

# Embedded Code Processing ermöglicht die Einbindung Anwender-programmierter online und offline Analyseroutinen in eine AE-Standardsoftware

Hartmut VALLEN, Jochen VALLEN, Jens FORKER, Vallen-Systeme, Icking  
Tomoki SHIOTANI, Tobishima, Chiba, Japan

**Kurzfassung.** Obwohl heute sehr leistungsfähige und vielseitige Softwarepakete für die Analyse von Schallemissionsdaten, wie beispielsweise VisualAE™, verfügbar sind, erfordert die immer größer werdende Zahl unterschiedlichster Testobjekte die Nutzung applikationsspezifischer Analyseroutinen, insbesondere wenn eine spezielle Analyseverfahren auf eine große Zahl gleicher Objekte angewendet werden soll.

Manche Serviceanbieter haben aus umfangreichen Testdaten gleicher Prüfobjekte leistungsfähige Analyseverfahren entwickelt und möchten diese auch in Echtzeit nutzen.

Mithilfe des Embedded Code Processors (**ECP**) können nun Anwender selbst entwickelte Softwarealgorithmen in das Standardpaket (VisualAE) einbinden, testen und verifizieren. Jedes vom Standardpaket ermittelte Ergebnis (Hit Features, externe Parameter, Ortungs- bzw. Clusterergebnisse, Hintergrundgeräusch, usw.), kann in die eigene Routine eingelesen und zu einem oder mehreren neuen Ergebnissen verarbeitet und an nachfolgende Diagramme oder Prozessoren, z.B. den Alarmprozessor, ausgegeben werden.

Der Beitrag gibt eine Übersicht über die Struktur des Standardsoftwarepakets und zeigt beispielhaft einige Möglichkeiten des ECP auf.

## Grundlagen:

Bild 1 zeigt das grundsätzliche Datenflussschema zur Analyse von Schallemissionsdaten: Die Daten werden von einer Datenquelle, normalerweise eine Datei mit gemessenen Schallemissionsdaten, gelesen und Datensatz für Datensatz an verschiedene Verarbeitungsstufen weitergegeben. Eine Verarbeitungsstufe kann z.B. ein Filterprozessor sein, in dem bestimmte Daten ausgefiltert werden, oder ein Ortungsprozessor, der Ortungsergebnisse aufbereitet und weiterleitet, oder ein „Visual“, welches Ergebnisse grafisch oder numerisch visualisiert.

Der Anwender kann beliebige Verarbeitungsstufen einfügen und miteinander verknüpfen. So können sehr leistungsfähige und für spezifische Anwendungen angepasste Analysestrukturen realisiert werden.

Es sind eine Reihe von elementaren und komplexeren Datenprozessoren und Visuals verfügbar:

## Visuals

Ein Visual ist die letzte Verarbeitungsstufe eines Zweigs der Analysestruktur. Es dient zur Visualisierung aufbereiteter Ergebnisse in einer vom Anwender weitgehend selbst bestimmbarer Weise. Man unterscheidet folgende Visuals:

### *Korrelation, Punktdiagramm*

Zeigt Wertepaare (oder -triplets) als Punkt oder Symbol in einem 2- oder 3-dimensionalen Koordinatensystem. Für jede Achse kann der Anwender das darzustellende Ergebnis auswählen. Beispiel: Y-Ortung gegen X-Ortung oder Amplitude über der Zeit.

### *Liniendiagramme*

Wie ein Punktdiagramm, jedoch sind die Punkte über Linien in der Sequenz des Entstehens verbunden. Beispiel: Ereignissumme über externem Parameter.

### *Statistische Verteilung*

Zeigt die Zahl des Auftretens bestimmter Ergebnisse, gruppiert nach einem Wertintervall, differentiell (Werte pro Intervall, als Linie oder vertikale Balken) oder kumulativ (Summe über alle Wertintervalle von unten nach oben oder von oben nach unten). Auch Min, Max und Durchschnittswerte sind darstellbar.

### *Zeitverlauf*

Die Punkte zeitlich aufeinanderfolgender Messwerte werden mit einer Linie verbunden.

### *Numerisches Listing*

Jede Zeile steht für einen Datensatz. Die anzuzeigenden Messwerte bzw. Ergebnisse werden vom Nutzer ausgewählt und in Spalten dargestellt.

### *Wellenformdiagramm*

Zeigt einen beliebig wählbaren Ausschnitt einer Wellenform in Zeit- und/oder Frequenzdomäne.

### *Datenprozessoren*

Diese verarbeiten End- oder Zwischenergebnisse des aktuellen Datensatzes, fügen weitere Ergebnisse hinzu, filtern ggfs. Datensätze aus und geben die resultierenden Daten an die nächste Verarbeitungsstufe weiter. Es gibt folgende Prozessoren:

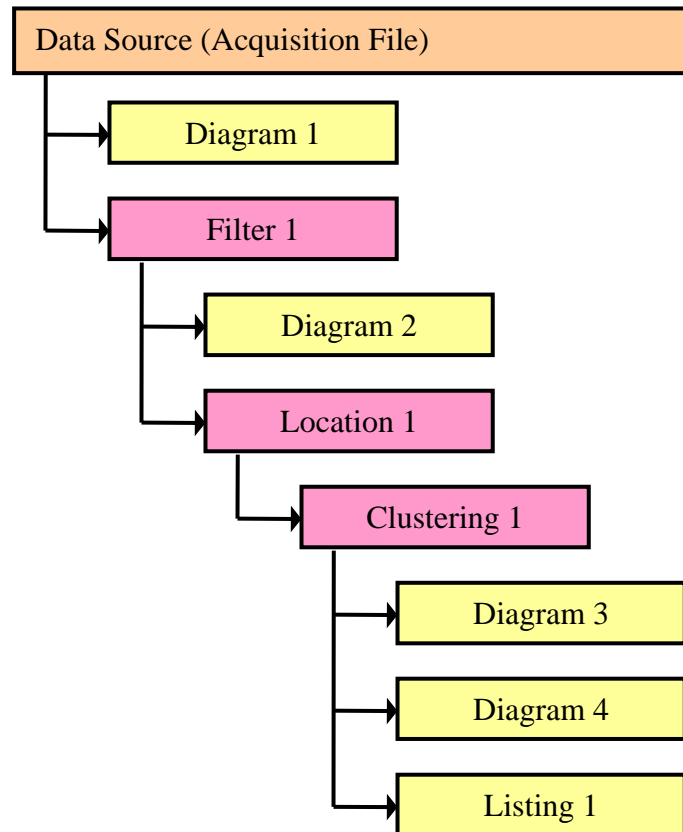


Bild 1: Prinzipielles Beispiel für ein Datenflussschema.

### *Filterprozessor*

Vergleicht Datensätze mit nutzerdefinierten Kriterien und lässt hiervon abhängig den AE-Datensatz durch oder hält ihn zurück.

### *Ortungsprozessor*

Softwarealgorithmus der aus bekannten Sensorpositionen und gemessenen Laufzeitunterschieden den Quellort und andere Ortungsergebnisse ausgibt.

### *Clusterprozessor*

Zeigt Datenkonzentrationen an wobei die Anzeigeschwelle durch den Nutzer definiert werden kann. Bsp: zeigt an ob pro vordefinierter Fläche eine gewisse Anzahl georteter Ereignisse erreicht oder überschritten wurde.

### *Polygonprozessor*

Indiziert Datensätze je nachdem ob diese in einem Punktdiagramm innerhalb oder ausserhalb anwenderbestimmter Polygonflächen in einem Punktdiagramm fallen.

### *Alarmprozessor*

Der Nutzer kann Warnungs- und Alarmkriterien definieren. Sobald diese Kriterien erreicht oder überschritten werden, führt der Alarmprozessor eine Warnungs- bzw. eine Alarmaktion aus. Als Aktionen können z.B. Emails versandt werden, WAV-Dateien abgespielt oder die externen Alarm- und Warning-Leitung und -LED aktiviert werden.

### *Grading Prozessor*

Berechnet Severity Index und Historic Index nach Timothy Fowler's Gleichungen [6].

### *User Prozessor*

Bietet eine Reihe statistischer und arithmetischer Funktionen an, um AE-Daten mit sich oder anderen AE-Daten mathematisch zu verknüpfen. Beispiel:  $A * X + B$ , Amplituden-Distanz-Korrektur, Energieraten, u. a..

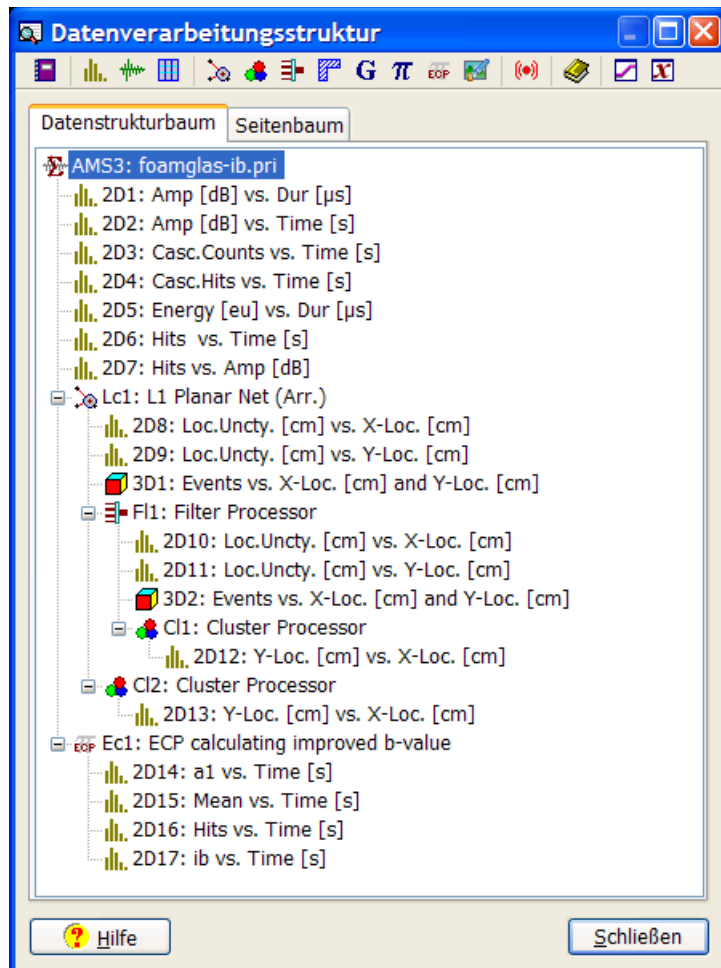


Bild 2: Beispiel eines realen Datenflusseschemas

### *Embedded Code Processor (ECP).*

Ein Prozessor, der Anwender-programmierten Code verarbeitet. Der Code wird beim Laden der VAE-Einstellungen kompiliert und in den Code von VisualAE eingebettet.

## Potentielle Anwendungen des ECP

Der ECP wurde entwickelt um beispielsweise folgende Probleme individuell, flexibel und anwendungsgerecht lösen zu können:

- Implementierung von öffentlich zugänglichen Analyseprozeduren, z.B. die französische AFIAP Richtlinie sowie proprietären Analyseprozeduren, z.B. „CEF“ des TUEV Austria für kleine Flüssiggasbehälter (“CEF” steht für “Cluster Evaluation Factor”).
- Ermittlung proprietärer oder öffentlich bekannter Kennwerte, wie z.B. Felicity Ratio [1], LoadRatio [2], Calm Ratio [2], improved B-Value [3-5], etc..
- Automatische Überwachung eines Tests auf Test-Halt-Kriterien.
- Automatische Ermittlung der Stadien: “Loading”, “Holding the load”, “unloading”.
- Intelligente Überwachung des Hintergrundgeräuschs mit entsprechender Alarmierung.
- Cluster- oder Polygon-spezifische Datenverarbeitung.
- Glättung parametrischer Daten.
- Auswertung von Zeitserien.
- Kanalspezifische Ankunftszeitkorrektur (falls Wellenleiter unterschiedlicher Länge verwendet werden).
- Anwender-programmierte Ortungsalgorithmen.

## ECP Programmiersprache

ECP nutzt eine leistungsfähige Programmiersprache, die eine einfache prozedurale Syntax (ähnlich zu Pascal) mit leistungsfähigen Datenbeschreibungskonstrukten kombiniert.

Der Code wird von der VisualAE-Einstellungsdatei (\*.VAE) gelesen und in einen Zwischencode kompiliert, der während der Ausführung interpretiert wird. Der Nutzer benötigt keine separate Entwicklungsumgebung und hat dennoch die volle Flexibilität um eigenes Know-How zu implementieren.

## Anwendungsbeispiel „Improved B-Value“ (ib-value)

Prinzip und Anwendung des *ib-values* wurde u. a. in [3] bis [5] beschrieben. Im Prinzip ist der *ib-value* die Steigung der kumulativen Amplitudenverteilung von hohen zu niedrigen Amplituden zwischen zwei Amplituden-Eckpunkten ( $a_1$ ,  $a_2$ ).

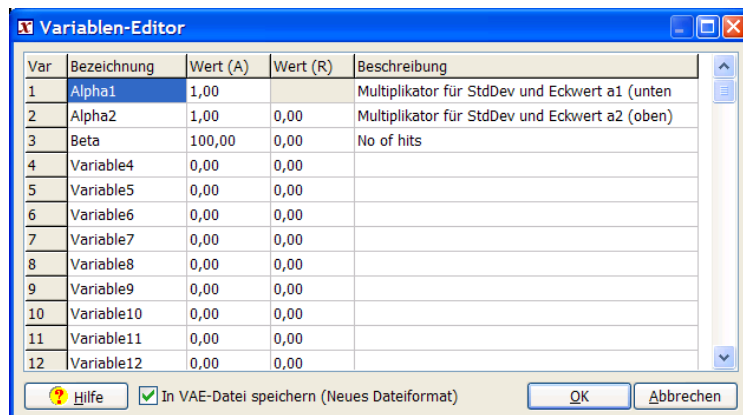


Bild 3 Variableneditor dient zur Parametereingabe f Alpha und Beta

Die Eckpunkte werden dynamisch aus dem Mittelwert „Mean“ und der Standardabweichung „StdDev“ der letzten „Beta“ Amplituden in dB ermittelt. Für die Eckwerte ergibt sich dann:

$$a_1 = \text{Mean} - \text{StdDev} * \text{Alpha}_1$$
$$a_2 = \text{Mean} + \text{StdDev} * \text{Alpha}_2.$$

Alpha1, Alpha2 und die Fensterlänge Beta sind

Konstanten mit denen der Algorithmus materialspezifisch optimiert werden kann.

Dieses Beispiel erfordert somit die Vorgabe der drei Parameter Alpha1, Alpha2 und Beta, das Einlesen der Amplituden, und die Berechnung und Ausgabe von *ib*-value, *a1*, *a2*, Mean, und StdDev.

### Eingabe von Parametern

Der Variableneditor in VisualAE (Bild 3) dient zur zentralen Eingabe von Variablen, die z.B. in unterschiedlichen Filterprozessoren als Kriterium verwendet werden können. Im ECP kann z.B. der Wert mit der Bezeichnung Alpha1 in Bild 3 wie folgt in die lokale Variable A11 übernommen werden:

„A11 = VAEVariables.Alpha1“

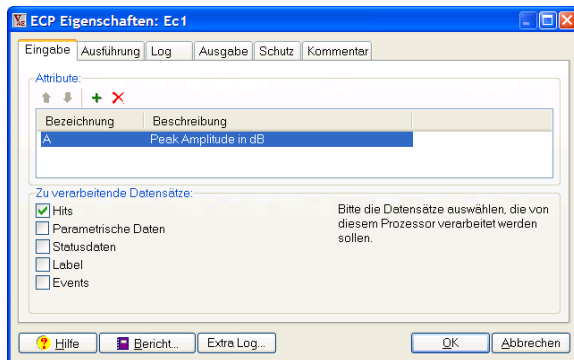


Bild 4 Definition der Eingänge und der zu verarbeitenden Datensatzarten

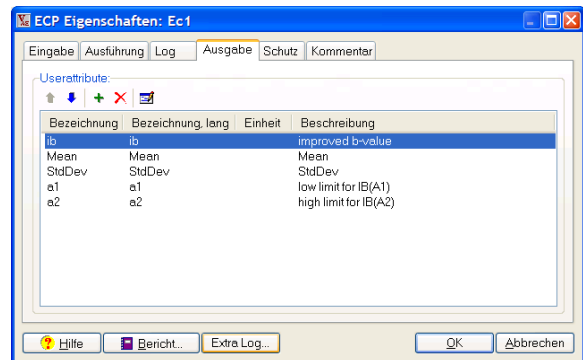


Bild 5 Definition der Ergebnisausgabe

### Eingabe der Amplitude und Ausgabe der ECP Ergebnisse

Das Menü in Bild 4 dient zur Auswahl der Eingabevariablen. So wie man im Listing ein weiteres Resultat in die Listingzeile hinzufügt, kann man beim ECP einen Eingang hinzufügen.

Bild 5 zeigt wie der Programmierer die an VisualAE zurückzugebenden neuen (vom ECP berechneten) Ergebnisse definiert. Diese Ergebnisse können von nachgeschalteten Prozessoren oder Visuals weiter verarbeitet werden.

Bild 6 ist ein Ausschnitt aus dem Code zur Berechnung des *ib*-values.

Bild 7 ist ein Beispiel einer VisualAE-Ergebnisgrafik mit Diagrammen die auch den Verlauf der Resultate Mean, StdDev, *a1*, *a2* und *ib* anzeigen.

### Zusammenfassung

Mit Embedded Code Processing (ECP) kann ein Standardsoftwarepaket um praktisch unbegrenzte Funktionsmöglichkeiten erwei-

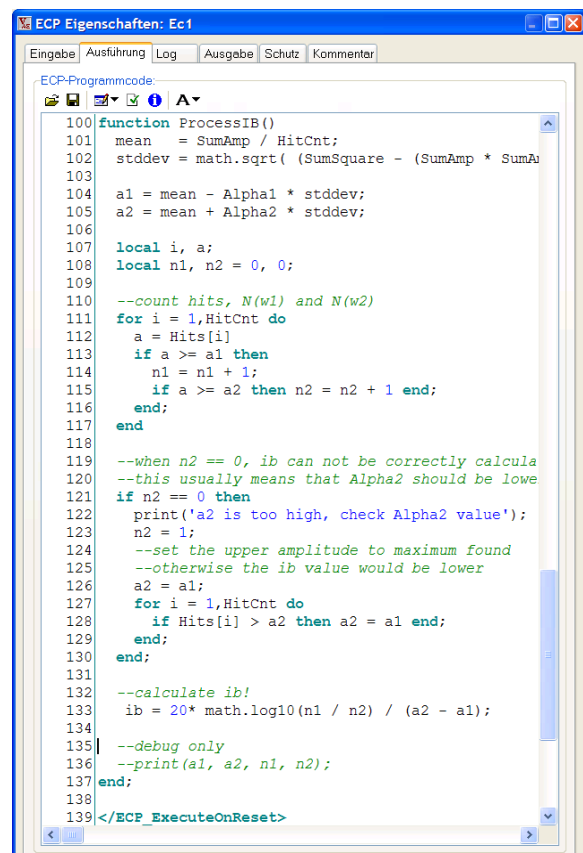


Bild 6 Beispielcode

tert werden, die sowohl online als auch offline zur Verfügung stehen. Es können beliebige Resultate in beliebiger Komplexität zu neuen Resultaten verrechnet und anschließend in VisualAE wie die bekannten Resultate angezeigt oder weiter verarbeitet werden. Solche Funktionserweiterungen können vom Anwender ohne Mitwirkung des Herstellers durchgeführt werden. Nicht gezeigt wurden in diesem Beitrag die Möglichkeiten den Code für eine Lizenzierung an Andere vor Einsicht und Veränderung zu schützen.

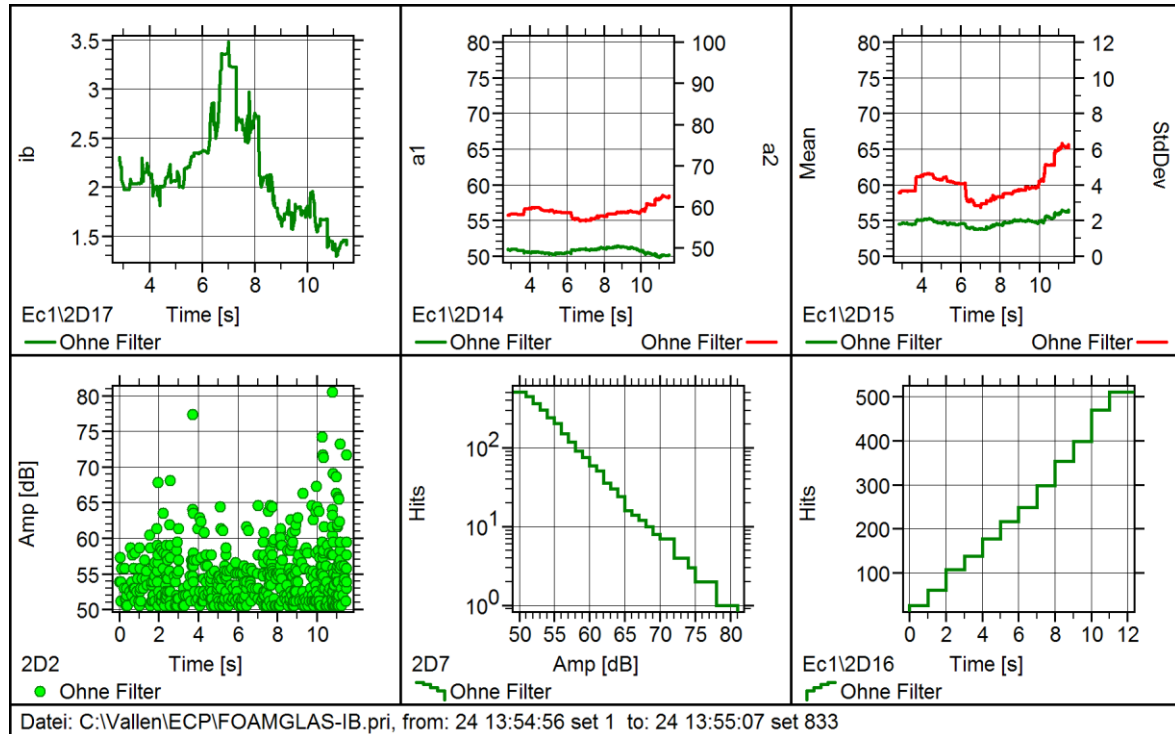


Bild 7 Beispiel einer VisualAE Grafik zur Darstellung von normalen und von ECP-Ergebnissen

## Referenzen

- [1] Fowler, T.J. Experience with acoustic emission monitoring of chemical process industry vessels. Progress in Acoustic Emission III, JSNDI, pp. 150-162, 1986
- [2] The Japanese Society for Non-Destructive Inspection: Recommended practice for in situ monitoring of concrete structures by acoustic emission, NDIS2421, p. 24, 2000.
- [3] Shiotani T., Fujii K., Aoki T., Amou K., “Evaluation of progressive Failure using AE Sources and Improved b-value on Slope Model Test, “Progress in Acoustic Emission VII, eds Kishi T., Mori Y., Enoki M., JSNDI Tokyo, pp529-534, 1994
- [4] Shiotani, T., Yuyama S., Li Z.W., Ohtsu M., “Application of an Improved b-value to quantitative evaluation of fracture process in concrete material”, Journal of AE Vol. 19, pp 118-133, 2001
- [5] Shiotani T., Nakanishi Y., Luo X., Haya H. “Damage Assessment in railway sub-structures deteriorated using AE technique”. Proceedings EWGAE 2004, lecture 21(CD), pp 255-266 (print).
- [6] Recommended Practice for Acoustic Emission Evaluation of Fibre Reinforced Plastic (FRP) Tanks and Pressure Vessels, Draft I Oct 1999, CARP, p 12-8+