

OPL – Die Open Programming Language Die Lösung zur Entwicklung von mobilen Anwendungen?

Martin Dehler

Schumannstraße 18
67061 Ludwigshafen
martin.dehler@gmx.de

Abstract: Die Open Programming Language (OPL) ist eine unter der Lesser Gnu Public Licence frei verfügbare Programmiersprache für das derzeit am weitesten verbreitete Smartphone Betriebssystem von Symbian. Mit seiner BASIC ähnlichen Syntax, der kostenlosen Verfügbarkeit von Runtime und Entwicklungsumgebung und der möglichen Nutzung der kompletten C++ API über OPX Erweiterungen hat OPL das Potential, die Entwicklung mobiler Software wesentlich zu erleichtern und damit die Verbreitung von mobilem Computing in der Medizin nachhaltig zu fördern.

1 Mobile medizinische Software von und für Jedermann?

Im Gesundheitswesen findet elektronische Datenverarbeitung (EDV) eine immer größere Beachtung. Insbesondere medizinische Softwarelösungen für mobile Endgeräte, die direkt beim Patienten eingesetzt werden können, bieten das Potenzial, eine weite Verbreitung zu finden. Vom Einsatz mobiler medizinischer Anwendungen erhofft man sich eine Verbesserung von Arbeitsprozessen, der Qualität der erbrachten Leistungen und nicht zuletzt eine Erhöhung der Wirtschaftlichkeit.

Eine möglichst weite Verbreitung mobiler medizinischer EDV-Anwendungen setzt sowohl eine weit verbreitete Hardwareplattform, als auch eine geeignete Programmiersprache voraus. Sowohl für die Hardwareplattform, als auch für die Programmiersprache wäre es wünschenswert, wenn sie nur geringe Investitionskosten erfordern würden. Industrielösungen, kommerzielle Lösungen oder Auftragsprogrammierung sind regelmäßig mit hohen Investitionskosten verbunden. Hohe Anfangsinvestitionen sind eine Hürde die viele EDV-Projekte in der Medizin schon im Planungsstadium scheitern lässt.

1.1 Anforderungen an eine Hardwareplattform

Die Geräteklasse die in der Bevölkerung und auch bei medizinischem Personal bereits heute am weitesten verbreitet ist, sind Mobiltelefone. Bei Mobiltelefonen der neueren

Generation ist die Frage, ob es sich um ein Telefon mit Computer oder um einen Computer mit Telefon handelt eher philosophischer Natur. In der Vergangenheit sind sowohl Telefone als auch Computer zunehmend mobiler geworden und bewegen sich in ihrer Entwicklung immer mehr aufeinander zu. Mit Telefon-, SMS- und Internetfunktionalität, Datenbanken und Text-, Tabellen-, und Präsentationssoftware ist in Form des Smartphones der Brückenschlag bereits vollzogen worden. Im Heise-Verlag erschien Anfang 2003 das c't special 2/2003 zum Thema Handhelds, in dem die verschiedenen Geräteklassen folgendermaßen definiert wurden [Wö03]:

- PDA (Personal Digital Assistant): handgroßes Geräte mit Display, gelegentlich auch als Handheld bezeichnet.
- Wireless PDA/Handheld: PDA mit integriertem Mobiltelefon.
- Smartphone: Geräte, die vom Design und der Funktionalität überwiegend Merkmale eines Mobiltelefons aufweisen, aber ein für Mobiltelefon-Verhältnisse großes, meist farbiges, Display haben, PIM-Synchronisation mit dem PC beherrschen und sich nachträglich mit Zusatzapplikationen versehen lassen.

Das Marktforschungsinstitut Canals hat im März 2003 eine Studie veröffentlicht, nach der die Zahl der verkauften Smartphones im Jahr 2003 verglichen mit dem Vorjahr drastisch ansteigen wird. Mit geschätzten 3,3 Millionen Geräten soll sogar die Zahl der verkauften reinen PDA's (2,8 Millionen) erstmals deutlich überschritten werden [BJ03]. Ein Hinweis, dass dieser vorhergesagte Trend sich tatsächlich so entwickelt hat, sind die von Canals für das vierte Quartal 2003 veröffentlichten Verkaufszahlen von 1,2 Millionen PDA's und 2,2 Millionen Smartphones in Europa, dem mittlerem Osten und Afrika [JL04].

1.2 Anforderungen an eine Programmiersprache

Eine große Programmvielfalt lässt sich insbesondere durch Verfügbarkeit einer leicht zu erlernenden Programmiersprache, mit schnellem Lernerfolg auch für Programmieranfänger, erreichen. Dies ermöglicht einer großen Zahl von potentiellen Softwareentwicklern, die auch IT-Laien (wie z.B. Ärzte) sein können, den Zugang zur Programmierung von selbstentwickelten Anwendungen.

2 Das Symbian Betriebssystem

2.1 Geschichte

Symbian entstand 1998 als eigenständiges Unternehmen der Anteilseigner Ericsson, Motorola, Nokia, und Psion. Unternehmensziel war die Entwicklung eines offenen

Betriebssystem für Mobiltelefone der Zukunft. Dabei brachte Psion sein Betriebssystem Know-how in Form von EPOC Release 5 ein, das im Zuge der Unternehmensgründung in Symbian OS Version 5 umbenannt wurde. Symbian konzentriert sich dabei nur auf die Entwicklung des Betriebssystems, die Benutzeroberflächen und Anwendungen werden von den Lizenznehmern für die von ihnen entwickelten Smartphones entweder selbst erstellt oder lizenziert und dann an Hardware und herstellerspezifisches Benutzeroberflächendesign angepasst.

2.2 aktueller Stand

Symbian gehört im März 2004 sieben Anteilseignern, die mit der Ausnahme Psion's zu den Marktführern im Mobilfunkgeschäft gehören (Abbildung 1). Die verschiedenen Symbian Betriebssystemversionen sind derzeit von 13 Lizenznehmern lizenziert (Abbildung 1). In den Jahren 2002 und 2003 wurden von fünf Lizenznehmern, die 18 verschiedene Smartphones vertreiben, weltweit 8,67 Millionen Smartphones mit dem Symbian Betriebssystem verkauft (über 10 Millionen seit Unternehmensgründung) [Le04]. Neun Lizenznehmer hatten am Ende des vierten Quartals 2003 26 weitere Endgeräte in der Entwicklung [Le04].

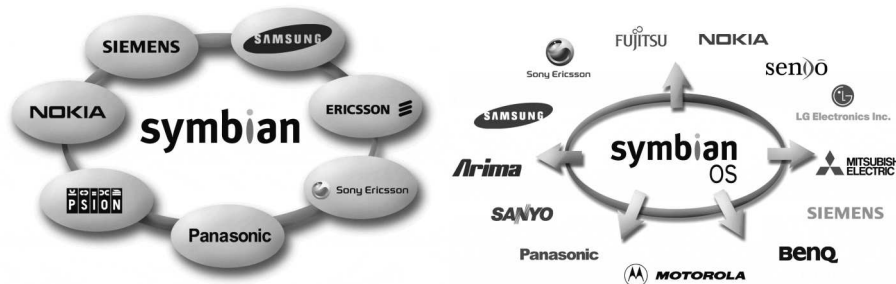


Abbildung 1: Symbian Anteilseigner und Lizenznehmer [Sy04]

Ein einfaches Endgerät für die hier diskutierte Programmiersprache OPL, wie das Nokia 3650, war Mitte März 2004 bei den vier deutschen Mobilfunknetzbetreibern mit 2-Jahresvertrag für durchschnittlich 20 € (0-50 €) erhältlich. Ohne Mobilfunkvertrag lag der Straßenpreis bei ca. 190 € Smartphones mit Symbian Betriebssystem sind bereits im untersten Preissegment verfügbar und damit als massentauglich einstuftbar.

2.3 Benutzeroberflächen für das Symbian Betriebssystem

Während sich Symbian ausschließlich mit der Entwicklung des offenen Betriebssystems beschäftigt, werden die Benutzeroberflächen und serienmäßigen Anwendungen derzeit von einem Subunternehmen und einem Anteilseigner entwickelt. Die Smartphonehersteller lizenzieren für jedes Gerät das Betriebssystem und dazu eine Benutzeroberfläche. Derzeit sind Benutzeroberflächen von Nokia, dem größten Symbian Anteilseigner, und

von UIQ einer 100%-igen Tochterfirma von Symbian verfügbar. Neben der reinen Benutzeroberfläche beinhalten diese Plattformen auch Anwendungsprogramme. Immer enthalten sind Anwendungen für die Telefonfunktion, Adressverwaltung, Kalender, E-Mail/SMS/MMS-Client und Web/WAP-Browser. Die auf dem Smartphone ausgelieferten Anwendungen können von den Lizenznehmern beliebig erweitert werden. Typische weitere Anwendungen, die teilweise auch von den Herstellern der Benutzeroberflächen angeboten werden, sind Aufgabenverwaltung, Notizfunktion, Rechner, Dokumentbetrachter (z.B. für Microsoft Office, Adobe Portable Document Format), Synchronisation nach SyncML-Standard und Spiele. Auch die grafische Gestaltung der Benutzeroberfläche kann teilweise vom Lizenznehmer vorgenommen werden. Die Benutzerlogik bleibt hiervon aber unberührt. Abbildung 2 gibt einen Überblick über die im März 2004 verfügbaren Benutzeroberflächen und die damit ausgestatteten, in Europa erhältlichen, Smartphones. In Abbildung 3 sind einige der bereits erhältlichen Symbian Smartphones abgebildet.

Hersteller	Nokia			UIQ
Oberfläche	Series 80	Series 60	Series 90	UIQ 2.x
Display (Pixel)	640*200	176*208	640*320	208-240*320
Eingabe	QWERTZ-Tastatur, 4-Softkeys	Zifferblock, 2 Softkeys, 5-Wege Navigationstaste	Touchscreen, virtuelle Bildschirmtastatur, 5-Wege Navigationstaste	Touchscreen, Handschrifterkennung, virtuelle Bildschirmtastatur, (Zifferblock)
Geräte Europa in Klammern: 3/04 noch nicht lieferbar	Nokia Communicator 9210i/(9500)	Nokia 7650/3650/ N-Gage/3660/6600, Siemens SX1, (Samsung SGH-D700), (Sendo X), (Panasonic X700)	(Nokia 7700)	SonyEricsson P800/P900, Motorola A920/A925/(A1000), (BenQ P30)

Abbildung 2: Benutzeroberflächen für das Symbian Betriebssystem (Stand: März 2004)



Abbildung 3: einige Symbian Smartphones [Sy04]

2.3 Softwareentwicklung

Der Autor hat nicht die technische Kompetenz und Erfahrung, um alle derzeit für das Symbian Betriebssystem verfügbaren Entwicklungsoptionen zu vergleichen und abschließend zu bewerten. Im Folgenden werden die im März 2004 verfügbaren Optionen zur Softwareentwicklung zusammenfassend dargestellt.

Symbian selber unterstützt die Erstellung von Programmen mit den Programmiersprachen C++ und Java. Für beide Optionen ist die Verwendung eines Software Development Kits (SDK) zwingend erforderlich. Verschiedene SDK's sind sowohl von den Smartphoneherstellern, als auch von Symbian kostenlos erhältlich.

C++

Die Anwendungsentwicklung mit der komplexen Hochsprache C++ wird für alle derzeit verfügbaren Geräte mit Symbian Betriebssystem von Symbian selber und auch den Geräteherstellern unterstützt. Für die C++ Entwicklung wird neben einem SDK zwingend eine kostenpflichtige Entwicklungsumgebung, wie Microsoft Visual C++, Metrowerks CodeWarrior oder Borland C++ Builder Mobile Edition, benötigt.

Java

Java wird von Symbian in Form von PersonalJava (basierend auf Version 1.1.1a) und J2ME MIDP (Java 2 Micro Edition – Mobile Information Device Profile) in der Version 1.0 (teilweise 2.0) unterstützt. Während alle Smartphones mit Symbian Betriebssystem J2ME MIDP Version 1.0 unterstützen, ist PersonalJava nur in Geräten mit Series 80 und UIQ Benutzeroberfläche implementiert. Eine Übersicht über die Implementierung der verschiedenen Java Varianten und Java API's gibt Abbildung 4.

Der grundsätzliche Vorteil der weitgehenden Plattformunabhängigkeit von Java wird durch die uneinheitliche modulare Ergänzung mit standardisierten API's, die aber nicht von allen Geräteherstellern bzw. in allen Geräten implementiert werden, teilweise wieder eingeschränkt. So beinhaltet zum Beispiel das Siemens SX1 eine gerätespezifische Siemens Game API, während die Nokia UI API nur teilweise unterstützt wird [Si03]. J2ME MIDP läuft in einer Sandbox und hat damit keinen direkten Zugriff auf Betriebssystemfunktionen. Aus Gründen der Datensicherheit ist dies begrüßenswert, um aber weitere Funktionalität zu implementieren, sind spezifische API's erforderlich, die teilweise zeitaufwendig im Java Community Process als Java Specification Requests (JSR) erarbeitet werden müssen [Sy02]. Die Speicherung von Daten ist mit J2ME MIDP nur innerhalb der Sandbox im Record Management System (RMS) möglich. Ein Zugriff auf das native Datenbankformat des Symbian Betriebssystems oder ein Datenexport z.B. in Textdateien ist nicht möglich.

Benutzer- oberfläche	Version	Geräte Europa in Klammern: 3/04 noch nicht lieferbar	PersonalJava	JavaPhone API	MIDP	CLDC 1.0	Nokia UI API	WMA	MMAPI	Bluetooth API	Siemens Game API
Series 80		Nokia 9210i	✓	✓	1.0	✓					
		(Nokia 9500)	✓	✓	2.0	✓	✓				
Series 90		(Nokia 7700)			2.0	✓	✓	✓	✓	✓	
Series 60	1.0	Nokia 7650			1.0	✓	✓	✓			
	1.0	Nokia 3650/ N-Gage/3660, Siemens SX1, (Samsung SGH-D700), (Sendo X)			1.0	✓	(✓)	✓	✓		(✓)
	2.0	Nokia 6600/(7610), (Panasonic X700)			2.0	✓	✓	✓	✓	✓	
UIQ	2.0	SonyEricsson P800, Motorola A920/A925, (BenQ P30)	✓		1.0	✓					
	2.1	SonyEricsson P900, (Motorola A1000)	✓		2.0	✓		✓		✓	

Abbildung 4: Symbian Java Implementierung

Erläuterung der Abkürzungen in Abbildung 4: MIDP (Mobile Information Device Profile), CLDC (Connected Limited Device Configuration), UI (User Interface), API (Application Program Interface), WMAPI (Wireless Messaging API (JSR-120)), MMAPI (Mobile Media API (JSR-135)), Bluetooth API (JSR-82 ohne OBEX).

AppForge MobileVB

MobileVB (mobile Visual Basic 6.0) von der Firma AppForge ist für mehrere Plattformen (Symbian (Series 60, UIQ, Series 80), Palm OS, Windows Mobile 2003, Pocket PC 2000/2002) verfügbar. Die MobileVB Lizenz für alle Plattformen (Symbian, Palm OS und Pocket PC) kostet 1000 US\$, die für eine Symbian Benutzeroberfläche (Series 60, Series 80 oder UIQ) 249 US\$ und erfordert damit für den experimentierfreudigen IT-Laien eine nicht unerhebliche Investition (Preise März 2004). Die Mobile VB Runtime ist vergleichsweise speicherintensiv und benötigt z.B. auf Series 60 Geräten 1,1 MB.

Python

Eine Portierung von Python wurde von Nokia auf der O'Reilly Emerging Technologies Conference im Februar 2004 angekündigt. Dabei wurde bereits eine funktionsfähige beta Version für die Series 60 Benutzeroberfläche demonstriert, aber noch kein Veröffentlichungsdatum genannt.

3 OPL

3.1 Geschichte

Die Geschichte der Open Programming Language (OPL) begann 1984. Damals wurde OPL als Organizer Programming Language auf dem Psion Organiser II eingeführt, der vorher nur in Assembler programmiert werden konnte. Der Hauptzweck von OPL sollte damals die Möglichkeit sein, Datenbanken für die im Organiser II eingebaute Datenbankanwendung zu erstellen bzw. Zugriff darauf zu erhalten. Wesentliche Merkmale von OPL sind seit seiner ersten Veröffentlichung eine weitestgehende Rückwärtskompatibilität. OPL Programme können im Regelfall mit wenig Programmieraufwand an neue Betriebssystemplattformen angepasst werden. Ein Schlüsselmerkmal von OPL war und ist die Möglichkeit, OPL Programme komplett auf dem mobilen Endgerät zu entwickeln.

Mit der weiteren Entwicklung der Geräte der Firma Psion über das Betriebssystem SIBO seit 1991 (Serie 3 (a,b,c,mx), Workabout) und EPOC ab 1997 (Serie 5 (mx,mxPRO), revo, netBook, netPad) wurde OPL weiter entwickelt und der Funktionsumfang gesteigert. Mit der Gründung von Symbian 1998 wurde der Unternehmensteil Psion Software in Symbian integriert und Psion's Betriebssystem EPOC Release 5 wurde zum Symbian Betriebssystem Version 5.

Mit dem Übergang des EPOC-Betriebssystems zu Symbian war auch die Zukunft von OPL lange Zeit ungewiss. Dem persönlichen Einsatz einiger Symbian Mitarbeiter und dem dauerhaftem Interesse der OPL Entwicklergemeinde war es zu verdanken, dass Symbian 2001 eine erste Version von OPL für den Nokia 9210 Communicator als alpha Version entwickelte und kostenfrei verfügbar gemacht hat. Im Januar 2003 wurden weiterentwickelte Versionen von Runtime und Programmierumgebung für den Nokia 9210 Communicator von Symbian veröffentlicht. Im April 2003 wurde dann auf der Symbian Entwicklerkonferenz Exposium03 der OPL-Quellcode -jetzt als Open und nicht mehr Organizer Programming Language- unter der Lesser Gnu Public Licence (LGPL) zur Weiterentwicklung durch die OPL Entwicklergemeinde von Symbian veröffentlicht [ASE03]. Zeitgleich wurde eine erste alpha Version einer OPL Runtime für Geräte mit Series 60 Benutzeroberfläche veröffentlicht [ASE03].

3.2 aktueller Stand

Seit der Freigabe als Open-Source Projekt findet sich der OPL Quellcode bei Sourceforge.net und wird da von einer kleinen, aber aktiven Gemeinde weiterentwickelt. Zur Verfügbarkeit der OPL Komponenten im März 2004 siehe Abbildung 5.

3.3 Funktionsumfang

Die Syntax von OPL erinnert stark an die von BASIC und PASCAL. Unterstützt werden die Variablentypen Integer, long Integer, Float und String (255 Zeichen) als lokale und globale Variablen sowie als Arrays. Zur Steuerung des Programmablaufs stehen neben Schleifen, Sprüngen und Verzweigungsfunktionen auch Befehle zur Auswertung von Systemereignissen zur Verfügung. Dazu gehören unter anderem Ein- und Ausschalten des Gerätes, Auswertung von Touchscreeneingaben oder Betriebssystemanforderungen an ein OPL Programm wie 'Programm beenden' oder 'neue Datei erstellen'. Neben als Befehlen definierten mathematischen und trigonometrischen Funktionen verfügt OPL über die Funktion EVAL, die in Textform beschriebene mathematische Ausdrücke berechnet. Zur Bildschirmausgabe stehen Grafikbefehle für verschiedene geometrische Formen, Bitmaps und Zeichensätze in verschiedenen Formatierungen zur Verfügung. Soweit von der Hardware unterstützt, kann mit den Grafikbefehlen der komplette RGB Farbraum mit seinen 16,8 Millionen Farben verwendet werden. Zur Benutzerinteraktion stehen mehrdimensionale Menüs und Dialoge zur Verfügung. In den Dialogen werden verschiedene Eingabeformate wie Text, Freitext, Datum, Zeit, Listenauswahl und Dateiauswahl unterstützt. Weitere Befehle umfassen die Datei- und Verzeichnisverwaltung, basale Druckfunktionen (auch in Dateien) und vielfältige Zeit- und Datumsberechnungen. Eine einfache aber effiziente Fehlerbehandlung ist in OPL integriert. Umfassenden Überblick über alle OPL Befehle gibt die Online Dokumentation OPL Wiki (s.u. 7.).

Die Funktionalität von OPL kann durch in C++ geschriebene Erweiterungen, die OPX genannt werden, auf die komplette C++ API Funktionalität des Symbian Betriebssystems erweitert werden. So sind z.B. OPX verfügbar, die Zugriff auf die Kontakt- und Kalenderdatenbank bieten oder das Senden von Nachrichten als E-Mail, SMS, über Infrarot,... ermöglichen.

Das Symbian Betriebssystem beinhaltet ein relativ komplexes Datenbankmodell. Unter OPL ist die Erstellung von Datenbanken mit mehreren Tabellen möglich. Zum Zugriff auf Datenbanken hat OPL basale SQL-Funktionalität implementiert (SELECT, FROM, WHERE, LIKE, ORDER BY, ASC, DESC). Mit entsprechenden OPX-Erweiterungen ist der Zugriff auf die komplette Funktionalität des Datenbankmodells möglich (alle unter C++ verfügbaren Feldtypen, Indices).

Mit OPL kann vom kurzfristig entwickeltem Prototyp bis zur fertigen komplexen Anwendung die ganze Evolution eines Softwareprojekts mit einer Entwicklungsplattform geleistet werden. Dabei kann die Entwicklung sowohl isoliert auf dem Endgerät, unter Verwendung eines Emulators auf dem PC oder kombiniert erfolgen.

4 Komponenten von OPL

Die Programmiersprache OPL besteht aus mehreren Komponenten, die alle kostenlos unter der Lesser Gnu Public License (LGPL) im Internet erhältlich sind. Es folgt ein

Vorstellung der einzelnen Komponenten und eine Übersicht über die Verfügbarkeit der OPL Komponenten im März 2004.

4.1 Runtime

Die Laufzeitumgebung in Form der OPL Runtime ist zum Ausführen des vom OPL Translator erzeugten Bytecodes notwendig.

Für die Geräte mit Symbian Betriebssystem sind auch auf dem PC lauffähige Versionen der OPL Runtime verfügbar. So können mit einem Emulator, die mit den kostenfreien Software Development Kits der Smartphonehersteller verfügbar sind, OPL Programme auch auf dem PC entwickelt und getestet werden.

4.2 Editor

Prinzipiell handelt es sich beim OPL Editor um einen Texteditor, in den der OPL Translator (s.u. 4.3) integriert ist. Für die Benutzeroberflächen Series 80 und UIQ ist der OPL Editor bereits verfügbar. Für die tastaturlosen Smartphones mit Series 60 Benutzeroberfläche ist im März 2004 nur eine noch unveröffentlichte alpha Version des OPL Editors verfügbar. OPL Programme für alle Benutzeroberflächen können auch auf dem PC mit einem beliebigem Texteditor geschrieben werden und dann mit der PC Version des OPL Translators (OPLTRAN) in Bytecode übersetzt werden, der dann entweder im Emulator auf dem PC oder auf dem Endgerät ausgeführt werden kann.

Die Erstellung von Quellcode auf Geräten mit relativ kleinem Display und eingeschränkten Eingabemöglichkeiten (z.B. haben Series 60 Smartphones nur einen Ziffernblock und einige Funktionstasten) ist zwar nur eingeschränkt möglich, wird aber teilweise durch den Vorteil, 'immer und überall' programmieren zu können, aufgewogen. Weiterhin sind für alle tastaturlosen Benutzeroberflächen des Symbian Betriebssystems Treiber für externe Infrarot-Tastaturen von mehreren Herstellern verfügbar.

4.3 Translator

Der OPL Translator ist ein Programm, das den OPL Quelltext in, von der OPL Runtime ausführbaren, Bytecode kompiliert. Er ist sowohl für das mobile Endgerät, als auch für den PC (als OPLTRAN) verfügbar. Der OPL Translator überprüft die Syntax des Quellcodes auf strukturelle Richtigkeit (IF...ENDIF, Schleifen, ...), nicht deklarierte Variablen und korrekte Syntax der OPL Befehle.

4.4 Tools

Die OPL Tools sind Hilfsprogramme, die nur für den PC verfügbar sind.

- BMLCONV zur Konvertierung von PC Bitmaps (*.BMP) in Multi-Bitmaps (*.MBM), wie sie das Symbian Betriebssystem verwendet.
- MAKESIS zur Erstellung von Symbian Installationsdateien (*.SIS).
- Werkzeuge zum Erstellen von mehrsprachigen Resourcedateien.

4.5 Übersicht über die Verfügbarkeit OPL Komponenten

Abbildung 5 gibt eine Übersicht über die Verfügbarkeit der OPL Komponenten für das Symbian Betriebssystem im März 2004. Neben den verschiedenen Versionen des Symbian Betriebssystems sind die in Europa verfügbaren Geräte, die auf diesen Versionen beruhen, aufgeführt.

Symbian Betriebssystem Version	6.0	6.1	7.0	7.0s	
Geräte Europa <small>in Klammern: 3/04 noch nicht lieferbar</small>	Nokia Communicator 9210i	Nokia 7650, 3650, N-Gage, 3660 Siemens SX1 (Samsung SGH-D700) (Sendo X)	SonyEricsson P800, P900 Motorola A920/A925/(A1000), (BenQ P30)	Nokia 6600/(7610) (Panasonic X700)	(Nokia 7700)
Benutzeroberfläche	Nokia Series 80	Nokia Series 60	UIQ	Nokia Series 60	Series 90
OPL Runtime	✓	✓ (beta)	✓ (beta)	✓ (beta)	pre-alpha
OPL Editor	✓	alpha	✓ (beta)	alpha	pre-alpha
OPL Translator	✓	✓ (beta)	✓ (beta)	✓ (beta)	pre-alpha
OPL Tools <small>(nur PC)</small>	✓	✓	✓	✓	pre-alpha

Abbildung 5: Verfügbarkeit der OPL Komponenten (Stand: März 2004)

5 Programmbeispiele

Grundsätzlich können mit OPL beliebige Programme erstellt werden. Ein Merkmal von OPL ist die Fähigkeit, die Entwicklung von einfachsten Anwendungen bis hin zu

komplexen Anwendungen, die mit Hilfe von OPX Erweiterungen die komplette C++ API Funktionalität des Symbian Betriebssystems nutzen können, mit einer Entwicklungsumgebung zu bewerkstelligen. OPL läßt sich sowohl als Skriptsprache, wie auch als Rapid Application Development Werkzeug nutzen. Im Folgenden werden einige vom Autor mit OPL realisierte medizinische Anwendungsbeispiele für die Series 60 Plattform vorgestellt.

5.1 AaDO₂ – Alveolo-arterielle Sauerstoffpartialdruckdifferenz

Die Alveolo-arterielle Sauerstoffpartialdruckdifferenz (AaDO₂) ist die Differenz der Sauerstoffpartialdrücke im alveolären Mischgas und dem arteriellen Blut. Sie beschreibt die Leistungsfähigkeit des Gasaustauschsystems Lunge-Blut für Sauerstoff. Die meisten Blutgasmessgeräte können die AaDO₂ nach Eingabe des Sauerstoffgehalts der Einatemluft berechnen. Im klinischem Alltag im Operationsaal und auf Intensivstationen wird diese Möglichkeit allerdings häufig nicht genutzt. Da der Grad und die zeitliche Entwicklung einer Gasaustauschstörung, insbesondere bei kritisch kranken Patienten, eine wichtige Information darstellt, besteht oft der Wunsch, die AaDO₂ zu berechnen.

Formel: $AaDO_2 = [(FiO_2 \cdot 7,13) - (PaCO_2 / 0,85)] - PaO_2$

Das OPL Programm zur Berechnung der AaDO₂ umfaßt nur wenige Zeilen Code. Es verfügt nicht über eine grafische Benutzeroberfläche. In der Kürze und Einfachheit des Codes zeigt sich aber, dass ein OPL Programm zur Durchführung einfacher Berechnungen in minutenschnelle zu erstellen ist.

OPL Code zur Berechnung der AaDO₂ (mit Kommentaren):

```
PROC AaDO2:
  LOCAL AaDO2,FiO2,PaO2,PaCO2
  PRINT "Berechnung der AaDO2"
  PRINT "PaO2 [mmHg]", :INPUT PaO2  REM O2-Partialdruck im arteriellem Blut
  PRINT "PaCO2 [mmHg]", :INPUT PaCO2  REM CO2-Partialdruck im arteriellem Blut
  PRINT "FiO2 [%]", :INPUT FiO2  REM prozentualer O2-Gehalt der Einatemluft
  PRINT "AaDO2 "+FIX$(((FiO2*7.13)-(PaCO2/0.85))-PaO2,1,5)+" [mmHg]"
  GET
ENDP
```

Da in diesem Beispiel kein gerätespezifischer Code verwendet wird, ist es auf jedem Gerät, für das eine OPL Runtime verfügbar ist, ausführbar.

Der 'logische Kern' eines OPL Programms ist im wesentlichen auf allen OPL tauglichen Geräten nutzbar. Es ist nur die, wenn auch manchmal zeitintensive, Anpassung der Benutzeroberfläche an die jeweilige Hardwareplattform notwendig. Teilweise sind auch unterschiedliche Versionen der OPX Erweiterungen erforderlich, da die C++ API in den verschiedenen Versionen des Symbian Betriebssystems zwar weitestgehend rückwärtskompatibel, aber nicht durchgängig identisch ist.

5.2 CalGCS – Calculate Glasgow-Coma-Scale

Die Glasgow-Coma-Scale wurde 1974 von Teasdale et al. entwickelt und ist der verbreitetste Score zur Beschreibung der Bewusstseinslage. Die Glasgow-Coma-Scale wird insbesondere bei Patienten mit Schädel-Hirn-Trauma verwendet, wo sie auch prognostische Bedeutung hat [Te74].

5.3 DOSING – Dosierung bei Niereninsuffizienz

DOSING ist ein Projekt der Abteilung Innere Medizin VI (Klinische Pharmakologie und Pharmakoepidemiologie) der Medizinischen Klinik und Poliklinik an der Universität Heidelberg (Ärztlicher Direktor: Prof. Dr. med. Walter E. Haefeli, Koordination: Dr. Meret Martin-Facklam) [De84; Fa01; Ma00]. Unter www.dosing.de werden im Internet verschiedene Dienstleistungen rund um den Themenkomplex Arzneimittel-Anwendung und Sicherheit angeboten.

Hierzu gehört unter anderem der Service Dosisanpassung bei Niereninsuffizienz. Die Medikamentendatenbank und die Berechnungsroutinen sind in Form von HTML Dokumenten und JavaScript realisiert. Die Webseiten können damit prinzipiell auch lokal auf jedem mobilen Endgeräten mit einem JavaScript fähigem Browser gespeichert und genutzt werden. Allerdings sind die Ressourcen, die Browser mit entsprechendem Funktionsumfang benötigen, meist relativ groß und die zu speichernden Datenmengen erheblich. Ein OPL Programm, das eine auf dem Gerät gespeicherte Medikamentendatenbank ausliest, bei Bedarf die Kreatinin-clearance und die Ausscheidungskapazität berechnet und daraus Vorschläge zur Dosisanpassung berechnet, war relativ einfach zu realisieren.

5.4 CalMEES – Calculate Mainz-Emergency-Evaluation-Score

Der Mainz-Emergency-Evaluation-Score (MEES) wurde 1992 von Hennes et al. entwickelt und validiert [HRD92]. Der MEES dient in der Versorgung eines Notfallpatienten im präklinischen Rettungsdienst zur Zustands- und Verlaufsbeurteilung [Me96]. Dabei werden die Daten von 7 lebenswichtigen Funktionen erfasst und je nach Abweichung vom physiologischen Normalwert einem Punktwert von 1-4 zugeordnet. Die erfassten lebenswichtigen Funktionen sind: Bewusstsein (in Form der Glasgow-Coma-Scale), Schmerz, Atemfrequenz, arterielle Sauerstoffsättigung, Herzfrequenz, Blutdruck und EKG-Rhythmus. Der Punktwert 1 beschreibt einen akut lebensgefährlichen Zustand, der Punktwert 4 den physiologischen Normalzustand. Dabei kann der MEES Punktwerte zwischen 8 und 28 annehmen. Die Erfassung und Berechnung des MEES erfolgt zu den beiden Zeitpunkten Erstbefund am Notfallort und Übergabe im Krankenhaus. Die Differenz dieser beiden Punktwerte bildet den delta-MEES. Ein delta-MEES von mehr als ± 2 Punkten wird als Verbesserung bzw. Verschlechterung des Patientenzustandes interpretiert.

Seit der Version 4.0 ist der MEES integraler Bestandteil des Notarztprotokolls nach Empfehlung der DIVI (Deutsche interdisziplinäre Vereinigung für Intensivmedizin) [Mö99; Re99]. Das DIVI Notarztprotokoll ist ein zwei DIN A4 Seiten umfassendes Papierprotokoll, das in Deutschland eine sehr weite Verbreitung zur Dokumentation von Notarzteinsätzen hat. Die mit dem DIVI Protokoll erfassten Daten sind weitgehend mit den Daten des minimalen Notarzt Datensatzes (MIND) identisch, der als Qualitätsmanagementinstrument im Rettungsdienst empfohlen wird [FM96; MMM98].

In der Realität wird derzeit aber nur ein Bruchteil der Notarzteinsätze EDV auswertbar erfasst und noch weniger Datensätze einer zentralen, vergleichenden Auswertung zugänglich gemacht [MF96]. Da der MEES das zentrale Element eines empfohlenen Qualitätsmanagements ist und an vielen Stellen erhebliche Hemmungen bestehen, in Hard- und Software für ein mobiles EDV Erfassungssystem zu investieren, ist eine zeitnahe Erfassung der entsprechenden Daten auf einem Smartphone bereits im Einsatz oder unmittelbar danach, eine preiswerte und damit verbreitungstaugliche Variante um ein minimalistisches Qualitätsmanagement zu betreiben.

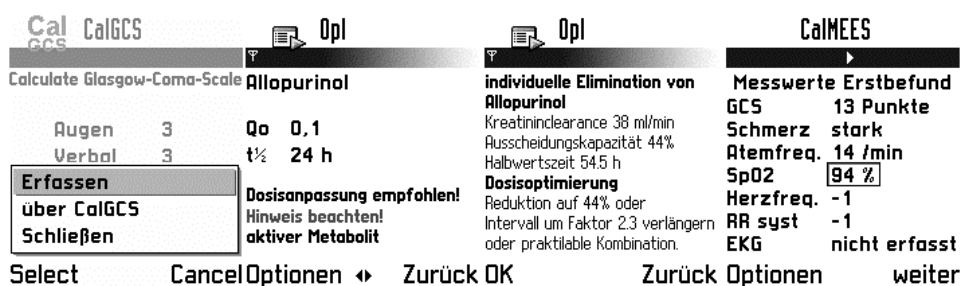


Abbildung 6: Screenshots der Beispielprogramme für die Series 60 Oberfläche aus Kapitel 5.2-5.4

6 Fazit

Mit OPL ist eine leicht erlernbare prozedurale Programmiersprache für das derzeit am weitesten verbreitete Smartphone Betriebssystem der Firma Symbian verfügbar. Der Einstieg in die Entwicklung von OPL Programmen erfordert auch vom Programmieranfänger nur minimalen zeitlichen und monetären Aufwand und verspricht schnelle Erfolgserlebnisse. Die Einstiegshürden beim Erlernen von OPL sind wesentlich niedriger als bei C++ und auch bei Java. Zudem ist OPL mit der Möglichkeit durch OPX Erweiterungen auf die komplette C++ API Funktionalität des Betriebssystems erweitert zu werden, deutlich flexibler als J2ME MIDP, das in einer Sandbox läuft und damit keinen unmittelbaren Zugriff auf basale Betriebssystemfunktionen und das Dateisystem erlaubt. Da OPL unter der Lesser Gnu Public License (LGPL) veröffentlichte freie Software ist, fallen keinerlei Kosten in Form von Lizenzgebühren wie bei AppForge's MobileVB an, was OPL gerade für Nischenlösungen mit geringer Verbreitung und niedrigem Budget prädestiniert. Durch die freie Verfügbarkeit des OPL Quellcodes ist es möglich, den Quellcode auf eventuelle Sicherheitslücken hin zu untersuchen und sich an

der Weiterentwicklung der Programmiersprache zu beteiligen. OPL ist für alle derzeit verbreiteten Benutzeroberflächen des Symbian Betriebssystems verfügbar. Einer Portierung auf andere Betriebssysteme steht nichts entgegen. Als unter einer Open-Source Lizenz veröffentlichte Programmiersprache unterliegt OPL nicht, teilweise langwierigen, Definitionsprozessen wie im Java Community Prozess, was z.B. die Aufmerksamkeit von IBM auf OPL gelenkt hat. Durch die knapp 20-jährige Geschichte des OPL Vorgängers gibt es bereits vielfältige gut dokumentierte Codebeispiele, die meistens nur geringe Anpassungen für die aktuellen Geräte bzw. Benutzeroberflächen benötigen. Durch die Möglichkeit OPL Programme auch direkt auf dem Endgerät zu entwickeln, ist ein Programmieren 'immer und überall' möglich. Allerdings ist dies bei Smartphones, die nicht über eine Tastatur verfügen, nur extrem eingeschränkt bzw. nur mit einer externen Tastatur möglich. Andererseits ist mit den in den kostenfreien SDK's enthaltenen Emulatoren der Smartphones auch eine PC basierte Entwicklung möglich, ohne ein Gerät mit Symbian Betriebssystem verfügbar zu haben. Für die Verwendung in bezüglich elektromagnetischer Verträglichkeit kritischen Bereichen, unterstützen alle Geräte mit Symbian Betriebssystem entweder serienmäßig oder durch Anwendungen von Drittanbietern einen 'flight mode', in dem der Mobilfunkteil des Gerätes abgeschaltet werden kann.

OPL kann und will kein Ersatz für komplexe objektorientierte Hochsprachen wie C++ oder Java sein. OPL hat aber mit seiner Kombination aus Einfachheit und Erweiterungsmöglichkeiten bei gleichzeitig freier Verfügbarkeit und frei zugänglichem Quellcode das Potential, die Entwicklung mobiler Software wesentlich zu erleichtern und damit die Verbreitung von mobilem Computing in der Medizin nachhaltig zu fördern.

7 weiterführende Informationen über OPL

Der komplette OPL Quellcode für Runtime, Editor, Translator, OPX Erweiterungen und die OPL-Dokumentation sind seit der Freigabe als Open-Source unter der LGPL Lizenz durch Symbian im April 2003 bei Sourceforge.net verfügbar [ASE03].

- <http://opl-dev.sourceforge.net/> - Die 'offizielle' Homepage von OPL. Hier finden sich aktuelle Runtimes ('user package') und Entwicklungsumgebungen ('developer package').
- <https://sourceforge.net/projects/opl-dev/> - Heimat des OPL-Quellcodes bei Sourceforge.net und Anlaufstelle für alle, die sich an der Entwicklung von OPL beteiligen wollen.
- <http://www.allaboutopl.com/wiki/OPLWikiHome/> - Die OPL Wiki ist die interaktive Online-Dokumentation der OPL-Befehle und bietet auch weiterführende Informationen und Anleitungen rund um OPL.
- <http://opl.symbiandiaries.com/> - Informationen, Neuigkeiten, Artikeln und Links rund um OPL.

Literaturverzeichnis

- [ASE03] Andrews, R.; Smith, H.; Spence, E.: Symbian publishes OPL language source code under LGPL. Symbian Ltd (Pressemitteilung), 28.04.2003, www.symbian.com.
- [BJ03] Buss, A.; Jones, C.: Smart phones will overtake handhelds this year in EMEA. Canalys, 24.03.2003, www.canalys.com.
- [De84] Dettli, L.: The kidney in pre-clinical and clinical pharmacokinetics. *Jpn J Clin Pharmacol Ther*, 1984; S. 241-254.
- [Fa01] Falconnier, A. D.; Haefeli, W. E.; Schoenenberger, R. A.; Surber, C.; Martin-Facklam, M.: Drug dosage in patients with renal failure optimized by immediate concurrent feedback. *J.Gen.Intern.Med.*, 06.2001; S. 369-375.
- [FM96] Friedrich, H. J.; Messelken, M.: Der minimale Notarzt Datensatz (MIND). *Der Notarzt*, 1996; S. 186-190.
- [HRD92] Hennes, H. J.; Rheinhardt, T.; Dick, W.: MEES Beurteilung des Patienten mit dem Mainz Emergency Evaluation Score. *Notfallmedizin*, 1992; S. 130-136.
- [JL04] Jones, C.; Lashford, R.: Over 1 million Handhelds, 2 million Smartphones shipped in Q4 2003. Canalys, 26.04.2004, www.canalys.com.
- [Le04] Levin, D.: Q4 2003 Highlights. Symbian Ltd (Pressemitteilung), 23.02.2004, www.symbian.com.
- [Ma00] Martin-Facklam, M.: Individualisierung der Arzneimitteltherapie bei Nieren- und Leberinsuffizienz. *Ther.Umsch.*, 09.2000; S. 568-572.
- [Me96] Messelken, M.: Evaluation der Ergebnisqualität von Notarzteinsätzen mit dem MEES - Ergebnisse einer multizentrischen Praktikabilitätsstudie. *Der Notarzt*, 1996; S. 60-64.
- [MF96] Messelken, M.; Friedrich, H. J.: Empfehlung für eine Standardauswertung von Notarzteinsätzen. *Der Notarzt*, 1996; S. 191-193.
- [MMM98] Messelken, M.; Martin, J.; Milewski, P.: Ergebnisqualität in der Notfallmedizin. *Notfall & Rettungsmedizin*, 1998; S. 143-149.
- [Mö99] Moecke, H.; Dirks, D.; Friedrich, H. J.; Hennes, H. J.; Lackner, C.; Messelken, M.; Neumann, C.; Pajonk, F. G.; Reng, M.; Schächinger, U.; Violka, T.: DIVI-Notarzteinsatzprotokoll - Version 4.0. *Der Notarzt*, 1999; S. 97-100.
- [Re99] Reinhardt, T.; Hennes, H. J.: Mainz Emergency Evaluation Score (MEES). *Notfall & Rettungsmedizin*, 1999; S. 380-381.
- [Si03] Siemens mobile: Siemens SX1 - Technical Note for Application Developers Version 1.1. Siemens AG, 11.2003, www.siemens-mobile.com/dveloper/.
- [Sy02] Symbian Developer Support: Differences between PersonalJava and MIDP Java Environments. Symbian Ltd, 06.12.2002, www.symbian.com/developer/.
- [Sy04] Symbian Picture Library: Pressefoto (frei verfügbar). Symbian Ltd, 15.03.2004, www.symbian.com/press-office/.
- [Sy04] Symbian Picture Library: Pressefoto (frei verfügbar). Symbian Ltd, 15.03.2004, www.symbian.com/press-office/.
- [Te74] Teasdale, G.; Jennett, B.: Assessment of coma and impaired consciousness. A practical scale. *Lancet*, 13.07.1974; S. 81-84.
- [Wö03] Wöltje, H.: ct special Handhelds 02/2003. *ct special*, 2003; S. 9.