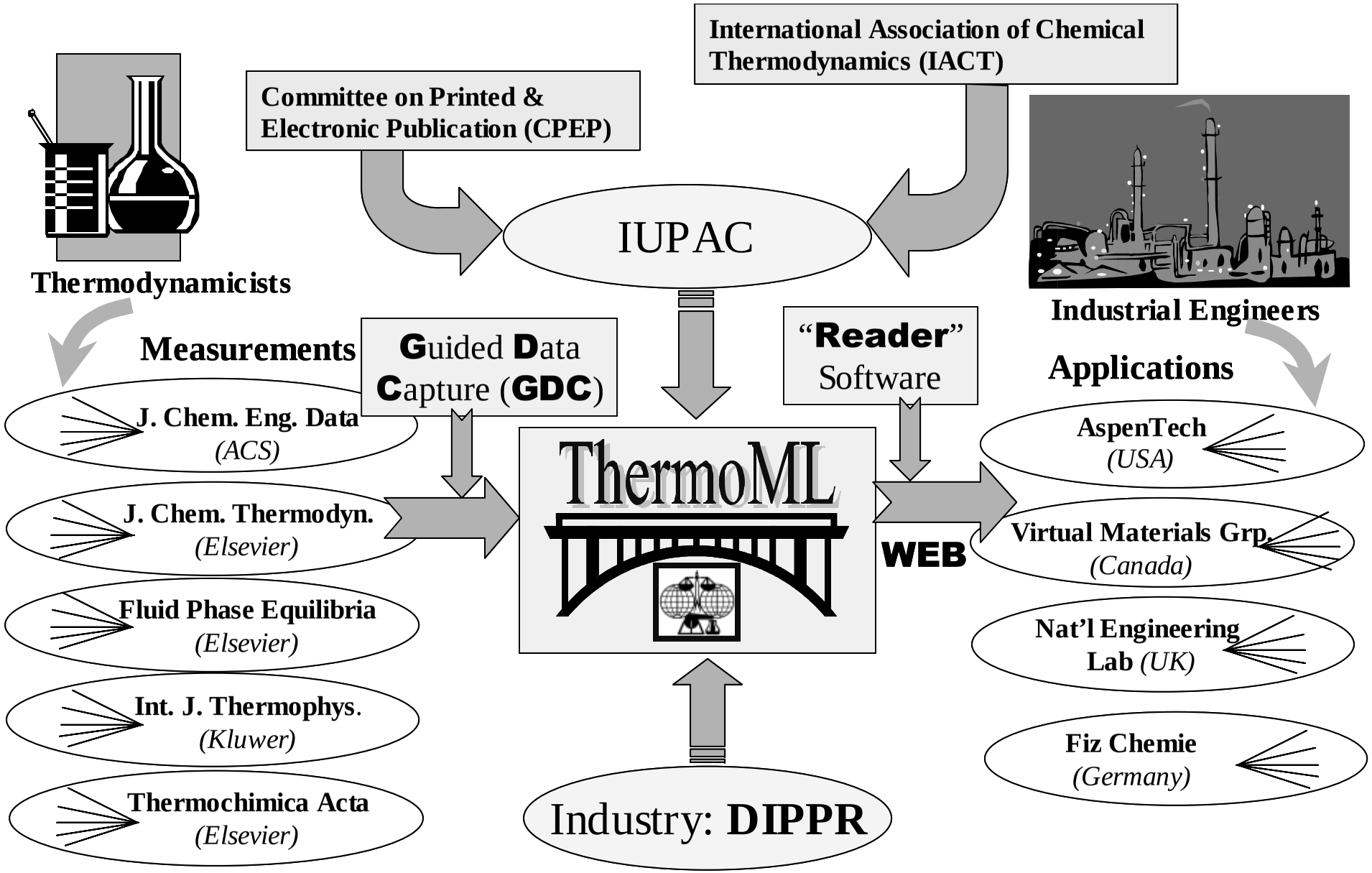

Fig. 32. The global thermodynamic data communication process between data providers (authors of journal articles) and data users (process simulation companies and industrial engineers) [46].