

Boymann, Simon; Maschotta, Ralph; Steuer, Dunja:

**Methodik der softwaretechnischen Realisierung eines komplexen
Biosignalverarbeitungstools**

<i>Zuerst erschienen in:</i>	Biomedizinische Technik = Biomedical Engineering. - Berlin [u.a.] : de Gruyter. - 46 (2001), S1, S. 330-331. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik (DGBM) im VDE ; 35 (Bochum) : 2001.09.19-21
<i>Erstveröffentlichung:</i>	2001
<i>Datum Digitalisierung:</i>	2009-08-06
<i>ISSN (online):</i>	1862-278X
<i>ISSN(print)</i>	0013-5585
<i>DOI:</i>	10.1515/bmte.2001.46.s1.330
<i>[Zuletzt gesehen:</i>	2019-12-06]

„Im Rahmen der hochschulweiten Open-Access-Strategie für die Zweitveröffentlichung identifiziert durch die Universitätsbibliothek Ilmenau.“

“Within the academic Open Access Strategy identified for deposition by Ilmenau University Library.”

„Dieser Beitrag ist mit Zustimmung des Rechteinhabers aufgrund einer (DFG-geförderten) Allianz- bzw. Nationallizenz frei zugänglich.“

„This publication is with permission of the rights owner freely accessible due to an Alliance licence and a national licence (funded by the DFG, German Research Foundation) respectively.“



METHODIK DER SOFTWARETECHNISCHEN REALISIERUNG EINES KOMPLEXEN BIOSIGNALVERARBEITUNGSTOOLS

S.Boymann, R.Maschotta, D. Steuer

Institut für Biomedizinische Technik und Informatik, TU Ilmenau

boymann@informatik.tu-ilmenau.de

EINFÜHRUNG

Auf dem Gebiet der Signalanalyse existiert durch langjährige Forschung ein großer Fundus an Algorithmen, die in verschiedenen Programmiersprachen und mit unterschiedlichsten Techniken des Softwareentwurfs realisiert wurden. Eine Wiederverwendung dieser Algorithmen erweist sich deshalb oftmals als schwierig. Durch die komplexen Anforderungen an Signalanalyseverfahren besteht heute ein großer Bedarf an universellen Entwicklungssystemen. Deshalb wird nach Wegen gesucht, diesen Bedarf zu befriedigen und neue Analysestrategien zu entwickeln.

Zusammen mit Partnern aus der Industrie wurden innerhalb eines Projektes¹ die Grundlagen für ein komponentenbasiertes Analysetool mit modernsten Methoden des Softwareentwurfs erstellt. Ein Schwerpunkt dabei ist die Implementierung von komplexen adaptiv rekursiven Schätzfunktionen und deren Validierung in der Praxis.

MATERIALIEN UND METHODEN

Die aus der Aufgabenstellung abgeleiteten Anforderungen an ein komponentenbasiertes Analysetool sind, dass es (1) eine Bibliothek elementarer Signalverarbeitungsalgorithmen besitzt, (2) über einen Mechanismus verfügt, diese Algorithmen effektiv miteinander zu kombinieren, (3) definierte Schnittstellen zur Verfügung stellt, um neue Berechnungsvorschriften in die Bibliothek aufzunehmen, (4) möglich ist, externe Strukturen (z.B. MATLAB®-Funktionalität) zu integrieren.

diesem Zweck wurde die Softwarearchitektur mit Hilfe objektorientierter Modelle in UML-Notation erstellt [3]. Neben der Softwarearchitektur lassen sich so auch die natürlichen Beziehungen der einzelnen Algorithmen untereinander abbilden.

Neben der Implementierung klassischer Verfahren der Signalverarbeitung ist ein wichtiger Aspekt die Realisierung einer Bibliothek elementarer adaptiver Schätzverfahren. Damit stehen für Signale, die sich als Realisierung stationärer Prozesse modellieren lassen, Methoden der Zeitreihenanalyse mit Schätzfunktionen im Zeit- und Spektralbereich zur Verfügung. Auf Grund ihrer Struktur sind sie besonders zur ereignisnahen Analyse geeignet. In Verbindung mit dem Signalanalysetool lassen sich diese Grundalgorithmen zu komplexen Verfahren wie z.B. die adapt. Varianz oder eine adapt. Trendermittlung, bzw. Vektor- und Matrixanalyse kombinieren.

Um verschiedenste Berechnungsalgorithmen miteinander kombinieren zu können, wurde das System nach dem Tee-and-Join-Pipeline-System, einer Variante des Pipes and Filters Architekturmuster konzipiert [1]. Die einzelnen Berechnungsalgorithmen werden dabei als Filter implementiert, die durch Pipes miteinander verbunden werden. Ein Filter ist maximal mit einer Ausgangspipe verbunden, diese Pipe kann dann mit mehreren Filtern verbunden werden. So werden Verzweigungen und Rückkopplungen und damit der Aufbau einer komplexen Verarbeitungspipeline aus verschiedenen Berechnungsalgorithmen möglich (Abb. 1). Globale Informationen wie Laufzeitinformationen oder Fehlermeldungen werden in globalen Objekten gehalten.

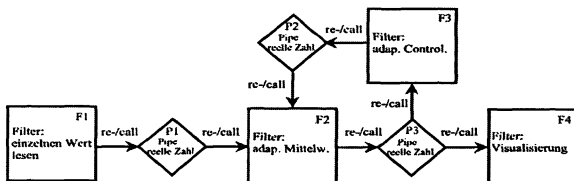


Abb. 1: Beispiel einer Pipeline

Die Methodik der softwaretechnischen Realisierung des entwickelten Systems besteht in einer modernen objektorientierten Architektur unter Verwendung verbreiteter Architektur- und Entwurfsmuster[1][2]. Zu

Die Pipelines sind so aufgebaut, dass verschiedene Filter miteinander verbunden werden können. Die Kombinationsmöglichkeit zweier Filter wird hauptsächlich durch den Datentyp der Ein- bzw. Ausgänge be-

stimmt. Diese werden durch ein Zahlenformat (Ganze Zahl, reelle Zahl, kompl. Zahl, u.s.w) und eine bestimmte Größe beschrieben. Die Größe gibt an ob der Filter für die Onlineanalyse implementiert wurde (Einzelwert) oder für die Offlineanalyse, wobei die Anzahl der in einem Schritt berechneten Werte vom implementierten Algorithmus abhängt.

ERGEBNISSE

Sowohl die objektorientierte Realisierung der Algorithmen, als auch die Konzeption des Entwicklungssystems unterstützen die Verknüpfung elementarer Berechnungsvorschriften zu komplexen Signalverarbeitungsstrategien.

Nach dem jetzigen Stand der Entwicklung wurden die Grundlagen für ein modulares, erweiterbares Signal- und Bildverarbeitungstool entwickelt, das den aktuellen Qualitätsanforderungen an Software entspricht.

Bei der Verarbeitung von Biosignalen, bei denen ereignisnah (Monitoring) oder offline, oft mehrere Kenngrößen berechnet werden, lässt sich die Komplexität des Zusammenspiels mehrerer verschiedener Signalverarbeitungsalgorithmen zeigen. Eine Beispiel ist die Detektion von QRS-Komplexen mit adapt. Verfahren zur Ermittlung der Herzfrequenz bzw. daraus abgeleitet eines Tachogramms [4]. Dazu wird ein adapt. Mittelwert, eine adapt. Autokorrelation, ein adapt. α -Quantilschätzer und Komparator miteinander kombiniert. Das Ergebnis der Autokorrelation kann weiter verwendet werden um das Periodogramm des EKG-Signals zu berechnen.

Eine Pipeline wird in einer adäquaten Form beschrieben und von einem geeigneten Parser umgesetzt. Die Beschreibungsform ist der Problemdomäne angepasst. Komplexe adapt. Schätzfunktionen lassen sich z.B. durch arithmetische Verknüpfung elementarer adapt. Elemente erzeugen. Deshalb wurde ein Compiler implementiert, der eine arithmetische Nachricht bearbeitet und daraus eine Pipeline erzeugt.

DISKUSSION

Die Verwendung des Pipes and Filters Musters hat mehrere Vorteile: Flexibilität der Berechnungspipeline durch Austausch oder Neuordnung der Filter, Wiederverwendung bzw. einfache Integrierung bereits implementierter Algorithmen, rapid Prototyping von Pipelines, einfache Speicherungsmöglichkeit von Zwischendaten und eventuelle Vorteile durch parallele Berechnung von Filtern. Auf der anderen Seite ist die Nutzung gemeinsamer Daten und die Fehlerbehandlung oft problematisch. Auch die Laufzeit und Speicherkosten sind aufgrund häufiger Datenübermittlung zwischen den Filtern und mehrfachen Methodenaufrufen relativ hoch.[1]

Je nachdem, welche Filter mit welcher Granularität verwendet werden ändert sich auch das Laufzeitverhalten des Systems. Verwendet man nur Filter, die Grundrechenoperationen implementieren oder Algorithmen die nur einzelne Werte berechnen, erhöht sich der Ver-

waltungsaufwand für die Kommunikation zwischen den Filtern. Man erreicht jedoch eine sehr hohe Flexibilität der Berechnungspipeline. Es ist möglich die Funktionsweise oder auch die Kombination verschiedener Analysestrategien zu testen. Nach erfolgreichem Test kann dieser als eigenständiger Filter implementiert werden und so der Aufwand für die Kopplung der Filter eingespart werden.

Eine Möglichkeit, den Funktionsumfang des Analysetools effizient zu erweitern, ist das Einbinden externer Komponenten. Die dafür notwendigen Schnittstellen können mit einem Minimum an Aufwand realisiert werden. So kann die MATLAB® - Funktionalität z.B. zur Visualisierung eingebunden werden.

Ein Nachteil der komplexen Architektur ist, dass eine Pipeline nicht ohne erheblichen Aufwand in einen, zur Verwendung in Mikrokontrollern geeigneten Code umgewandelt werden kann.

Bei der Implementierung neuer Algorithmen ist es notwendig, von den entsprechenden Basisklassen zu erben und vorgegebene abstrakte Methoden der Schnittstelle zu überschreiben. Auch beim Zugriff auf die Eingabedaten und der Speicherung der Ergebnisdaten müssen Restriktionen der Softwarearchitektur eingehalten werden.

ZUSAMMENFASSUNG

Perspektivisch soll, aufbauend auf den Grundlagen ein Signalanalysetool für den Einsatz in der Forschung als effektive Entwicklungsumgebung für neue Signalanalysestrategien, als auch für den Einsatz in der Industrie zur Bearbeitung der täglichen Routineaufgaben entwickelt werden.

LITERATURHINWEISE

- [1] F. Buschmann u.a. "Pattern orientierte Softwarearchitektur" Addison Wesley 1998
- [2] E. Gamma u.a. „Entwurfsmuster“ Addison Wesley 1996
- [3] Object Management Group „Unified Modeling Language“ <http://www.omg.org/>
- [4] D. Steuer „Nichtlineare Analyse und Klassifikation von instationären Biosignalen mit Anwendung in der Kognitionsforschung.“ Dissertation, TU Ilmenau, Fakultät für Informatik und Automatisierung, 2001.

¹ Projekt „ATISA“, gefördert durch das Thüringer Ministeriums für Wissenschaft, Forschung und Kultur (TMWFK: B699-00011)