

## Neue Entwicklungsmethode "SPV" für Software- und Hardware-Systeme

### Kurzbeschreibung

Stand: Mai 2011

In 15-jähriger Entwicklung entstand ein neues, hochleistungsfähiges Entwicklungsverfahren, Spezifikationsverfahren **"SPV"** genannt, mit dem die Entwicklung von Software- und Hardware-Systemen von derzeit drei Stufen, die der Spezifikation, der Implementierung und der Verifikation, auf eine Stufe, die der Spezifikation reduziert wurde. Aufgrund einfacher, übersichtlicher textueller Strukturen ist die Spezifikation zweifelsfrei lesbar, reviewbar bzw. diskutierbar und damit verifizierbar und kann auch von Personen die nicht an der Entwicklung beteiligt sind, schnell und exakt interpretiert werden.

Die SPV-Entwicklungs-Methodik ist für alle Entwicklungs-Vorhaben/-Bereiche, wie allgemeine Software- und Datenbank-Entwicklung, Software-/Hardware-Systeme (Embedded Systems), in Bereichen Organisation, Kommunikation, Automotive Anwendungen, Luft- und Raumfahrt, Medizin-Technik, etc., optimal geeignet.

Das SPV basiert auf einer rein funktionalen Entwicklungs-Methodik, d. h., das spezifizierte System ist Software-/Hardware-neutral und wird erst für eine Realisierung in Software- und/oder Hardware-Komplexe determiniert.

Für eine standardmäßige Weiterverarbeitung erfolgt über den SPV-Zwischen-Compiler die Umsetzung der Spezifikations-Datenbasis in Hochsprachen für den Software-Bereich in "C", "C++", etc., für den Hardware-Bereich in "VHDL", "VERILOG", etc..

Für den Hardware-Bereich wird dabei automatisch die Ablaufsteuerung erzeugt.

Durch völlig neue Eigenschaften und Automatismen der SPV-Entwicklungs-Methodik (DPA-/PCT-Anmeldung) wird der Entwickler optimal entlastet, der Entwicklungs-Aufwand mehr als halbiert und eine fehlerfreie Umsetzung der Spezifikations-Datenbasis in Hochsprachen ("C", "C++", etc., bzw. "VHDL", "VERILOG", etc.) garantiert. Durch eine weitreichende, kontrollierte Determinierbarkeit eines Gesamtentwicklungskomplexes können extrem kurze Entwicklungs-Zeiten erreicht werden.

### Die wesentlichsten funktionellen Eigenschaften bzw. Vorteile des "SPV" gegenüber dem Stand der Technik sind:

- Einheitliche Methode für alle Entwicklungs-Vorhaben/-Bereiche.
- Nur eine Entwicklungs-Ebene, die der Spezifikation, sie ist bereits die Implementierung.
- Einheitliche Entwicklungs-Datenbasis (Spezifikation), Software-/Hardware-neutral, mit beliebiger Aufteilbarkeit in Software- und/oder Hardware-Bereiche.

- Automatische Zuordnung von seriellen und parallelen Datenverarbeitungsschritten unabhängig von der Reihenfolge der Eingabe von Teilspezifikationen.
- Konsistenz-Überwachung von Spezifikations-Änderungen.
- Einfache Erstellung, Erweiterung bzw. Reduzierung von Architekturen (Toplevel, Schnittstellen, Komponenten) durch automatische Pfad-Zuweisung (Level-/Komponenten-Nummern).
- Aufgrund einfacher, übersichtlicher textueller Strukturen ist die Spezifikation zweifelsfrei lesbar, reviewbar bzw. diskutierbar und damit verifizierbar und kann auch von Personen die nicht an der Entwicklung beteiligt sind, schnell und exakt interpretiert werden.
- Garantiert fehlerfreie Umsetzung der SPV-Spezifikations-Datenbasis in Hochsprachen ("C", "C++", etc., bzw. "VHDL", "VERILOG", etc.) durch den SPV-Zwischen-Compiler für eine im Toolflow standardisierte Weiterverarbeitung. Eine Verifikation ab Hochsprache ist nicht erforderlich.
- Das SPV-Zwischen-Compiling basiert auf wenigen Parametern, die vom SPV automatisch erzeugt werden. Deshalb ist die SPV-Zwischen-Compiler-Entwicklung einfach und zu 100% verifizierbar.
- Durch Bereitstellung komplexer Methoden/Elemente bzw. komplexer Komponenten können komplexe Systeme auf hohem Level schnell entwickelt werden.
- Bei der Spezifizierung (=Implementierung) werden den Methoden/Elementen nur Daten und die Funktion für die Datenverarbeitung zugewiesen, die interne Funktions-Beschreibung der Methoden/Elemente wird dabei nicht benötigt.
- Durch die Zusammenführung von Spezifikation und Implementierung sowie dem Wegfall der Verifikation wird der System-Entwicklungs-Aufwand mehr als halbiert.
- Kurze System-Entwicklungszeiten durch eine optimale Aufteilbarkeit eines Gesamtentwicklungskomplexes in Einzelentwicklungskomplexe.  
Automatische Konsistenz-Prüfung bei nachfolgender Integration.

Mit dem "SPV" steht ein Entwicklungsverfahren zur Verfügung, in dem die Implementierung für ein Software- und Hardware-System ausschließlich durch die Spezifizierung des Operatings und der Komponenten-Datenverarbeitung erfolgt.

Damit sind Inkonsistenzen und Fehler bei der Implementierung durch falsche Interpretation der Spezifikation ausgeschlossen.

Für die Überprüfung bzw. Diskussion und Review der Spezifikation wurden im SPV übersichtliche textuelle Strukturen geschaffen, so dass die Entwicklungsergebnisse auch von Personen die nicht an der Entwicklung beteiligt sind, schnell und exakt interpretiert werden können.

Die SPV-Entwicklungsmethode basiert auf einer neuartigen Organisation des Operatings in Verbindung mit der automatischen Generierung neuartiger Parameter bei der Spezifizierung der Komponenten-Datenverarbeitung.

Hierdurch ist es möglich, für jede den Datenverarbeitungsablauf einer Komponenten-Operation steuernde Kriterien-Kombination, für eine Basis-Operation bestimmte Basis-Operations-Varianten (1, 2, 3, ..) zu spezifizieren.

Die Spezifizierung einer Basis-Operations-Variante erfolgt durch entsprechende Änderung von Datenverarbeitungs-Knoten einer Referenz-Operation, beispielsweise der Basis-Operation oder einer bereits bestehenden Basis-Operations-Variante. Dabei wird bei jeder Änderung eines Datenverarbeitungs-Knotens auf Konsistenz nachfolgender Datenverarbeitungsschritte geprüft und im Korrekturfall die entsprechend zu korrigierenden Datenverarbeitungs-Knoten angezeigt, so dass mit Abschluss der Spezifizierung einer Basis-Operations-Variante alle operativ beteiligten Datenverarbeitungs-Knoten fehlerfrei sind.

Bisher mussten die den Datenverarbeitungsablauf einer Komponenten-Operation steuernden Kriterien-Kombinationen jedem Datenverarbeitungs-Knoten mit "If-Then-Else-Konstrukten" zugeordnet werden, das bei umfangreichen Datenverarbeitungs-Abläufen und einer hohen Anzahl von Kriterien-Kombinationen zu Implementierungsfehlern führt und auch die Überprüfung der Datenverarbeitungsabläufe auf Richtigkeit im Sinne der Aufgabenstellung schwierig macht.

Die SPV-Spezifikations-Datenbasis kann beliebig in Software- und/oder Hardware-Systeme aufgeteilt werden und wird über einen SPV-Zwischen-Compiler in Hochsprache umgesetzt, beispielsweise für Softwaresysteme in "C", "C++", etc., für Hardwaresysteme in "VHDL", "VERILOG", etc.. Dabei werden für die Hochsprachen-Datenbasis die "If-Then-Else-Konstrukte" jedem Datenverarbeitungs-Knoten automatisch zugeordnet.

Bei Compilierung von Hardwaresystemen wird automatisch die Datenverarbeitungs-Ablaufsteuerung erzeugt.

Die Entwicklung des SPV-Zwischen-Compilers basiert auf einfachen, überschaubaren Parametern, die von der SPV-Spezifikation automatisch generiert werden. Deshalb ist die Entwicklung des SPV-Zwischen-Compilers zu 100% funktional verifizierbar bzw. testbar, so dass seine Compilierungs-Ergebnisse (Hochsprachen) fehlerfrei zur Verfügung stehen, d. h., die SPV-Spezifikations-Datenbasis ist fehlerfrei in der compilierten Hochsprache abgebildet.

Bei der SPV-Entwicklungs-Methode ist ausschließlich die Spezifikation das Entwicklungs-Ergebnis, eine Verifikation ab Hochsprache ist nicht erforderlich.

Im SPV werden parallele und serielle Datenverarbeitungsschritte einer Spezifikation automatisch zugeordnet, unabhängig von der Reihenfolge der Eingabe von Teilspezifikationen.

Änderungen der Spezifikation werden im SPV automatisch auf Konsistenz überwacht. Im Fehlerfall erfolgt eine automatische Anzeige und Führung durch die erforderlichen Korrekturschritte, so dass immer ein fehlerfreier operativer Ablauf der Datenverarbeitung gewährleistet ist.

Durch die Zusammenführung von Spezifikation und Implementierung sowie dem Wegfall der Verifikation wird der System-Entwicklungs-Aufwand mehr als halbiert.

Ausserdem kann im SPV ein Gesamtentwicklungskomplex ohne Schnittstellen-Probleme in Einzelentwicklungskomplexe determiniert an viele Entwickler zugewiesen und mit optimalem Wirkungsgrad bearbeitet werden. Bei anschließender Integration der Einzelentwicklungskomplexe in den Gesamtentwicklungskomplex wird automatisch auf Funktions-Konsistenz geprüft. Hierdurch können extrem kurze System-Entwicklungszeiten erreicht werden.

#### **Die SPV-Entwicklungstool-Realisierung gliedert sich in vier Programme:**

- INSTANZIEREN (Projekt-Toplevel, Komponenten)
- VERBINDEN (Toplevel-/Komponenten-Ports)
- OPERATING (Operations-Zuweisung)
- DATENVERARBEITUNG (Methoden-Ergebnis-Zuweisung in RTI)

Das SPV wurde von LPA-Innovations entwickelt (DPA-/PCT-Anmeldung);  
Geschäftsführung: Dipl.-Ing. Leo P. Anetseder.

Für die Realisierung der Entwicklungstools werden noch Industriepartner gesucht.  
Beteiligung auf der Basis von Exklusiv- oder Teil-Lizenz.

Aufwand und Zeitplan für die Realisierung der Entwicklungstools sowie die Beschaffung von Fördermitteln sind z. Zt. in Bearbeitung.