
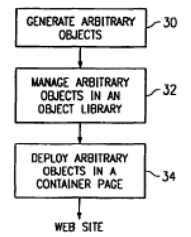



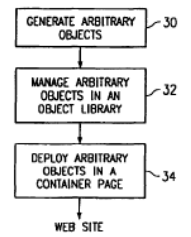
EXHIBIT C

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

 US007716629B2	
(12) United States Patent McAuley	(10) Patent No.: US 7,716,629 B2 (45) Date of Patent: *May 11, 2010
<hr/>	
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	6,026,433 A * 2/2000 D'Aulach et al. 709/217 6,028,998 A * 2/2000 Gledeman et al. 717/108 6,052,670 A * 4/2000 Johnson 705/27 6,199,082 B1 * 3/2001 Fernel et al. 715/522 6,219,680 B1 * 4/2001 Bernardo et al. 715/501.1 6,226,648 B1 * 5/2001 Appleman et al. 707/102 6,247,032 B1 * 6/2001 Bernardo et al. 715/530
(75) Inventor: Aubrey McAuley , Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc. , Fort Worth, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days. This patent is subject to a terminal disclaimer.	
(Continued)	
FOREIGN PATENT DOCUMENTS	
(21) Appl. No.: 10/999,911	CA 2110970 6/1995
OTHER PUBLICATIONS	
(22) Filed: Nov. 29, 2004	Adaptive Media, Inc., Dynamic Web Management Tools, WebOS/NeoFlash/StaticFlash, Assorted Documents, pp. MSVERT002 through MSVERT126.
(Continued)	
Primary Examiner—John Chavis	
(65) Prior Publication Data US 2005/0154486 A1 Jul. 14, 2005	(74) <i>Attorney, Agent, or Firm</i> —Scheef & Stone, L.L.P.; Jack D. Stone, Jr.
Related U.S. Application Data	
Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.	
(51) Int. Cl. G06F 9/45 (2006.01)	(57) ABSTRACT
(52) U.S. Cl. 717/100	A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library. The arbitrary objects are deployed from said object library into a design framework to create said computer application.
(58) Field of Classification Search 717/106, 717/100, 149	
See application file for complete search history.	
References Cited	
U.S. PATENT DOCUMENTS	
5,544,302 A * 8/1996 Nguyen 715/837 5,555,365 A * 9/1996 Selby et al. 715/765 5,894,554 A * 4/1999 Lowery et al. 709/203 5,895,476 A 4/1999 Orr et al. 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 715/513	
32 Claims, 2 Drawing Sheets	
	



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

 US007716629B2	
(12) United States Patent McAuley	(10) Patent No.: US 7,716,629 B2 (45) Date of Patent: *May 11, 2010
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	6,026,433 A * 2/2000 D'Aulach et al. 709/217 6,028,998 A * 2/2000 Gledeman et al. 717/108 6,052,670 A * 4/2000 Johnson 705/27 6,199,082 B1 * 3/2001 Fernel et al. 715/522 6,219,680 B1 * 4/2001 Bernardo et al. 715/501.1 6,226,648 B1 * 5/2001 Appleman et al. 707/102 6,247,032 B1 * 6/2001 Bernardo et al. 715/530
(75) Inventor: Aubrey McAuley , Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc. , Fort Worth, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.	(Continued)
This patent is subject to a terminal disclaimer.	FOREIGN PATENT DOCUMENTS
	CA 2110970 6/1995
(21) Appl. No.: 10/999,911	OTHER PUBLICATIONS
(22) Filed: Nov. 29, 2004	Adaptive Media, Inc., Dynamic Web Management Tools, WebOS/NovelFlash/StaticFlash, Assorted Documents, pp. MSVERT102 through MSVERT126.
(65) Prior Publication Data US 2005/0154486 A1 Jul. 14, 2005	(Continued)
Related U.S. Application Data	Primary Examiner—John Chavis
(63) Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.	(74) Attorney, Agent, or Firm—Scheef & Stone, L.L.P.; Jack D. Stone, Jr.
(51) Int. Cl. G06F 9/45 (2006.01)	(57) ABSTRACT
(52) U.S. Cl. 717/100	A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library into a design framework to create said computer application.
(58) Field of Classification Search 717/106, 717/100, 149	
See application file for complete search history.	
(56) References Cited U.S. PATENT DOCUMENTS	
5,544,302 A * 8/1996 Nguyen 715/837 5,555,365 A * 9/1996 Selby et al. 715/765 5,894,554 A * 4/1999 Lowery et al. 709/203 5,895,476 A * 4/1999 Orr et al. 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 715/513	
	32 Claims, 2 Drawing Sheets
	

Samsung's Infringing Android-Powered Smartphones

Acclaim SCH-R880	Galaxy S II/SGH i777
Admire Vitality	Galaxy S II/SGH-T989
Behold II	Galaxy S II Skyrocket
Captivate (Galaxy S)	Gem
Conquer 4G	Gravity Smart
Continuum (Galaxy S)	Infuse 4G
Dart	Intercept
Droid Charge	Moment
Epic 4G (Galaxy S)	Replenish
Epic 4G Touch (Galaxy S II)	Sidekick 4G
Exhibit 4G (Galaxy S)	Stratosphere
Exhibit II 4G	Transfix
Fascinate/Mesmerize (Galaxy S)	Transform Ultra
Galaxy Indulge R910/	Transform
Indulge R915	Vibrant (Galaxy S)
Galaxy Prevail	
Galaxy S 4G	


Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

(12) United States Patent		(10) Patent No.: US 7,716,629 B2	
McAuley		(45) Date of Patent: *May 11, 2010	
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK		6,026,433 A * 2/2000 D'Aulach et al. 709/217	
		6,028,998 A * 2/2000 Gledeman et al. 717/108	
		6,052,670 A * 4/2000 Johnson 705/27	
		6,199,082 B1 * 3/2001 Fernel et al. 715/522	
(75) Inventor: Aubrey McAuley , Austin, TX (US)		6,219,680 B1 * 4/2001 Bernardo et al. 715/501.1	
(73) Assignee: Vertical Computer Systems, Inc. , Fort Worth, TX (US)		6,226,648 B1 * 5/2001 Appleman et al. 707/102	
		6,247,032 B1 * 6/2001 Bernardo et al. 715/530	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.		(Continued)	
This patent is subject to a terminal disclaimer.		FOREIGN PATENT DOCUMENTS	
		CA 2110970 6/1995	
(21) Appl. No.: 10/999,911		OTHER PUBLICATIONS	
(22) Filed: Nov. 29, 2004		Adhesive Media, Inc., Dynamic Web Management Tools, WebOS/	
(65) Prior Publication Data		NewFlash/StdFlash, Assorted Documents, pp. MSVERT002 through MSVERT126.	
US 2005/0154486 A1 Jul. 14, 2005		(Continued)	
Related U.S. Application Data		Primary Examiner—John Chavis	
(63) Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.		(74) Attorney, Agent, or Firm— Scheef & Stone, L.L.P.; Jack D. Stone, Jr.	
(51) Int. Cl.		(57) ABSTRACT	
G06F 9/45 (2006.01)		A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library. The arbitrary objects are deployed from said object library into a design framework to create said computer application.	
(52) U.S. Cl. 717/100			
(58) Field of Classification Search 717/106, 717/100, 149			
See application file for complete search history.			
(56) References Cited			
U.S. PATENT DOCUMENTS			
5,544,302 A * 8/1996 Nguyen 715/837			
5,555,365 A * 9/1996 Selby et al. 715/765			
5,894,554 A * 4/1999 Lowery et al. 709/203			
5,895,476 A * 4/1999 Orr et al. 707/100			
5,903,894 A * 5/1999 Reneris 717/102			
5,930,512 A * 7/1999 Boden et al. 715/513			
5,956,736 A * 9/1999 Hanson et al. 715/513			
		32 Claims, 2 Drawing Sheets	

The Samsung Android-Powered Smartphone on Figure 1 infringe claims of U.S. Patent number 7,716,629 because they include "A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application." Samsung imports these Smartphones into the United States.

A user of these Samsung Android-Powered Smartphones can generate a computer application within the Android framework on Samsung's Android-Powered Smartphones themselves by using text editors or other text manipulation techniques provided on the Smartphones. After compiling the application, the executable can be put back on the Smartphone and run.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers


US007716629B2

(12) **United States Patent**
McAuley

(10) **Patent No.:** **US 7,716,629 B2**
(45) **Date of Patent:** ***May 11, 2010**

(54) **SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK**

(75) Inventor: **Aubrey McAuley**, Austin, TX (US)

(73) Assignee: **Vertical Computer Systems, Inc.**, Fort Worth, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.
This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/999,911**

(22) Filed: **Nov. 29, 2004**

(65) **Prior Publication Data**
US 2005/0154486 A1 Jul. 14, 2005

Related U.S. Application Data
(63) Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.

(51) **Int. Cl.** **G06F 9/45** (2006.01)

(52) **U.S. Cl.** **717/100**

(58) **Field of Classification Search** **717/106, 717/100, 149**
See application file for complete search history.

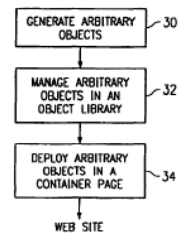
(56) **References Cited**
U.S. PATENT DOCUMENTS
5,544,302 A * 8/1996 Nguyen 715/837
5,555,365 A * 9/1996 Selby et al. 715/765
5,894,554 A * 4/1999 Lowery et al. 709/203
5,895,476 A 4/1999 Orr et al.
5,903,894 A * 5/1999 Reneris 707/100
5,930,512 A * 7/1999 Boden et al. 717/102
5,956,736 A * 9/1999 Hanson et al. 715/513

(Continued)
FOREIGN PATENT DOCUMENTS
CA 2110970 6/1995

OTHER PUBLICATIONS
Adhesive Media, Inc., Dynamic Web Management Tools, WebOS/NovelFlash/StaticFlash, Assorted Documents, pp. MSVERT102 through MSVERT126.
(Continued)
Primary Examiner—John Chavis
(74) *Attorney, Agent, or Firm*—Scheef & Stone, L.L.P.; Jack D. Stone, Jr.

(57) **ABSTRACT**
A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library. The arbitrary objects are deployed from said object library into a design framework to create said computer application.

32 Claims, 2 Drawing Sheets



Samsung's Infringing Android-Powered Tablets

Samsung Galaxy Tab10.1 with Verizon 4G LTE

Samsung Galaxy Tab 10.1

Samsung Galaxy Tab 7.0

Samsung Galaxy Tab 8.9

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers


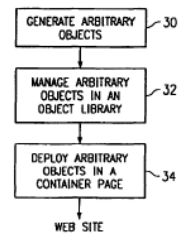
(12) United States Patent		(10) Patent No.:	US 7,716,629 B2
McAuley		(45) Date of Patent:	*May 11, 2010
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	6,026,433 A *	2/2000	D'Aulach et al. 709/217
	6,028,998 A *	2/2000	Goodman et al. 717/108
	6,052,670 A *	4/2000	Johnson 705/27
	6,199,082 B1 *	3/2001	Ferrel et al. 715/522
	6,219,680 B1 *	4/2001	Bernardo et al. 715/501.1
	6,226,648 B1 *	5/2001	Appleman et al. 707/102
	6,247,032 B1 *	6/2001	Bernardo et al. 715/530
(75) Inventor:	Aubrey McAuley , Austin, TX (US)		
(73) Assignee:	Vertical Computer Systems, Inc. , Fort Worth, TX (US)		
(*) Notice:	Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.		
	This patent is subject to a terminal disclaimer.		
(21) Appl. No.:	10/999,911		
(22) Filed:	Nov. 29, 2004		
(65) Prior Publication Data	US 2005/0154486 A1 Jul. 14, 2005		
Related U.S. Application Data	Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.		
(51) Int. Cl.	G06F 9/45 (2006.01)		
(52) U.S. Cl.	717/100		
(58) Field of Classification Search	717/106, 717/100, 149		
See application file for complete search history.			
(56) References Cited	U.S. PATENT DOCUMENTS		
	5,544,302 A *	8/1996	Nguyen 715/837
	5,555,365 A *	9/1996	Selby et al. 715/765
	5,894,554 A *	4/1999	Lowery et al. 709/203
	5,895,476 A *	4/1999	Orr et al. 707/100
	5,903,894 A *	5/1999	Remeris 717/102
	5,930,512 A *	7/1999	Boden et al. 715/513
	5,956,736 A *	9/1999	Hanson et al. 715/513
32 Claims, 2 Drawing Sheets			

The Samsung Android-Powered Tablet Computers on Figure 2 infringe claims of U.S. Patent number 7,716,629 because they include **"A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application."** Samsung imports these Tablet Computers into the United States.

A user of these Samsung Android-Powered Tablet Computers can generate a computer application within the Android framework on Samsung's Android-Powered Tablet Computers themselves by using text editors or other text manipulation techniques provided on the Tablet Computers. After compiling the application, the executable can be put back on the Tablet Computer and run.

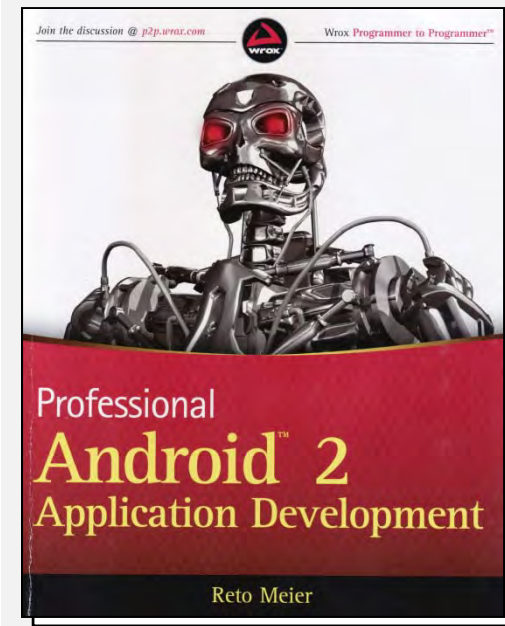
We will refer from now on to both Samsung's Android-Powered Smartphones and Samsung's Android-Powered Tablet Computers as "Android Devices".

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers


 US007716629B2	
(12) United States Patent McAuley	
(10) Patent No.: (45) Date of Patent:	US 7,716,629 B2 *May 11, 2010
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	
(75) Inventor: Aubrey McAuley, Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc., Fort Worth, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days. This patent is subject to a terminal disclaimer.	
(Continued)	
FOREIGN PATENT DOCUMENTS	
CA	2110970 6/1995
OTHER PUBLICATIONS	
Adhesive Media, Inc., Dynamic Web Management Tools, WebOS/NeoFlash/StaticFlash, Assorted Documents, pp. MSVERT102 through MSVERT126.	
(Continued)	
(21) Appl. No.: 10/999,911 (22) Filed: Nov. 29, 2004 (65) Prior Publication Data US 2005/0154486 A1 Jul. 14, 2005 Related U.S. Application Data (63) Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.	
(51) Int. Cl. G06F 9/45 (2006.01)	
(52) U.S. Cl. 717/106, 717/100, 149	
(58) Field of Classification Search See application file for complete search history.	
(56) References Cited U.S. PATENT DOCUMENTS 5,544,302 A * 8/1996 Nguyen 715/837 5,555,365 A * 9/1996 Selby et al. 715/765 5,894,554 A * 4/1999 Lowery et al. 709/203 5,895,476 A 4/1999 Orr et al. 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 715/513	
32 Claims, 2 Drawing Sheets	
	

INFRINGEMENT BY THE ANDROID FRAMEWORK

The book "Professional Android 2 Application Development", ISBN 978-0-470-5656552-0 written by Reto Meier will be used to explain the infringement. "Reto Meier is a software developer who has been involved in Android since its initial release in 2007. He is an Android Developer Advocate at Google". This book will be referred to henceforth as the "Meier Book".



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers



US007716629B2

(12) **United States Patent**
McAuley

(10) **Patent No.:** **US 7,716,629 B2**
(45) **Date of Patent:** ***May 11, 2010**

(54) **SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK**

(75) **Inventor:** **Aubrey McAuley**, Austin, TX (US)

(73) **Assignee:** **Vertical Computer Systems, Inc.**, Fort Worth, TX (US)

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.

This patent is subject to a terminal disclaimer.

(21) **Appl. No.:** **10/999,911**

(22) **Filed:** **Nov. 29, 2004**

(65) **Prior Publication Data**
US 2005/0154486 A1 Jul. 14, 2005

Related U.S. Application Data
(63) Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.

(51) **Int. Cl.** **G06F 9/45** (2006.01)

(52) **U.S. Cl.** **717/106**

(58) **Field of Classification Search** **717/106, 717/100, 149**

See application file for complete search history.

(56) **References Cited**
U.S. PATENT DOCUMENTS

5,544,302 A *	8/1996	Nguyen	715/837
5,555,365 A *	9/1996	Selby et al.	715/765
5,894,554 A *	4/1999	Lowery et al.	709/203
5,895,476 A *	4/1999	Orr et al.	
5,903,894 A *	5/1999	Remeris	707/100
5,930,512 A *	7/1999	Boden et al.	717/102
5,956,736 A *	9/1999	Hanson et al.	715/513

32 Claims, 2 Drawing Sheets

(Continued)

FOREIGN PATENT DOCUMENTS

CA 2110970 6/1995

OTHER PUBLICATIONS

Adhesive Media, Inc., Dynamic Web Management Tools, WebOS/NewsFlash/Slide/Flash, Assorted Documents, pp. MSVERT002 through MSVERT126.

(Continued)

Primary Examiner—John Chavis
(74) *Attorney, Agent, or Firm—Scheef & Stone, L.L.P.; Jack D. Stone, Jr.*

(57) **ABSTRACT**

A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library. The arbitrary objects are deployed from said object library into a design framework to create said computer application.

30

GENERATE ARBITRARY OBJECTS

↓

32

MANAGE ARBITRARY OBJECTS IN AN OBJECT LIBRARY

↓

34

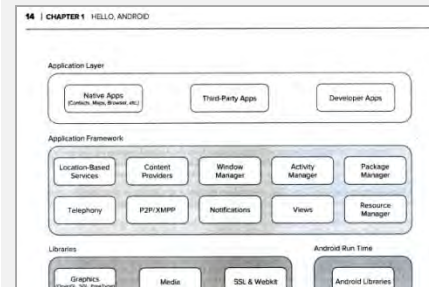
DEPLOY ARBITRARY OBJECTS IN A CONTAINER PAGE

↓

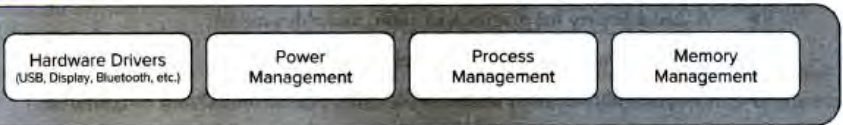
WEB SITE

GENERATION OF AN ANDROID APPLICATION ON THE SMARTPHONE

The Android Framework includes the Linux Operating System. As per the Meier Book, **Page 14**, in the lowermost block, it shows that the Linux is the lowest part of the Android Framework:



Linux Kernel



Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources.


► **Application layer** All applications, both native and third-party, are built on the application layer by means of the same API libraries. The application layer runs within the Android run time, using the classes and services made available from the application framework.

The Dalvik Virtual Machine

One of the key elements of Android is the Dalvik virtual machine. Rather than use a traditional Java virtual machine (VM) such as Java ME (Java Mobile Edition), Android uses its own custom VM designed to ensure that multiple instances run efficiently on a single device.

The Linux Portion of the Android Framework includes built-in command line commands such as CAT, VI, CP, MV and RM. (Cat Can create text files, VI can modify them, CP, MV and RM can Copy, Move, Rename and Delete files).

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers



US007716629B2

(12) **United States Patent**
McAuley

(10) **Patent No.:** **US 7,716,629 B2**
(45) **Date of Patent:** ***May 11, 2010**

(54) **SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK**

(75) **Inventor:** **Aubrey McAuley**, Austin, TX (US)

(73) **Assignee:** **Vertical Computer Systems, Inc.**, Fort Worth, TX (US)

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.
This patent is subject to a terminal disclaimer.

(21) **Appl. No.:** **10:999,911**

(22) **Filed:** **Nov. 29, 2004**

(65) **Prior Publication Data**
US 2005/0154486 A1 Jul. 14, 2005

Related U.S. Application Data

(63) Continuation of application No. 09/410,334, filed on Oct. 1, 1999, now Pat. No. 6,826,744.

(51) **Int. Cl.**
G06F 9/45 (2006.01)

(52) **U.S. Cl.** 717/100

(58) **Field of Classification Search** 717/106, 717/100, 149
See application file for complete search history.

(56) **References Cited**
U.S. PATENT DOCUMENTS

5,544,302 A *	8/1996	Nguyen	715/837
5,555,365 A *	9/1996	Selby et al.	715/765
5,894,554 A *	4/1999	Lowery et al.	709/203
5,895,476 A *	4/1999	Orr et al.	
5,903,894 A *	5/1999	Reneris	707/100
5,930,512 A *	7/1999	Boden et al.	717/102
5,956,736 A *	9/1999	Hanson et al.	715/513

(Continued)

FOREIGN PATENT DOCUMENTS

CA	2110970	6/1995
----	---------	--------

OTHER PUBLICATIONS

Adhesive Media, Inc., Dynamic Web Management Tools, WebOS/NovoFlash/StaticFlash, Assorted Documents, pp. MSVERT102 through MSVERT126.

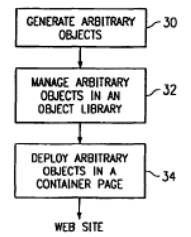
(Continued)

Primary Examiner—John Chavis
(74) *Attorney, Agent, or Firm*—Scheef & Stone, L.L.P.; Jack D. Stone, Jr.

(57) **ABSTRACT**

A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library. The arbitrary objects are deployed from said object library into a design framework to create said computer application.


32 Claims, 2 Drawing Sheets



In The Meier Book, **Page 19**, 4th paragraph, it shows that to create Android Applications , all that its needed is a Text Editor, or anything that can create text files:

Developing for Android | 19


You can download the latest version of the SDK for your development platform from the Android development homepage at <http://developer.android.com/sdk/index.html>

 Unless otherwise noted, the version of the Android SDK used for writing this book was version 2.1 r1.

The SDK is presented as a ZIP file containing only the latest version of the Android developer tools. Install it by unzipping the SDK into a new folder. (Take note of this location, as you'll need it later.)

Before you can begin development you need to add at least one SDK Platform; do this on Windows by running the "SDK Setup.exe" executable, or on MacOS or Linux by running the "android" executable in the tools subfolder. In the screen that appears, select the "Available Packages" option on the left panel, and then select the SDK Platform versions you wish to install in the "Sources, Packages, and Archives" panel on the right. The selected platform will then be downloaded to your SDK installation

The examples and step-by-step instructions provided are targeted at developers using Eclipse with the Android Developer Tool (ADT) plug-in. Neither is required, though — you can use any text editor or Java IDE you're comfortable with and use the developer tools in the SDK to compile, test, and debug the code snippets and sample applications.

 The examples included in the SDK are well documented and are an excellent source for full, working examples of applications written for Android. Once you've finished setting up your development environment it's worth going through them.

Developing with Eclipse

Using Eclipse with the ADT plug-in for your Android development offers some significant advantages.

Eclipse is an open-source IDE (integrated development environment) particularly popular for Java development. It's available for download for each of the development platforms supported by Android (Windows, MacOS, and Linux) from the Eclipse foundation homepage: www.eclipse.org/downloads/

There are many variations available; the following is the recommended configuration for Android:

- Eclipse 3.4 or 3.5 (Galileo)
- Eclipse JDT plug-in
- WST

WST and the JDT plug-in are included in most Eclipse IDE packages.

Using the Linux Portion of the Android Framework which includes built-in command line commands such as CAT, VI, CP, MV and RM, One can Create Arbitrary Objects, Manage Arbitrary Objects and Deploy these Arbitrary Objects.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page xvii** (more specifically 4th paragraph) and Page 4 (more specifically 7th bullet) essentially describes what is known in the industry as a framework.

INTRODUCTION

Now is an exciting time for mobile developers. Mobile phones have never been more popular, and powerful smartphones are now a popular choice for consumers. Stylish and versatile phones packing hardware features like GPS, accelerometers, and touch screens, combined with fixed-rate, reasonably priced data plans provide an enticing platform upon which to create innovative mobile applications.

A host of Android handsets are now available to tempt consumers, including phones with QVGA screens and powerful WVGA devices like the Motorola Droid and the Google Nexus One. The real win though, is for developers. With much existing mobile development built on proprietary operating systems that restrict the development and deployment of third-party applications, Android offers an open alternative. Without artificial barriers, Android developers are free to write applications that take full advantage of increasingly powerful mobile hardware and distribute them in an open market.

As a result, developer interest in Android devices has exploded as handset sales have continued to grow. In 2009 and the early parts of 2010 more than 20 Android handsets have been released from OEMs including HTC, Motorola, LG, Samsung, and Sony Ericsson. Android devices are now available in over

Built on an open source framework, and featuring powerful SDK libraries and an open philosophy, Android has opened mobile phone development to thousands of developers who haven't had access to tools for building mobile applications. Experienced mobile developers can now expand into the Android platform, leveraging the unique features to enhance existing products or create innovative new ones.

with no review process, for distributing free and paid apps to all compatible Android devices.

This book is a hands-on guide to building mobile applications using version 2 of the Android software development kit. Chapter by chapter, it takes you through a series of sample projects, each introducing new features and techniques to get the most out of Android. It covers all the basic functionality as well as exploring the advanced features through concise and useful examples.

Google's philosophy is to release early and iterate often. Since Android's first full release in October 2008, there have been seven platform and SDK releases. With such a rapid release cycle, there are likely to be regular changes and improvements to the software and development libraries. While the Android engineering team has worked hard to ensure backwards compatibility, future releases are likely to date some of the information provided in this book.

Nonetheless, the explanations and examples included here will give you the grounding and knowledge needed to write compelling mobile applications using the current SDK, along with the flexibility to quickly adapt to future enhancements.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, Page xvii (more specifically 4th paragraph) and **Page 4** (more specifically 7th bullet) essentially describes what is known in the industry as a framework.

4 | CHAPTER 1 HELLO, ANDROID

stack produced and supported by the Open Handset Alliance and designed to operate on any handset that meets the requirements. Google has now released its first direct-to-consumer handset, the Nexus 1, but this device remains simply one hardware implementation running on the Android platform.

ANDROID: AN OPEN PLATFORM FOR MOBILE DEVELOPMENT

Google's Andy Rubin describes Android as:

The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation. (<http://googleblog.blogspot.com/2007/11/whereas-my-gphone.html>)

Put simply, Android is a combination of three components:

- A free, open-source operating system for mobile devices
- An open-source development platform for creating mobile applications
- Devices, particularly mobile phones, that run the Android operating system and the applications created for it

More specifically, Android is made up of several necessary and dependent parts, including the following:

- A hardware reference design that describes the capabilities required for a mobile device to support the software stack.

A run time used to execute and host Android applications, including the Dalvik virtual machine and the core libraries that provide Android-specific functionality. The run time is designed to be small and efficient for use on mobile devices.

- An application framework that agnostically exposes system services to the application layer, including the window manager and location manager, content providers, telephony, and sensors.
- A user interface framework used to host and launch applications.
- Preinstalled applications shipped as part of the stack.
- A software development kit used to create applications, including tools, plug-ins, and documentation.

What really makes Android compelling is its open philosophy, which ensures that you can fix any deficiencies in user interface or native application design by writing an extension or replacement. Android

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page 28** explains how the UI (Form) "decouples" the application logic (function) from your UI design (form) on its first paragraph.

28 | CHAPTER 2 GETTING STARTED

Listing 2-2 shows the UI layout defined in the main.xml file created by the Android project template.

LISTING 2-2: Hello World layout resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
```

Defining your UI in XML and inflating it is the preferred way of implementing your user interfaces, as it neatly decouples your application logic from your UI design.

To get access to your UI elements in code, you add identifier attributes to them in the XML definition. You can then use the `findViewById` method to return a reference to each named item. The following XML snippet shows an ID attribute added to the Text View widget in the Hello World template:

```
<TextView
    android:id="@+id/myTextView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World, HelloWorld"
/>
```

And the following snippet shows how to get access to it in code:

```
TextView myTextView = (TextView) findViewById(R.id.myTextView);
```

Alternatively (although it's not generally considered good practice), you can create your layout directly in code, as shown in Listing 2-3.

LISTING 2-3: Creating layouts in code

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    LinearLayout.LayoutParams lp;
    lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.FILL_PARENT);

    LinearLayout.LayoutParams textViewLP;
    textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);

    TextView myTextView = new TextView(this);
    myTextView.setText("Hello World, HelloWorld");
```


Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

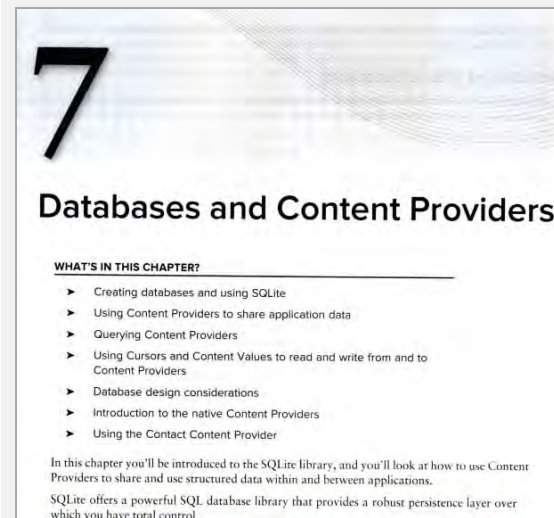
a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page 209** defines how the "Content Providers" "decouples" data storage (content) from your application logic (function) on its third paragraph.



Content Providers offer a generic interface to any data source by decoupling the data storage layer from the application layer.

INTRODUCING ANDROID DATABASES

Structured data persistence in Android is provided through the following mechanisms:

- **SQLite Databases** When managed, structured data is the best approach, Android offers the SQLite relational database library. Every application can create its own databases over which it has complete control.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

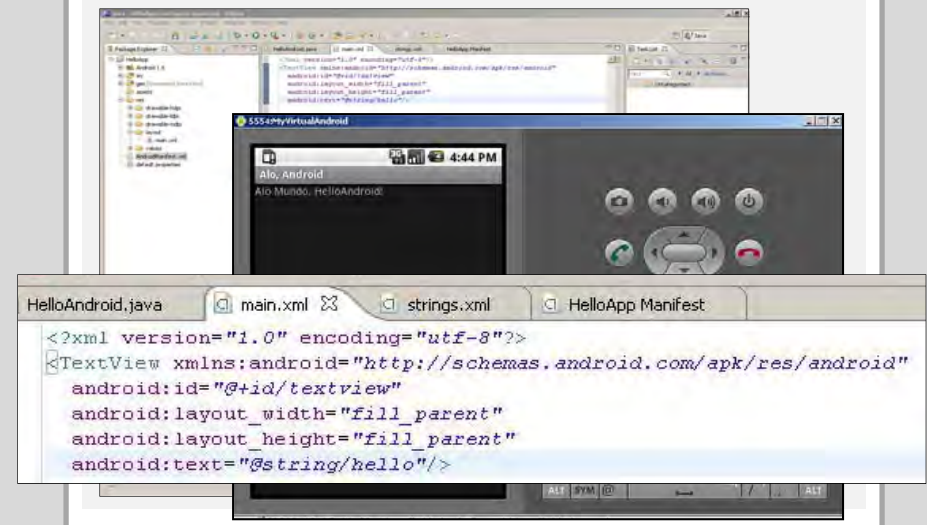
and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android's objects can be arbitrary objects. We created a sample application using the Android Software Developers' Kit (Android SDK). In the following example:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

The Arbitrary Object "textview" is being called by its name (android:id="@+id/textview"), and additionally has three parameters.

When compiling and running the application, the compilation completes, and the application runs fine (it prints "Alo mundo,Hello Android!") on the Android device as seen in here:



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

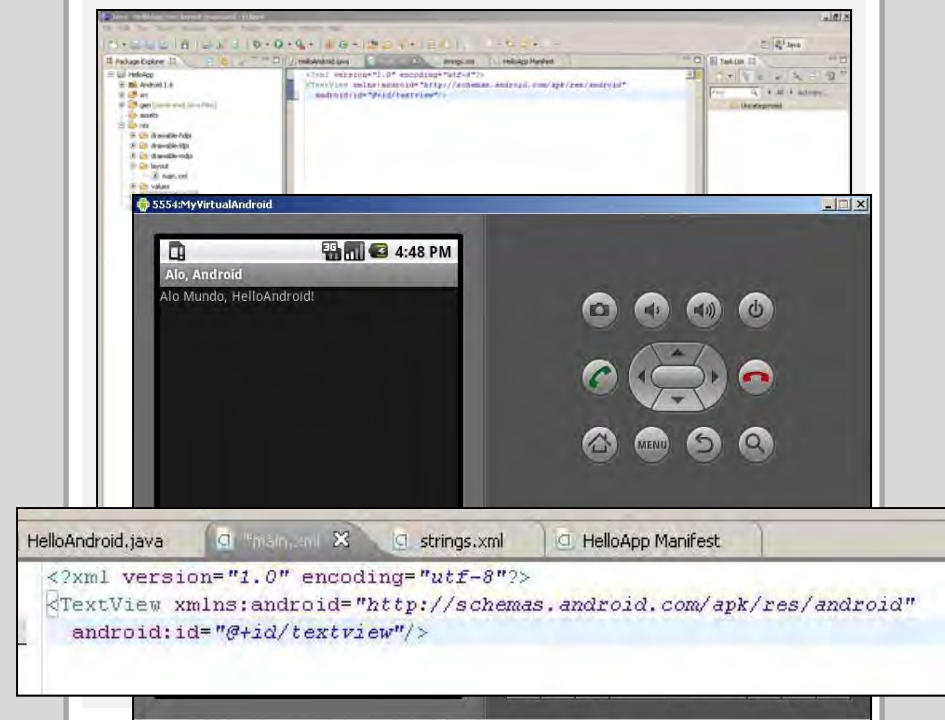
a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

However, when we modify the call to “textview” to include just the name of the Arbitrary Object, the call looks this way:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
/>
```

When making the above modification, the program compiles and the application runs normally, as can be seen on this screen print (We will refer to this later as “Sample Application”)



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The creation of those Arbitrary objects can be using command line tools, such as a text editor, or using the recommended tool , Eclipse. Using a text editor is more work because all of the files needed to create the object must be edited correctly and placed on the correct location (folders), in order for the compiler to work. Using Eclipse facilitates the process

The steps for generating a Computer application on the Android Framework when not using Eclipse are:

- Creation of the application either by manually creating each source file with a text editor
- Compiling of the application using a Java Compiler
- Creating a "package" (A file which's extension is .APK", using the Android SDK named apkbuilder.bat , which is distributed with the Android SDK.
- Moving that package file to the Android device, either by using Google's "Android Market" application (in the case of a SmartPhone), or using the builtin android application named "File Manager", clicking on the file, and Android will then install it on the device.

When using eclipse, all is necessary is to click on the "Run" menu – that is what was done on the application described earlier.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

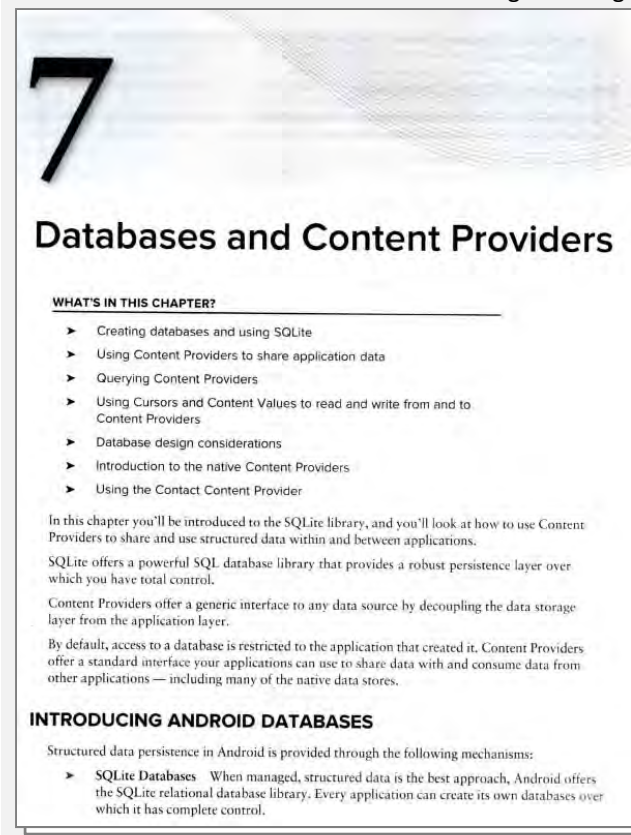
A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects **used in generating said content of said computer application**, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The content can be generated by the “Content Provider” feature of Android. Meier Book, **Pages 209-211** is a description of how it works. The title of the chapter, and throughout the whole text, it clarifies that the Android Framework can be used for generating the content:



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

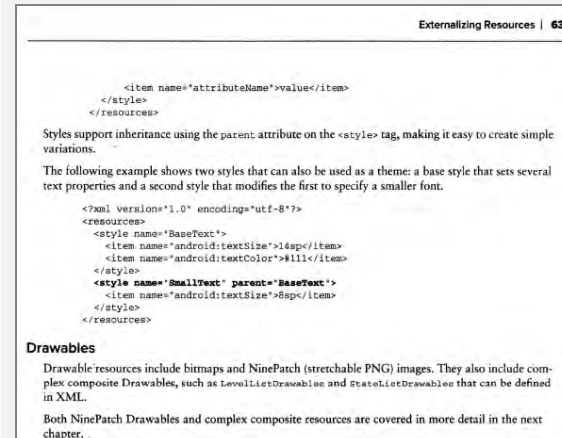
A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, **form objects used in defining said form of said computer application**, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Form in the Android Framework normally is defined on the Resource folder, referred to as "R" or "Layouts", having the default "layout.xml" as the Form, but this name can be changed. On Meier Book, **Page 63** and 64 describes how they work. More specifically, the last three paragraphs of **Page 63** define Layouts as your "Presentation Layer" and "User Interface", which is Form.



Layout resources let you decouple your presentation layer by designing user interface layouts in XML rather than constructing them in code.

The most common use of a layout is for defining the user interface for an Activity. Once defined in XML, the layout is "inflated" within an Activity using setContentView, usually within the onCreate method. You can also reference layouts from within other layout resources, such as layouts for each row in a List View. More detailed information on using and creating layouts in Activities can be found in Chapter 4.

Using layouts to create your screens is best-practice UI design in Android. The decoupling of the layout from the code lets you create optimized layouts for different hardware configurations, such as varying screen sizes, orientation, or the presence of keyboards and touchscreens.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, **form objects used in defining said form of said computer application**, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Also on Meier Book, **Page 50**, it describes “Activities” as Presentation Layer (which can be Form) of Android Applications.

50 | CHAPTER 3 CREATING APPLICATIONS AND ACTIVITIES

Mobile devices come in a large variety of shapes and sizes and are used across the world. In this chapter you'll learn how to externalize resources to ensure your applications run seamlessly on different hardware (particularly different screen resolutions and pixel densities), in different countries, and supporting multiple languages.

Next you'll examine the Application class, and learn how to extend it to provide a place for storing application state values.

Arguably the most important of the Android building blocks, the Activity class forms the basis for all your user interface screens. You'll learn how to create new Activities and gain an understanding of their life cycles and how they affect the application lifetime.

Finally, you'll be introduced to some of the Activity subclasses that simplify resource management for some common user interface components such as maps and lists.

WHAT MAKES AN ANDROID APPLICATION?

Android applications consist of loosely coupled components, bound by an application manifest that describes each component and how they all interact, as well as the application metadata including its

Activities Your application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an Activity is equivalent to a Form. You'll learn more about Activities later in this chapter.

tions. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 9.

- **Content Providers** Shareable data stores. Content Providers are used to manage and share application databases. They're the preferred means of sharing data across application boundaries. This means that you can configure your own Content Providers to permit access from other applications and use Content Providers exposed by others to access their stored data. Android devices include several native Content Providers that expose useful databases like the media store and contact details. You'll learn how to create and use Content Providers in Chapter 7.
- **Intents** An inter-application message-passing framework. Using Intents you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- **Broadcast Receivers** Intent broadcast consumers. If you create and register a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Meier Book, **Page 50**, second bullet, describes “Services” as “The invisible workers of the application”, (Which is Functionality):

90 CHAPTER 3 CREATING APPLICATIONS AND ACTIVITIES

Mobile devices come in a large variety of shapes and sizes and are used across the world. In this chapter you'll learn how to externalize resources to ensure your applications run seamlessly on different hardware (particularly different screen resolutions and pixel densities), in different countries, and supporting multiple languages.

Next you'll examine the Application class, and learn how to extend it to provide a place for storing application state values.

Arguably the most important of the Android building blocks, the Activity class forms the basis for all your user interface screens. You'll learn how to create new Activities and gain an understanding of their life cycles and how they affect the application lifetime.

Finally, you'll be introduced to some of the Activity subclasses that simplify resource management for some common user interface components such as maps and lists.

WHAT MAKES AN ANDROID APPLICATION?

Android applications consist of loosely coupled components, bound by an application manifest that describes each component and how they all interact, as well as the application metadata including its hardware and platform requirements.

The following six components provide the building blocks for your applications:

- **Activities** Your application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an Activ-

Services The invisible workers of your application. Service components run in the background, updating your data sources and visible Activities and triggering Notifications. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 9.

Android devices include several native Content Providers that expose useful databases like the media store and contact details. You'll learn how to create and use Content Providers in Chapter 7.

- **Intents** An inter-application message-passing framework. Using Intents you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- **Broadcast Receivers** Intent broadcast consumers. If you create and register a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The object library, when using eclipse is described as the "Workspace" of the Android Applications. If not using Eclipse, the management of these objects can be done manually by using normal command-line commands, such as "Copy", "Rename", "Delete" or "Move", inside of a folder structure which is mandated by the Android SDK so the Computer Application can be built properly.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

These Arbitrary objects are deployed by building the complete application as described earlier then moving that package file to the Android device, either by using Google's "Android Market" application (in the case of a SmartPhone), or using the builtin android application named "File Manager", clicking on the file, and Android will then install it on the device.

The Meier Book, page XVII, on its 5th Paragraph, describes that the Android Market can be used for distribution, or Deployment:

INTRODUCTION

Now is an exciting time for mobile developers. Mobile phones have never been more popular, and powerful smartphones are now a popular choice for consumers. Stylish and versatile phones packing hardware features like GPS, accelerometers, and touch screens, combined with fixed-rate, reasonably priced data plans provide an enticing platform upon which to create innovative mobile applications.

A host of Android handsets are now available to tempt consumers, including phones with QVGA screens and powerful WVGA devices like the Motorola Droid and the Google Nexus One. The real win though, is for developers. With much existing mobile development built on proprietary operating systems that restrict the development and deployment of third-party applications, Android offers an open alternative. Without artificial barriers, Android developers are free to write applications that take full advantage of increasingly powerful mobile hardware and distribute them in an open market.

As a result, developer interest in Android devices has exploded as handset sales have continued to grow. In 2009 and the early parts of 2010 more than 20 Android handsets have been released from OEMs including HTC, Motorola, LG, Samsung, and Sony Ericsson. Android devices are now available in over 26 countries on more than 32 carriers. In the United States, Android devices are available on all four major carriers: T-Mobile, Verizon, AT&T, and Sprint. Additionally, you can now buy the unlocked Google Nexus One handset directly from Google at <http://www.google.com/phone>.

Built on an open source framework, and featuring powerful SDK libraries and an open philosophy, Android has opened mobile phone development to thousands of developers who haven't had access to tools for building mobile applications. Experienced mobile developers can now expand into the

Using the Android Market for distribution, developers can take advantage of an open marketplace, with no review process, for distributing free and paid apps to all compatible Android devices.

This book is a manual on how to building mobile applications using version 2 of the Android software development kit. Chapter by chapter, it takes you through a series of sample projects, each introducing new features and techniques to get the most out of Android. It covers all the basic functionality as well as exploring the advanced features through concise and useful examples.

Google's philosophy is to release early and iterate often. Since Android's first full release in October 2008, there have been seven platform and SDK releases. With such a rapid release cycle, there are likely to be regular changes and improvements to the software and development libraries. While the Android engineering team has worked hard to ensure backwards compatibility, future releases are likely to date some of the information provided in this book.

Nonetheless, the explanations and examples included here will give you the grounding and knowledge needed to write compelling mobile applications using the current SDK, along with the flexibility to quickly adapt to future enhancements.

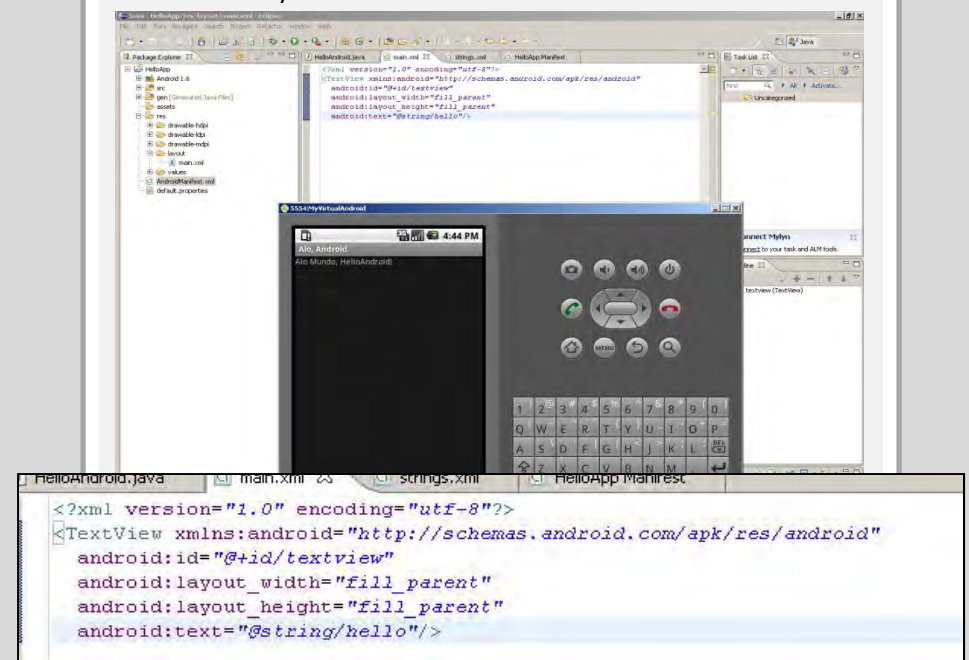
Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 4.

The system of claim 1, wherein the third set of executable instructions are for deploying arbitrary objects locally.

The Computer Application demonstrated on Claim 1, Element 1, makes use of arbitrary objects locally.

Bellow Picture, shows in the arbitrary object named textview being accessed locally within the mobile device.

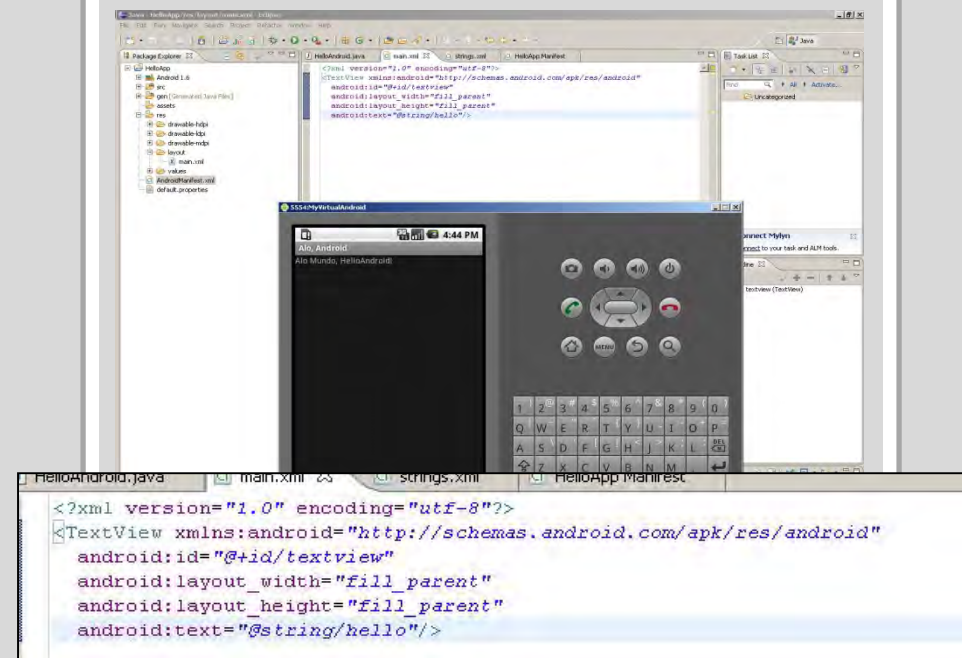


Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 8.

The system of claim 1, wherein the third set of executable instructions include instructions to access and deploy arbitrary objects into said design framework using said corresponding arbitrary names.

The Android Framework enables access and deployment of said arbitrary names into said design framework. Bellow picture shows the deployment of said arbitrary objects (the arbitrary object named "textview") accessed and deployed into said design framework with the said arbitrary names (textview):

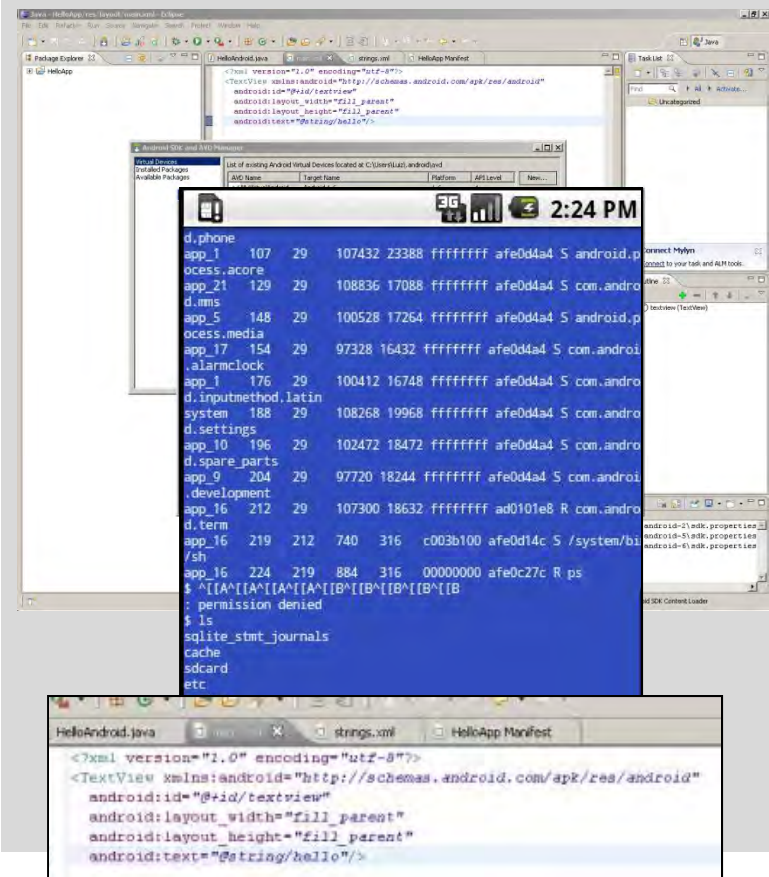


Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 10.

The system of claim 1, further comprising executable instructions for caching objects.

The Android Framework caches objects. In this picture showing the Android terminal emulation application, the response to the command "ls" shows a subdirectory named "cache":



```
d,phone
app_1 107 29 107432 23388 ffffffff afe0d4a4 S android.p
ocess.acore
app_21 129 29 108836 17088 ffffffff afe0d4a4 S com.andro
d.mms
app_5 148 29 100528 17264 ffffffff afe0d4a4 S android.p
ocess.media
app_17 154 29 97328 16432 ffffffff afe0d4a4 S com.andro
,alarmclock
app_1 176 29 100412 16748 ffffffff afe0d4a4 S com.andro
d.inputmethod.latin
system 188 29 108268 19968 ffffffff afe0d4a4 S com.andro
d.settings
app_10 196 29 102472 18472 ffffffff afe0d4a4 S com.andro
d.spare_parts
app_9 204 29 97720 18244 ffffffff afe0d4a4 S com.andro
,development
app_16 212 29 107300 18632 ffffffff ad0101e8 R com.andro
d.term
app_16 219 212 740 316 c003b100 afe0d14c S /system/bi
/sh
app_16 224 219 884 316 00000000 afe0c27c R ps
$ ^[[A^[[A^[[A^[[A^[[B^[[B^[[B^[[B^[[B
: permission denied
$ ls
sqlite_stmt_journals
cache
sdcard
etc
```

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 12.

The system of claim 1, further comprising executable instructions for generating arbitrary objects in a programming language that is compatible and supported by said host system.

The Android Framework allows such arbitrary objects to be developed in Java, which is a programming language supported by the host system. This can be shown in "Meyer Book", page 28 (below), the program named "LISTING 2-3", written in Java:

28 | CHAPTER 2 GETTING STARTED

Listing 2-2 shows the UI layout defined in the main.xml file created by the Android project template.



LISTING 2-2: Hello World layout resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Hello World, HelloWorld"
    />
</LinearLayout>
```

Defining your UI in XML and inflating it is the preferred way of implementing your user interfaces, as it neatly decouples your application logic from your UI design.

To get access to your UI elements in code, you add identifier attributes to them in the XML definition. You can then use the findViewById method to return a reference to each named item. The following XML snippet shows an ID attribute added to the Text View widget in the Hello World template:

LISTING 2-3: Creating layouts in code

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    LinearLayout.LayoutParams lp;
    lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
                                      LinearLayout.LayoutParams.FILL_PARENT);

    LinearLayout.LayoutParams textViewLP;
    textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
                                              LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);

    TextView myTextView = new TextView(this);
    myTextView.setText("Hello World, HelloWorld");
}
```

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page xvii** (more specifically 4th paragraph) and Page 4 (more specifically 7th bullet) essentially describes what is known in the industry as a framework.

INTRODUCTION

Now is an exciting time for mobile developers. Mobile phones have never been more popular, and powerful smartphones are now a popular choice for consumers. Stylish and versatile phones packing hardware features like GPS, accelerometers, and touch screens, combined with fixed-rate, reasonably priced data plans provide an enticing platform upon which to create innovative mobile applications.

A host of Android handsets are now available to tempt consumers, including phones with QVGA screens and powerful WVGA devices like the Motorola Droid and the Google Nexus One. The real win though, is for developers. With much existing mobile development built on proprietary operating systems that restrict the development and deployment of third-party applications, Android offers an open alternative. Without artificial barriers, Android developers are free to write applications that take full advantage of increasingly powerful mobile hardware and distribute them in an open market.

As a result, developer interest in Android devices has exploded as handset sales have continued to grow. In 2009 and the early parts of 2010 more than 20 Android handsets have been released from OEMs including HTC, Motorola, LG, Samsung, and Sony Ericsson. Android devices are now available in over

Built on an open source framework, and featuring powerful SDK libraries and an open philosophy, Android has opened mobile phone development to thousands of developers who haven't had access to tools for building mobile applications. Experienced mobile developers can now expand into the Android platform, leveraging the unique features to enhance existing products or create innovative new ones.

with no review process, for distributing free and paid apps to all compatible Android devices.

This book is a hands-on guide to building mobile applications using version 2 of the Android software development kit. Chapter by chapter, it takes you through a series of sample projects, each introducing new features and techniques to get the most out of Android. It covers all the basic functionality as well as exploring the advanced features through concise and useful examples.

Google's philosophy is to release early and iterate often. Since Android's first full release in October 2008, there have been seven platform and SDK releases. With such a rapid release cycle, there are likely to be regular changes and improvements to the software and development libraries. While the Android engineering team has worked hard to ensure backwards compatibility, future releases are likely to date some of the information provided in this book.

Nonetheless, the explanations and examples included here will give you the grounding and knowledge needed to write compelling mobile applications using the current SDK, along with the flexibility to quickly adapt to future enhancements.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, Page xvii (more specifically 4th paragraph) and **Page 4** (more specifically 7th bullet) essentially describes what is known in the industry as a framework.

4 | CHAPTER 1 HELLO, ANDROID

stack produced and supported by the Open Handset Alliance and designed to operate on any handset that meets the requirements. Google has now released its first direct-to-consumer handset, the Nexus 1, but this device remains simply one hardware implementation running on the Android platform.

ANDROID: AN OPEN PLATFORM FOR MOBILE DEVELOPMENT

Google's Andy Rubin describes Android as:

The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation. (<http://googleblog.blogspot.com/2007/11/whereas-my-gphone.html>)

Put simply, Android is a combination of three components:

- A free, open-source operating system for mobile devices
- An open-source development platform for creating mobile applications
- Devices, particularly mobile phones, that run the Android operating system and the applications created for it

More specifically, Android is made up of several necessary and dependent parts, including the following:

- A hardware reference design that describes the capabilities required for a mobile device to support the software stack.

A run time used to execute and host Android applications, including the Dalvik virtual machine and the core libraries that provide Android-specific functionality. The run time is designed to be small and efficient for use on mobile devices.

- An application framework that agnostically exposes system services to the application layer, including the window manager and location manager, content providers, telephony, and sensors.
- A user interface framework used to host and launch applications.
- Preinstalled applications shipped as part of the stack.
- A software development kit used to create applications, including tools, plug-ins, and documentation.

What really makes Android compelling is its open philosophy, which ensures that you can fix any deficiencies in user interface or native application design by writing an extension or replacement. Android

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page 28** explains how the UI (Form) "decouples" the application logic (function) from your UI design (form) on its first paragraph.

28 | CHAPTER 2 GETTING STARTED

Listing 2-2 shows the UI layout defined in the main.xml file created by the Android project template.

LISTING 2-2: Hello World layout resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
```

Defining your UI in XML and inflating it is the preferred way of implementing your user interfaces, as it neatly decouples your application logic from your UI design.

To get access to your UI elements in code, you add identifier attributes to them in the XML definition. You can then use the `findViewById` method to return a reference to each named item. The following XML snippet shows an ID attribute added to the Text View widget in the Hello World template:

```
<TextView
    android:id="@+id/myTextView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World, HelloWorld"
/>
```

And the following snippet shows how to get access to it in code:

```
TextView myTextView = (TextView) findViewById(R.id.myTextView);
```

Alternatively (although it's not generally considered good practice), you can create your layout directly in code, as shown in Listing 2-3.

LISTING 2-3: Creating layouts in code

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    LinearLayout.LayoutParams lp;
    lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.FILL_PARENT);

    LinearLayout.LayoutParams textViewLP;
    textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);

    TextView myTextView = new TextView(this);
    myTextView.setText("Hello World, HelloWorld");
```


Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

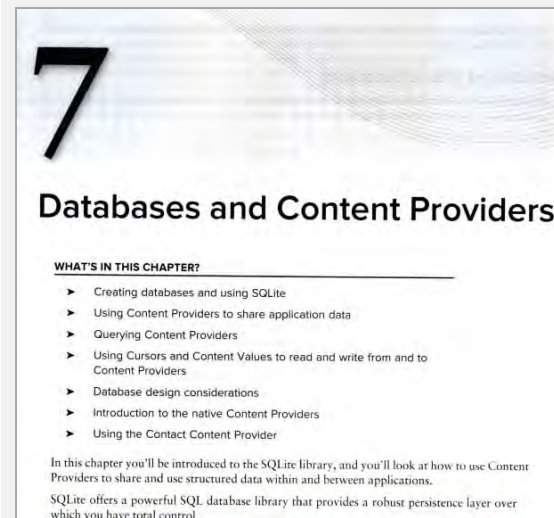
a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page 209** defines how the "Content Providers" "decouples" data storage (content) from your application logic (function) on its third paragraph.



Content Providers offer a generic interface to any data source by decoupling the data storage layer from the application layer.

INTRODUCING ANDROID DATABASES

Structured data persistence in Android is provided through the following mechanisms:

- **SQLite Databases** When managed, structured data is the best approach, Android offers the SQLite relational database library. Every application can create its own databases over which it has complete control.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

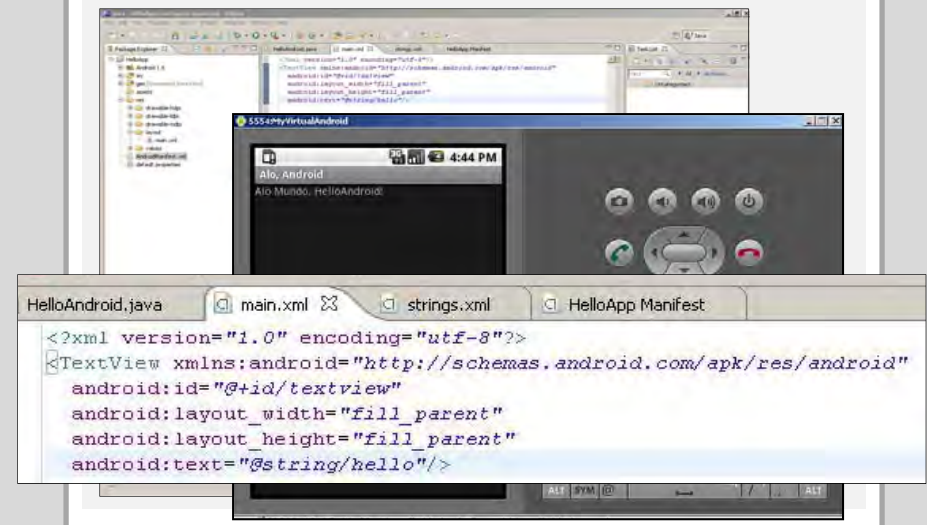
and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Android's objects can be arbitrary objects. We created a sample application using the Android Software Developers' Kit (Android SDK). In the following example:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

The Arbitrary Object "textview" is being called by its name (android:id="@+id/textview"), and additionally has three parameters.

When compiling and running the application, the compilation completes, and the application runs fine (it prints "Alo mundo,Hello Android!") on the Android device as seen in here:



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

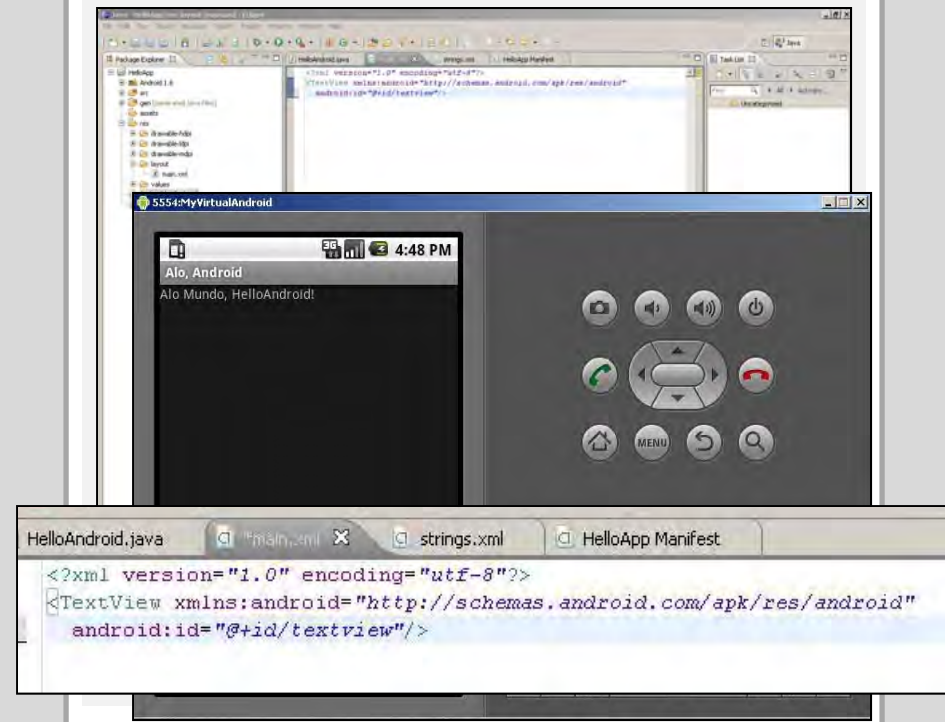
a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

However, when we modify the call to "textview" to include just the name of the Arbitrary Object, the call looks this way:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
/>
```

When making the above modification, the program compiles and the application runs normally, as can be seen on this screen print (We will refer to this later as "Sample Application")



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The creation of those Arbitrary objects can be using command line tools, such as a text editor, or using the recommended tool, Eclipse. Using a text editor is a more work because all of the files needed to create the object need to be edited correctly and placed on the correct location (folders), in order for the compiler to work. Using Eclipse facilitates the process

The steps for generating a Computer application on the Android Framework when not using Eclipse are:

- Creation of the application either by manually creating each source file with a text editor
- Compiling of the application using a Java Compiler
- Creating a "package" (A file which's extension is .APK", using the Android SDK named apkbuilder.bat, which is distributed with the Android SDK.
- Moving that package file to the Android device, either by using Google's "Android Market" application (in the case of a SmartPhone), or using the built in android application named "File Manager", clicking on the file, and Android will then install it on the device.

When using eclipse, all is necessary is to click on the "Run" menu – that is what was done on the application described earlier.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

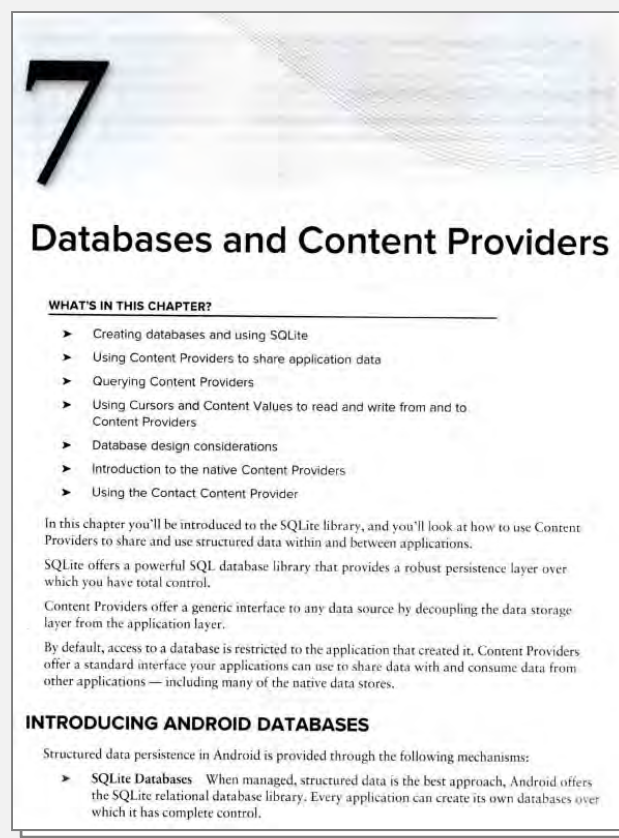
A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names **of content objects used in generating said content of said computer application**, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The content can be generated by the “Content Provider” feature of Android. Meier Book, **Pages 209-211** is a description of how it works. The title of the chapter, and throughout the whole text, it clarifies that the Android Framework can be used for generating the content:



Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

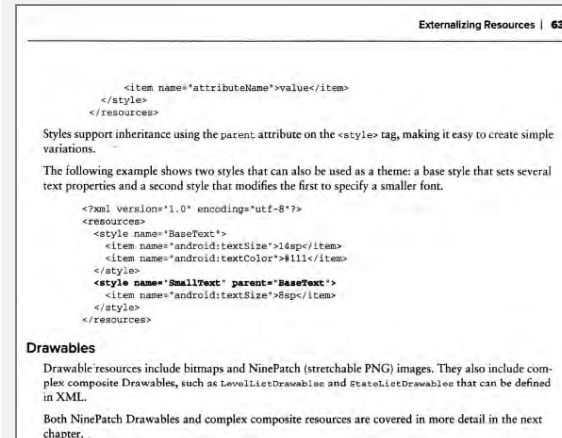
A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, **form objects used in defining said form of said computer application**, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Form in the Android Framework normally is defined on the Resource folder, referred to as "R" or "Layouts", having the default "layout.xml" as the Form, but this name can be changed. On Meier Book, **Page 63** and 64 describes how they work. More specifically, the last three paragraphs of **Page 63** define Layouts as your "Presentation Layer" and "User Interface", which is Form.



Layout resources let you decouple your presentation layer by designing user interface layouts in XML rather than constructing them in code.

The most common use of a layout is for defining the user interface for an Activity. Once defined in XML, the layout is "inflated" within an Activity using setContentView, usually within the onCreate method. You can also reference layouts from within other layout resources, such as layouts for each row in a List View. More detailed information on using and creating layouts in Activities can be found in Chapter 4.

Using layouts to create your screens is best-practice UI design in Android. The decoupling of the layout from the code lets you create optimized layouts for different hardware configurations, such as varying screen sizes, orientation, or the presence of keyboards and touchscreens.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, **form objects used in defining said form of said computer application**, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Also on Meier Book, **Page 50**, it describes “Activities” as Presentation Layer (which can be Form) of Android Applications.

50 | CHAPTER 3 CREATING APPLICATIONS AND ACTIVITIES

Mobile devices come in a large variety of shapes and sizes and are used across the world. In this chapter you'll learn how to externalize resources to ensure your applications run seamlessly on different hardware (particularly different screen resolutions and pixel densities), in different countries, and supporting multiple languages.

Next you'll examine the Application class, and learn how to extend it to provide a place for storing application state values.

Arguably the most important of the Android building blocks, the Activity class forms the basis for all your user interface screens. You'll learn how to create new Activities and gain an understanding of their life cycles and how they affect the application lifetime.

Finally, you'll be introduced to some of the Activity subclasses that simplify resource management for some common user interface components such as maps and lists.

WHAT MAKES AN ANDROID APPLICATION?

Android applications consist of loosely coupled components, bound by an application manifest that describes each component and how they all interact, as well as the application metadata including its

Activities Your application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an Activity is equivalent to a Form. You'll learn more about Activities later in this chapter.

tions. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 9.

- **Content Providers** Shareable data stores. Content Providers are used to manage and share application databases. They're the preferred means of sharing data across application boundaries. This means that you can configure your own Content Providers to permit access from other applications and use Content Providers exposed by others to access their stored data. Android devices include several native Content Providers that expose useful databases like the media store and contact details. You'll learn how to create and use Content Providers in Chapter 7.
- **Intents** An inter-application message-passing framework. Using Intents you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- **Broadcast Receivers** Intent broadcast consumers. If you create and register a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Meier Book, **Page 50**, second bullet, describes “Services” as “The invisible workers of the application”, (Which is Functionality):

90 CHAPTER 3 CREATING APPLICATIONS AND ACTIVITIES

Mobile devices come in a large variety of shapes and sizes and are used across the world. In this chapter you'll learn how to externalize resources to ensure your applications run seamlessly on different hardware (particularly different screen resolutions and pixel densities), in different countries, and supporting multiple languages.

Next you'll examine the `Application` class, and learn how to extend it to provide a place for storing application state values.

Arguably the most important of the Android building blocks, the `Activity` class forms the basis for all your user interface screens. You'll learn how to create new Activities and gain an understanding of their life cycles and how they affect the application lifetime.

Finally, you'll be introduced to some of the Activity subclasses that simplify resource management for some common user interface components such as maps and lists.

WHAT MAKES AN ANDROID APPLICATION?

Android applications consist of loosely coupled components, bound by an application manifest that describes each component and how they all interact, as well as the application metadata including its hardware and platform requirements.

The following six components provide the building blocks for your applications:

- **Activities** Your application's presentation layer. Every screen in your application will be an extension of the `Activity` class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an `Activity`

Services The invisible workers of your application. Service components run in the background, updating your data sources and visible Activities and triggering Notifications. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 9.

Android devices include several native Content Providers that expose useful databases like the media store and contact details. You'll learn how to create and use Content Providers in Chapter 7.

- **Intents** An inter-application message-passing framework. Using Intents you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- **Broadcast Receivers** Intent broadcast consumers. If you create and register a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, **each arbitrary object being callable by name only**, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The Sample Application demonstrated in Claim 21, Element 1, calls the arbitrary object by name only.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

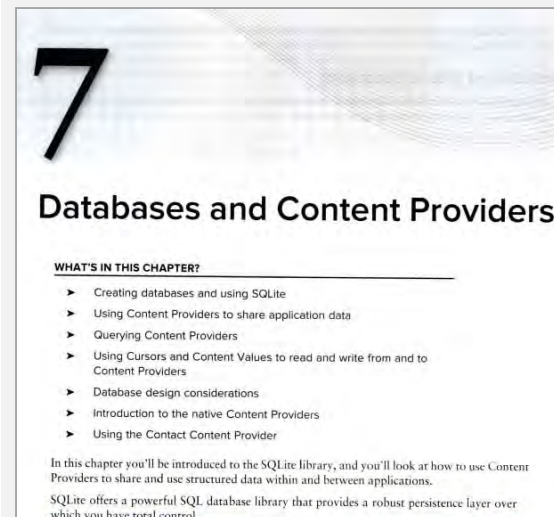
A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, **each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object**, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

In the Meier Book, there are many references of “decoupling” form from content, and from form, as a way of showing how these objects can be independently modified without changes to the other arbitrary objects. The Meier book, on **page 209**, defines how the “Content Providers” decouples data storage from your application logic on its third paragraph:



Content Providers offer a generic interface to any data source by decoupling the data storage layer from the application layer.

INTRODUCING ANDROID DATABASES

Structured data persistence in Android is provided through the following mechanisms:

- **SQLite Databases** When managed, structured data is the best approach, Android offers the SQLite relational database library. Every application can create its own databases over which it has complete control.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, **each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object**, and each arbitrary object further being interchangeable with other arbitrary objects;

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

The Meier Book, **Page 63**, under the headline named “Layouts”, defines how layouts decouple your user interface from your code:

Externalizing Resources | 63

```
<item name="@attributeName">value</item>
</style>
</resources>
```

Styles support inheritance using the parent attribute on the <style> tag, making it easy to create simple variations.

The following example shows two styles that can also be used as a theme: a base style that sets several text properties and a second style that modifies the first to specify a smaller font.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="BaseText">
    <item name="android:textSize">14sp</item>
    <item name="android:textColor">#111</item>
  </style>
  <style name="SmallText" parent="BaseText">
    <item name="android:textSize">8sp</item>
  </style>
</resources>
```

Drawables

Drawable resources include bitmaps and NinePatch (stretchable PNG) images. They also include complex composite Drawables, such as LevelListDrawables and StateListDrawables that can be defined in XML.

Both NinePatch Drawables and complex composite resources are covered in more detail in the next chapter.

All Drawables are stored as individual files in the `res/drawable` folder. The resource identifier for a Drawable resource is the lowercase file name without an extension.

Layout resources let you decouple your presentation layer by designing user interface layouts in XML rather than constructing them in code.

The most common use of a layout is for defining the user interface for an Activity. Once defined in XML, the layout is “inflated” within an Activity using setContentView, usually within the onCreate method. You can also reference layouts from within other layout resources, such as layouts for each row in a List View. More detailed information on using and creating layouts in Activities can be found in Chapter 4.

Using layouts to create your screens is best-practice UI design in Android. The decoupling of the layout from the code lets you create optimized layouts for different hardware configurations, such as varying screen sizes, orientation, or the presence of keyboards and touchscreens.

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:

a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application, each arbitrary object being callable by name only, each arbitrary object being independently modifiable without corresponding modifications being made to any other arbitrary object, **and each arbitrary object further being interchangeable with other arbitrary objects;**

a second set of executable instructions for managing said arbitrary objects in an arbitrary object library;

and a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

Again in the Meier Book, **Page 63** in the last paragraph it explains how these arbitrary objects are interchangeable, for the benefit of having the application support Android devices of varying screen sizes, keyboards and such:

Externalizing Resources | 63

```
<item name="@attributeName">value</item>
</style>
</resources>
```

Styles support inheritance using the parent attribute on the <style> tag, making it easy to create simple variations.

The following example shows two styles that can also be used as a theme: a base style that sets several text properties and a second style that modifies the first to specify a smaller font.


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="BaseText">
    <item name="android:textSize">14sp</item>
    <item name="android:textColor">#111</item>
  </style>
  <style name="SmallText" parent="BaseText">
    <item name="android:textSize">8sp</item>
  </style>
</resources>
```

Drawables

Drawable resources include bitmaps and NinePatch (stretchable PNG) images. They also include complex composite Drawables, such as LevelListDrawables and StateListDrawables that can be defined in XML.

Both NinePatch Drawables and complex composite resources are covered in more detail in the next chapter.

All Drawables are stored as individual files in the `res/drawable` folder. The resource identifier for a Drawable resource is the lowercase file name without an extension.



The preferred format for a bitmap resource is PNG, although JPG and GIF files are also supported.

Layouts

Layout resources let you decouple your presentation layer by designing user interface layouts in XML rather than constructing them in code.

The most common use of a layout is for defining the user interface for an Activity. Once defined in XML, the layout is "inflated" within an Activity using `setContentView`, usually within the `onCreate` method. You can also reference layouts from within other layout resources, such as layouts for each

Using layouts to create your screens is best-practice UI design in Android. The decoupling of the layout from the code lets you create optimized layouts for different hardware configurations, such as varying screen sizes, orientation, or the presence of keyboards and touchscreens.

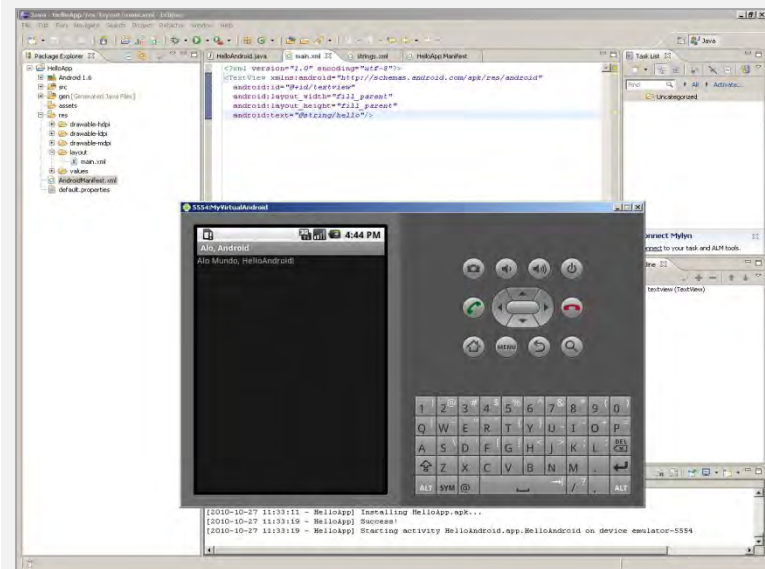
Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 24.

The system of claim 21, wherein the third set of executable instructions are for deploying arbitrary objects locally.

The Computer Application demonstrated on Claim 1, Element 1, makes use of arbitrary objects locally.

Below Picture, shows in the arbitrary object named textview being accessed locally within the mobile device.



```
main.xml
strings.xml
HelloAppManifest

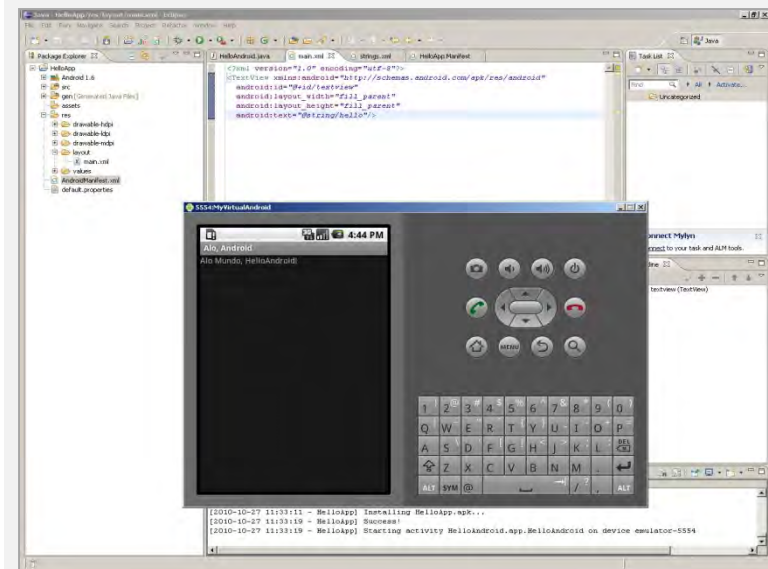
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 28.

The system of claim 21, wherein the third set of executable instructions include instructions to access and deploy arbitrary objects into said design framework using said corresponding arbitrary names.

The Android Framework enables access and deployment of said arbitrary names into said design framework. Bellow picture shows the deployment of said arbitrary objects (the arbitrary object named "textview") accessed and deployed into said design framework with the said arbitrary names (textview):



```

HelloApp.java  main.xml  strings.xml  HelloAppManifest
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 30.

The system of claim 21, further comprising executable instructions for caching objects.

Claim 30.

The system of claim 21, further comprising executable instructions for caching objects.

[illegible][illegible]

Infringement of United States Patent number 7,726,629 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 32.

The system of claim 21, further comprising executable instructions for generating arbitrary objects in a programming language that is compatible and supported by said host system.

The Android Framework allows such arbitrary objects to be developed in Java, which is a programming language supported by the host system. This can be shown in “Meyer Book”, **page 28** (below), the program named “LISTING 2-3”, written in Java:

28 | CHAPTER 2 GETTING STARTED

Listing 2-2 shows the UI layout defined in the main.xml file created by the Android project template.



LISTING 2-2: Hello World layout resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Hello World, HelloWorld"
    />
</LinearLayout>
```

Defining your UI in XML and inflating it is the preferred way of implementing your user interfaces, as it neatly decouples your application logic from your UI design.

To get access to your UI elements in code, you add identifier attributes to them in the XML definition. You can then use the `findViewById` method to return a reference to each named item. The following XML snippet shows an ID attribute added to the Text View widget in the Hello World template:

LISTING 2-3: Creating layouts in code

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);


    LinearLayout.LayoutParams lp;
    lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
                                      LinearLayout.LayoutParams.FILL_PARENT);

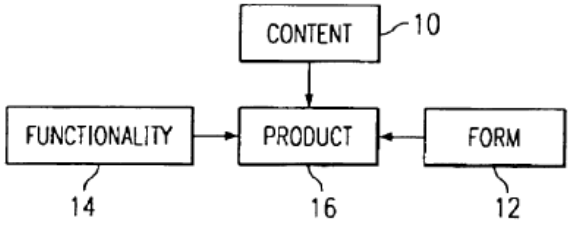
    LinearLayout.LayoutParams textViewLP;
    textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
                                                LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);

    TextView myTextView = new TextView(this);
    myTextView.setText("Hello World, HelloWorld");
}
```

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers


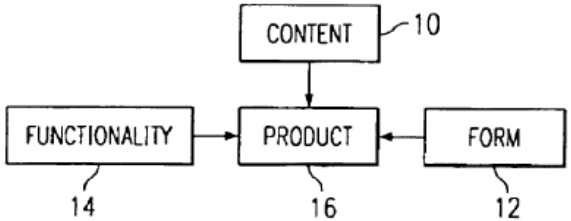
 US006826744B1	
(12) United States Patent McAuley	(10) Patent No.: US 6,826,744 B1 (45) Date of Patent: Nov. 30, 2004
 (54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	
(75) Inventor: Aubrey McAuley, Austin, TX (US)	6,026,433 A * 2/2000 D'Arlach et al. 707/10 6,128,998 A * 2/2000 Gloudeman et al. 717/108 6,152,670 A * 4/2000 Johnson 705/27 6,199,082 B1 * 3/2001 Fernel et al. 715/522 6,219,680 B1 * 4/2001 Bernardo et al. 707/501.1 6,226,648 B1 * 5/2001 Appleman et al. 707/102 6,247,032 B1 * 6/2001 Bernardo et al. 345/733 6,253,282 B1 * 6/2001 Gish 709/203 6,308,188 B1 * 10/2001 Bernardo et al. 707/501.1
(73) Assignee: Vertical Computer Systems, Inc., Austin, TX (US)	OTHER PUBLICATIONS Lewandowski, Framework for Component-Based Client/Server Computing, Mar. 1998, ACM, pp. 3-27.* * cited by examiner
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.	Primary Examiner—John Chavis (74) Attorney, Agent, or Firm—Brown Raysman Millstein Felder & Steiner LLP
(21) Appl. No.: 09/410,334	ABSTRACT (57) A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.
(22) Filed: Oct. 1, 1999	53 Claims, 2 Drawing Sheets
(51) Int. Cl.: G06F 9/45	
(52) U.S. Cl.: 717/108	
(58) Field of Search: 717/108; 707/103 R; 715/522	
(56) References Cited U.S. PATENT DOCUMENTS 5,544,302 A * 8/1996 Nguyen 345/837 5,555,365 A * 9/1996 Selby et al. 345/765 5,894,554 A * 4/1999 Lowery et al. 707/104.1 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 345/760	



```
graph TD; C[CONTENT 10] --> P[PRODUCT 16]; F[FUNCTIONALITY 14] --> P; FR[FORM 12] --> P
```




Infringement of United States Patent number 6,826,744 B1 by Samsung's Android-Powered Smartphones and Tablet Computers

 US006826744B1	
(12) United States Patent McAuley	(10) Patent No.: US 6,826,744 B1 (45) Date of Patent: Nov. 30, 2004
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	
(75) Inventor: Aubrey McAuley, Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc., Austin, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.	
(21) Appl. No.: 09/410,334 (22) Filed: Oct. 1, 1999	
(51) Int. Cl.: G06F 9/45 (52) U.S. Cl.: 717/108 (58) Field of Search: 717/108; 707/103 R; 715/522	
(56) References Cited U.S. PATENT DOCUMENTS	
5,544,302 A * 8/1996 Nguyen 345/837 5,555,365 A * 9/1996 Selby et al. 345/765 5,894,554 A * 4/1999 Lowery et al. 707/104.1 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 345/760	
OTHER PUBLICATIONS Lewandowski, Framework for Component-Based Client/Server Computing, Mar. 1998, ACM, pp. 3-27. * cited by examiner Primary Examiner—John Chavis (74) Attorney, Agent, or Firm—Brown Raysman Millstein Felder & Steiner LLP	
(57) ABSTRACT A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.	
53 Claims, 2 Drawing Sheets	
	

Samsung's Infringing Android-Powered Smartphones

Acclaim SCH-R880	Galaxy S II/SGH i777
Admire Vitality	Galaxy S II/SGH-T989
Behold II	Galaxy S II Skyrocket
Captivate (Galaxy S)	Gem
Conquer 4G	Gravity Smart
Continuum (Galaxy S)	Infuse 4G
Dart	Intercept
Droid Charge	Moment
Epic 4G (Galaxy S)	Replenish
Epic 4G Touch (Galaxy S II)	Sidekick 4G
Exhibit 4G (Galaxy S)	Stratosphere
Exhibit II 4G	Transfix
Fascinate/Mesmerize (Galaxy S)	Transform Ultra
Galaxy Indulge R910/	Transform
Indulge R915	Vibrant (Galaxy S)
Galaxy Prevail	
Galaxy S 4G	

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers



US006826744B1

(12) **United States Patent**
McAuley

(10) **Patent No.:** **US 6,826,744 B1**
 (45) **Date of Patent:** **Nov. 30, 2004**

(54) **SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK**

(75) **Inventor:** **Aubrey McAuley, Austin, TX (US)**

(73) **Assignee:** **Vertical Computer Systems, Inc., Austin, TX (US)**

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) **Appl. No.:** **09/410,334**

(22) **Filed:** **Oct. 1, 1999**

(51) **Int. Cl.:** **G06F 9/45**

(52) **U.S. Cl.:** **717/108**

(58) **Field of Search:** **717/108; 707/103 R; 715/522**

(56) **References Cited**

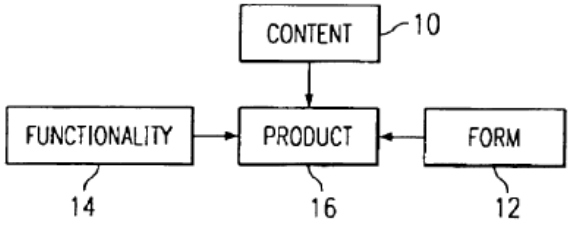
U.S. PATENT DOCUMENTS

5,544,302 A	* 8/1996	Nguyen	345/837
5,555,365 A	* 9/1996	Selby et al.	345/765
5,894,554 A	4/1999	Lowery et al.	707/104.1
5,903,894 A	* 5/1999	Reneris	707/100
5,930,512 A	* 7/1999	Boden et al.	717/102
5,956,736 A	* 9/1999	Hanson et al.	345/760

(57) **ABSTRACT**

A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.

53 Claims, 2 Drawing Sheets



```

graph TD
    C[CONTENT 10] --> P[PRODUCT 16]
    F[FUNCTIONALITY 14] --> P
    FR[FORM 12] --> P
  
```

SAMSUNG'S ANDROID-POWERED SMARTPHONES STANDARD APPLICATIONS INFRINGE


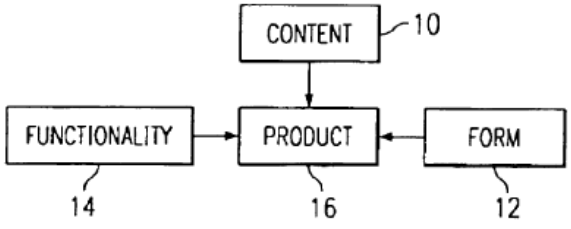
The Samsung Android-Powered Smartphones infringe claims of U.S. Patent number 6,826,744 because Android Framework standard applications are installed into these phones by Samsung during the manufacturing process (many of which are non-removable). The Smartphone is then imported into the United States containing these non-removable applications. These applications are developed using: **"A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application."**

The Samsung Smartphones in Figure 1 are Android Powered Smartphones. As such, they include the following standard Android apps:

- | | |
|------------------------|---------------|
| -Dialer | -Contacts |
| -SMS | -Calendar |
| -Email | -Phones |
| -Gmail | -Camera |
| -Google Maps | -Market |
| -Navigation | -Alarm Clock |
| -Internet Browser | -Calculator |
| -You Tube Video Player | -Music Player |

These standard applications, many of which are non-removable, form the heart of the Smartphone. Without them, the phone will not operate and could not be sold in the U.S.

Infringement of United States Patent number 6,826,744 B1 by Samsung's Android-Powered Smartphones and Tablet Computers

 US006826744B1	
(12) United States Patent McAuley	(10) Patent No.: US 6,826,744 B1 (45) Date of Patent: Nov. 30, 2004
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	
(75) Inventor: Aubrey McAuley, Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc., Austin, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.	
(21) Appl. No.: 09/410,334 (22) Filed: Oct. 1, 1999	
(51) Int. Cl.: G06F 9/45 (52) U.S. Cl.: 717/108 (58) Field of Search: 717/108; 707/103 R; 715/522	
(56) References Cited U.S. PATENT DOCUMENTS	
5,544,302 A * 8/1996 Nguyen 345/837 5,555,365 A * 9/1996 Selby et al. 345/765 5,894,554 A * 4/1999 Lowery et al. 707/104.1 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 345/760	
OTHER PUBLICATIONS Lewandowski, Framework for Component-Based Client/Server Computing, Mar. 1998, ACM, pp. 3-27. * cited by examiner Primary Examiner—John Chavis (74) Attorney, Agent, or Firm—Brown Raysman Millstein Felder & Steiner LLP	
(57) ABSTRACT A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.	
53 Claims, 2 Drawing Sheets	
 <pre> graph TD C[CONTENT 10] --> P[PRODUCT 16] F[FUNCTIONALITY 14] --> P FR[FORM 12] --> P </pre>	

Samsung's Infringing Android-Powered Tablets


Samsung Galaxy Tab10.1 with Verizon 4G LTE

Samsung Galaxy Tab 10.1

Samsung Galaxy Tab 7.0

Samsung Galaxy Tab 8.9

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers



US006826744B1

(12) **United States Patent**
McAuley

(10) **Patent No.:** **US 6,826,744 B1**
(45) **Date of Patent:** **Nov. 30, 2004**

(54) **SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK**

(75) **Inventor:** **Aubrey McAuley, Austin, TX (US)**

(73) **Assignee:** **Vertical Computer Systems, Inc., Austin, TX (US)**

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) **Appl. No.:** **09/410,334**

(22) **Filed:** **Oct. 1, 1999**

(51) **Int. Cl.:** **G06F 9/45**

(52) **U.S. Cl.:** **717/108**

(58) **Field of Search:** **717/108; 707/103 R; 715/522**

(56) **References Cited**

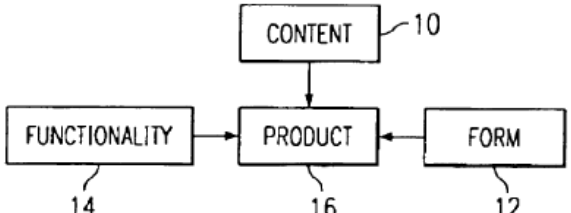
U.S. PATENT DOCUMENTS

5,544,302 A *	8/1996	Nguyen	345/837
5,555,365 A *	9/1996	Selby et al.	345/765
5,894,554 A *	4/1999	Lowery et al.	707/104.1
5,903,894 A *	5/1999	Reneris	707/100
5,930,512 A *	7/1999	Boden et al.	717/102
5,956,736 A *	9/1999	Hanson et al.	345/760

(57) **ABSTRACT**

A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.

53 Claims, 2 Drawing Sheets



```

graph TD
    C[CONTENT 10] --> P[PRODUCT 16]
    F[FUNCTIONALITY 14] --> P
    FORM[FORM 12] --> P
    
```

SAMSUNG'S ANDROID-POWERED TABLET COMPUTERS STANDARD APPLICATIONS INFRINGE


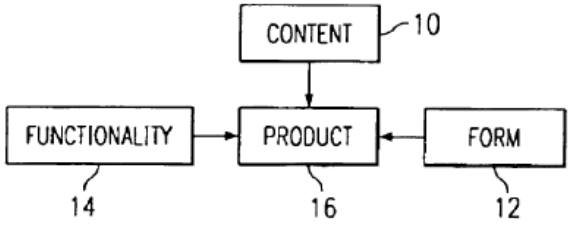
The Samsung Android-Powered Tablet Computers infringe claims of U.S. Patent number 6,826,744 because Android Framework standard applications are installed into these Tablet Computers by Samsung during the manufacturing process (many of which are non-removable). The Tablet Computer is then imported into the United States containing these non-removable applications. These applications are developed using: **"A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application."**

The Samsung Tablet Computers in Figure 2 are Android Powered Tablet Computers. As such, they include the following standard Android apps:

- | | |
|------------------------|---------------|
| -Dialer | -Contacts |
| -SMS | -Calendar |
| -Email | -Phones |
| -Gmail | -Camera |
| -Google Maps | -Market |
| -Navigation | -Alarm Clock |
| -Internet Browser | -Calculator |
| -You Tube Video Player | -Music Player |

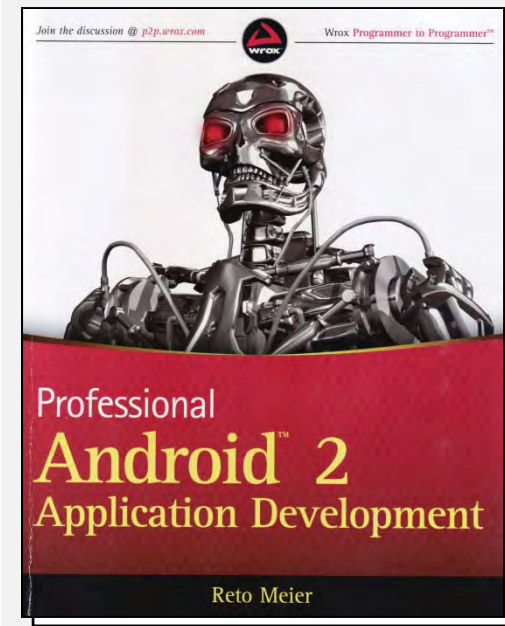
These standard applications, many of which are non-removable, form the heart of the Tablet Computer. Without them, the Tablet Computer will not operate and could not be sold in the U.S.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

 US006826744B1	
(12) United States Patent McAuley	(10) Patent No.: US 6,826,744 B1 (45) Date of Patent: Nov. 30, 2004
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	
(75) Inventor: Aubrey McAuley, Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc., Austin, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.	
(21) Appl. No.: 09/410,334	
(22) Filed: Oct. 1, 1999	
(51) Int. Cl. G06F 9/45	
(52) U.S. Cl. 717/108	
(58) Field of Search 717/108; 707/103 R; 715/522	
(56) References Cited U.S. PATENT DOCUMENTS	
5,544,302 A * 8/1996 Nguyen 345/837 5,555,365 A * 9/1996 Selby et al. 345/765 5,894,554 A * 4/1999 Lowery et al. 707/104.1 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 345/760	
(10) Patent No.: US 6,826,744 B1 (45) Date of Patent: Nov. 30, 2004	
(54) SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK	
(75) Inventor: Aubrey McAuley, Austin, TX (US)	
(73) Assignee: Vertical Computer Systems, Inc., Austin, TX (US)	
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.	
(21) Appl. No.: 09/410,334	
(22) Filed: Oct. 1, 1999	
(51) Int. Cl. G06F 9/45	
(52) U.S. Cl. 717/108	
(58) Field of Search 717/108; 707/103 R; 715/522	
(56) References Cited U.S. PATENT DOCUMENTS	
5,544,302 A * 8/1996 Nguyen 345/837 5,555,365 A * 9/1996 Selby et al. 345/765 5,894,554 A * 4/1999 Lowery et al. 707/104.1 5,903,894 A * 5/1999 Reneris 707/100 5,930,512 A * 7/1999 Boden et al. 717/102 5,956,736 A * 9/1999 Hanson et al. 345/760	
OTHER PUBLICATIONS Lewandowski, Framework for Component-Based Client/Server Computing, Mar. 1998, ACM, pp. 3-27. * cited by examiner Primary Examiner—John Chavis (74) Attorney, Agent, or Firm—Brown Raysman Millstein Felder & Steiner LLP	
(57) ABSTRACT A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.	
53 Claims, 2 Drawing Sheets	
 <pre> graph TD CONTENT[10] --> PRODUCT[16] FUNCTIONALITY[14] --> PRODUCT FORM[12] --> PRODUCT </pre>	

INFRINGEMENT BY THE ANDROID FRAMEWORK

The book "Professional Android 2 Application Development", ISBN 978-0-470-565652-0 written by Reto Meier will be used to explain the infringement. "Reto Meier is a software developer who has been involved in Android since its initial release in 2007. He is an Android Developer Advocate at Google". This book will be referred to henceforth as the "Meier Book".



Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page xvii (more specifically 4th paragraph)** and (more specifically 7th bullet) essentially describes what is known in Page 4 the industry as a framework.

INTRODUCTION

Now is an exciting time for mobile developers. Mobile phones have never been more popular, and powerful smartphones are now a popular choice for consumers. Stylish and versatile phones packing hardware features like GPS, accelerometers, and touch screens, combined with fixed-rate, reasonably priced data plans provide an enticing platform upon which to create innovative mobile applications.

A host of Android handsets are now available to tempt consumers, including phones with QVGA screens and powerful WVGA devices like the Motorola Droid and the Google Nexus One. The real win though, is for developers. With much existing mobile development built on proprietary operating systems that restrict the development and deployment of third-party applications, Android offers an open alternative. Without artificial barriers, Android developers are free to write applications that take full advantage of increasingly powerful mobile hardware and distribute them in an open market.

As a result, developer interest in Android devices has exploded as handset sales have continued to grow. In 2009 and the early parts of 2010 more than 20 Android handsets have been released from OEMs including HTC, Motorola, LG, Samsung, and Sony Ericsson. Android devices are now available in over

Built on an open source framework, and featuring powerful SDK libraries and an open philosophy, Android has opened mobile phone development to thousands of developers who haven't had access to tools for building mobile applications. Experienced mobile developers can now expand into the Android platform, leveraging the unique features to enhance existing products or create innovative new ones.

with no review process, for distributing free and paid apps to an compatible Android devices.

This book is a hands-on guide to building mobile applications using version 2 of the Android software development kit. Chapter by chapter, it takes you through a series of sample projects, each introducing new features and techniques to get the most out of Android. It covers all the basic functionality as well as exploring the advanced features through concise and useful examples.

Google's philosophy is to release early and iterate often. Since Android's first full release in October 2008, there have been seven platform and SDK releases. With such a rapid release cycle, there are likely to be regular changes and improvements to the software and development libraries. While the Android engineering team has worked hard to ensure backwards compatibility, future releases are likely to date some of the information provided in this book.

Nonetheless, the explanations and examples included here will give you the grounding and knowledge needed to write compelling mobile applications using the current SDK, along with the flexibility to quickly adapt to future enhancements.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, Page xvii (more specifically 4th paragraph) and **Page 4 (more specifically 7th bullet)** essentially describes what is known in the industry as a framework.

4 | CHAPTER 1 HELLO, ANDROID

stack produced and supported by the Open Handset Alliance and designed to operate on any handset that meets the requirements. Google has now released its first direct-to-consumer handset, the Nexus 1, but this device remains simply one hardware implementation running on the Android platform.

ANDROID: AN OPEN PLATFORM FOR MOBILE DEVELOPMENT

Google's Andy Rubin describes Android as:

The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation. (<http://googleblog.blogspot.com/2007/11/whereas-my-gphone.html>)

Put simply, Android is a combination of three components:

- A free, open-source operating system for mobile devices
- An open-source development platform for creating mobile applications
- Devices, particularly mobile phones, that run the Android operating system and the applications created for it

More specifically, Android is made up of several necessary and dependent parts, including the following:

- A hardware reference design that describes the capabilities required for a mobile device to support the software stack.

A run time used to execute and host Android applications, including the Dalvik virtual machine and the core libraries that provide Android-specific functionality. The run time is designed to be small and efficient for use on mobile devices.

- An application framework that agnostically exposes system services to the application layer, including the window manager and location manager, content providers, telephony, and sensors.
- A user interface framework used to host and launch applications.
- Preinstalled applications shipped as part of the stack.
- A software development kit used to create applications, including tools, plug-ins, and documentation.

What really makes Android compelling is its open philosophy, which ensures that you can fix any deficiencies in user interface or native application design by writing an extension or replacement. Android

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page 28** explains how the UI (Form) "decouples" the application logic (function) from your UI design (form) on its first paragraph.

28 | CHAPTER 2 GETTING STARTED

Listing 2-2 shows the UI layout defined in the main.xml file created by the Android project template.

LISTING 2-2: Hello World layout resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
```

Defining your UI in XML and inflating it is the preferred way of implementing your user interfaces, as it neatly decouples your application logic from your UI design.

To get access to your UI elements in code, you add identifier attributes to them in the XML definition. You can then use the `findViewById` method to return a reference to each named item. The following XML snippet shows an ID attribute added to the Text View widget in the Hello World template:

```
<TextView
    android:id="@+id/myTextView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World, HelloWorld"
/>
```

And the following snippet shows how to get access to it in code:

```
TextView myTextView = (TextView) findViewById(R.id.myTextView);
```

Alternatively (although it's not generally considered good practice), you can create your layout directly in code, as shown in Listing 2-3.

LISTING 2-3: Creating layouts in code

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    LinearLayout.LayoutParams lp;
    lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.FILL_PARENT);

    LinearLayout.LayoutParams textViewLP;
    textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);

    TextView myTextView = new TextView(this);
    myTextView.setText("Hello World, HelloWorld");
```

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

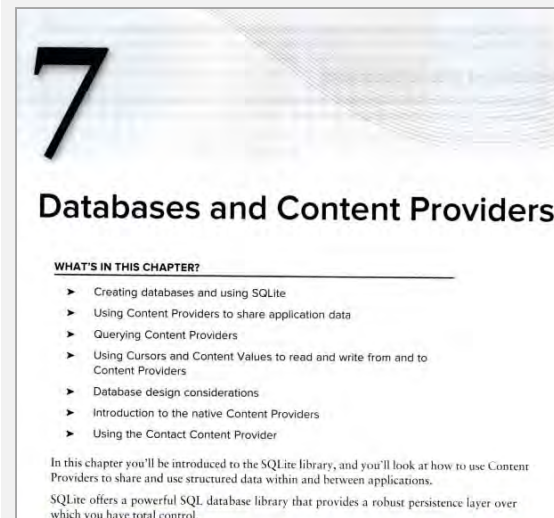
creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Android is a framework of arbitrary objects which separates form, function and content of a computer application, which runs on a host, either a smart phone, a tablet PC or even a virtual machine on a PC.

Meier Book, **Page 209** defines how the "Content Providers" "decouples" data storage (content) from your application logic (function) on its third paragraph.



Content Providers offer a generic interface to any data source by decoupling the data storage layer from the application layer.

INTRODUCING ANDROID DATABASES

Structured data persistence in Android is provided through the following mechanisms:

- **SQLite Databases** When managed, structured data is the best approach, Android offers the SQLite relational database library. Every application can create its own databases over which it has complete control.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

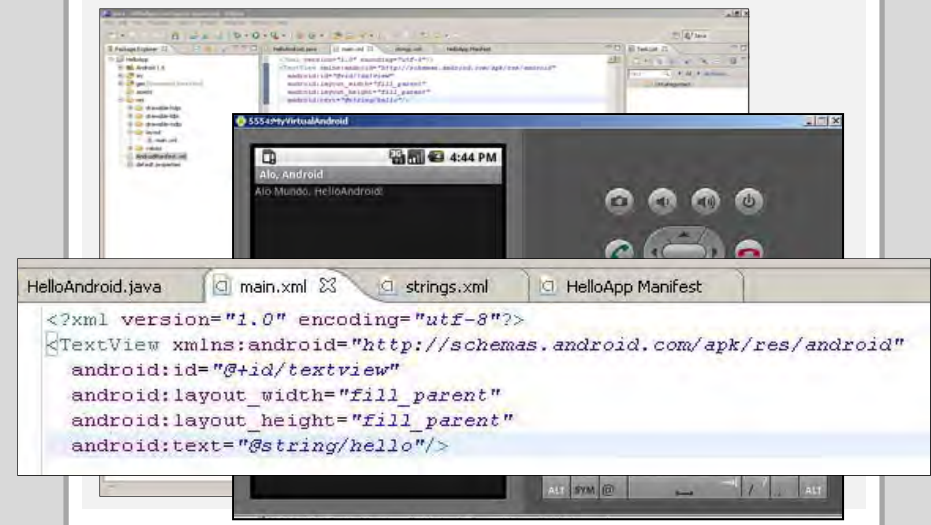
and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Android's objects are arbitrary objects. We created a sample application using the Android Software Developers' Kit (Android SDK). In the following example:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

The Arbitrary Object "textview" is being called by its name (android:id="@+id/textview"), and additionally has three parameters.

When compiling and running the application, the compilation completes, and the application runs fine (it prints "Alo mundo,Hello Android!") on the Android device as seen in here:



Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

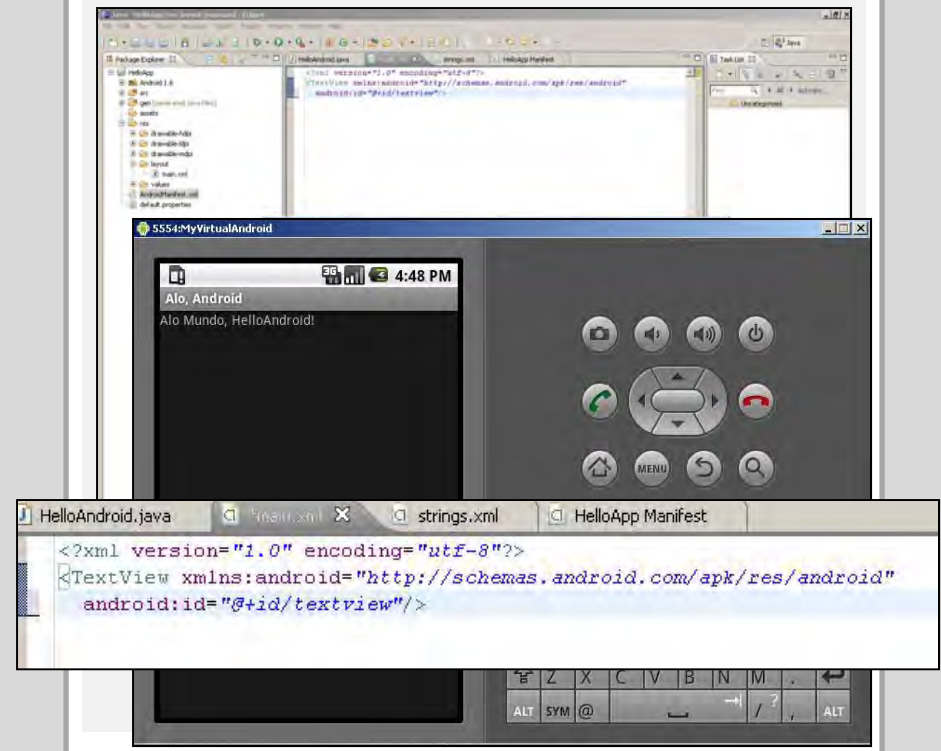
managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

However, when we modify the call to "textview" to include just the name of the Arbitrary Object, the call looks this way:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
```

When making the above modification, the program compiles and the application runs normally, as can be seen on this screen print:



Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

The creation of Arbitrary objects can be accomplished using command line tools, such as a text editor, or using the recommended tool, Eclipse. Using a text editor is more work because all of the files needed to create the object need to be edited correctly and placed on the correct location (folders) in order for the compiler to work. Using Eclipse facilitates the process.

The steps for generating a Computer application on the Android Framework when not using Eclipse are:

- Creation of the application by manually creating each source file with a text editor
- Compiling of the application using a Java Compiler
- Creating a "package" (A file which's extension is .APK", using the Android SDK named apkbuilder.bat , which is distributed with the Android SDK.
- Moving that package file to the Android device, either by using Google's "Android Market" application (in the case of a SmartPhone), or using the builtin android application named "File Manager", clicking on the file, and Android will then install it on the device.

When using eclipse, all is necessary is to click on the "Run" menu – that is what was done on the application described earlier.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

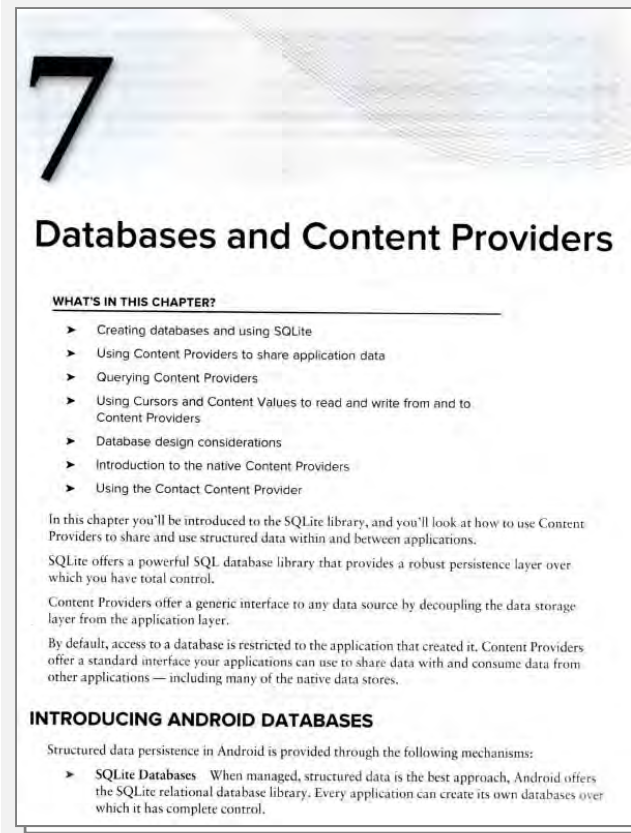
A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types **for generating said content of said computer application**, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

The content can be generated by the “Content Provider” feature of Android. Meier Book, **Pages 209-211** is a description of how it works. The title of the chapter, and throughout the whole text, it clarifies that the Android Framework can be used for generating the content:



Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

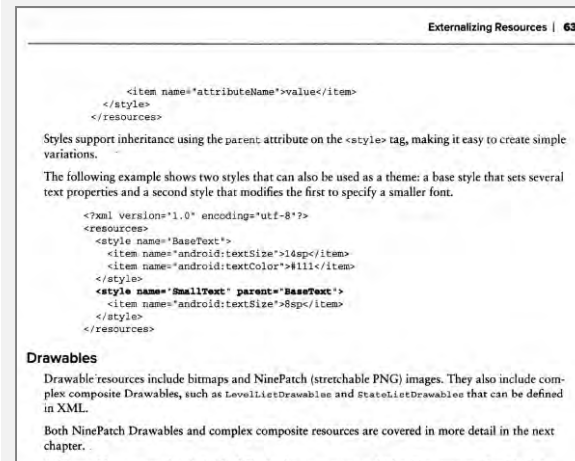
A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, **said form of said computer application**, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Form in the Android Framework normally is defined on the Resource folder, referred to as "R" or "Layouts", having the default "layout.xml" as the Form, but this name can be changed. On Meier Book, **Page 63** and 64 describes how they work. More specifically, the last three paragraphs of **Page 63** define Layouts as your "Presentation Layer" and "User Interface", which is Form.



Layout resources let you decouple your presentation layer by designing user interface layouts in XML rather than constructing them in code.

The most common use of a layout is for defining the user interface for an Activity. Once defined in XML, the layout is "inflated" within an Activity using setContentView, usually within the onCreate method. You can also reference layouts from within other layout resources, such as layouts for each row in a List View. More detailed information on using and creating layouts in Activities can be found in Chapter 4.

Using layouts to create your screens is best-practice UI design in Android. The decoupling of the layout from the code lets you create optimized layouts for different hardware configurations, such as varying screen sizes, orientation, or the presence of keyboards and touchscreens.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, **said form of said computer application**, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Also on Meier Book, **Page 50**, it describes "Activities" as Presentation Layer of Android Applications, which can be Form.

50 CHAPTER 3 CREATING APPLICATIONS AND ACTIVITIES

Mobile devices come in a large variety of shapes and sizes and are used across the world. In this chapter you'll learn how to externalize resources to ensure your applications run seamlessly on different hardware (particularly different screen resolutions and pixel densities), in different countries, and supporting multiple languages.

Next you'll examine the Application class, and learn how to extend it to provide a place for storing application state values.

Arguably the most important of the Android building blocks, the Activity class forms the basis for all your user interface screens. You'll learn how to create new Activities and gain an understanding of their life cycles and how they affect the application lifetime.

Finally, you'll be introduced to some of the Activity subclasses that simplify resource management for some common user interface components such as maps and lists.

WHAT MAKES AN ANDROID APPLICATION?

Android applications consist of loosely coupled components, bound by an application manifest that describes each component and how they all interact, as well as the application metadata including its

Activities Your application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an Activity is equivalent to a Form. You'll learn more about Activities later in this chapter.

tions. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 9.

- **Content Providers** Shareable data stores. Content Providers are used to manage and share application databases. They're the preferred means of sharing data across application boundaries. This means that you can configure your own Content Providers to permit access from other applications and use Content Providers exposed by others to access their stored data. Android devices include several native Content Providers that expose useful databases like the media store and contact details. You'll learn how to create and use Content Providers in Chapter 7.
- **Intents** An inter-application message-passing framework. Using Intents you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- **Broadcast Receivers** Intent broadcast consumers. If you create and register a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

Meier Book, **Page 50**, second bullet, describes “Services” as “The invisible workers of the application”, (Which is Functionality).

90 | CHAPTER 3: CREATING APPLICATIONS AND ACTIVITIES

Mobile devices come in a large variety of shapes and sizes and are used across the world. In this chapter, you'll learn how to externalize resources to ensure your applications run seamlessly on different hardware (particularly different screen resolutions and pixel densities), in different countries, and supporting multiple languages.

Next you'll examine the Application class, and learn how to extend it to provide a place for storing application state values.

Arguably the most important of the Android building blocks, the Activity class forms the basis for all your user interface screens. You'll learn how to create new Activities and gain an understanding of their life cycles and how they affect the application lifetime.

Finally, you'll be introduced to some of the Activity subclasses that simplify resource management for some common user interface components such as maps and lists.

WHAT MAKES AN ANDROID APPLICATION?

Android applications consist of loosely coupled components, bound by an application manifest that describes each component and how they all interact, as well as the application metadata including its hardware and platform requirements.

The following six components provide the building blocks for your applications:

- ▶ **Activities** Your application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an Activity

Services The invisible workers of your application. Service components run in the background, updating your data sources and visible Activities and triggering Notifications. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 9.

Android devices include several native Content Providers that expose useful databases like the media store and contact details. You'll learn how to create and use Content Providers in Chapter 7.

- ▶ **Intents** An inter-application message-passing framework. Using Intents you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- ▶ **Broadcast Receivers** Intent broadcast consumers. If you create and register a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

The object library, when using eclipse is described as the "Workspace" of the Android Applications. If not using Eclipse, the management of these objects can be done manually by using normal command-line commands , such as "Copy", "Rename" , "Delete" or "Move", inside of a folder structure which is mandated by the Android SDK so the Computer Application can be built properly.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 1.

A method for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application, said method comprising:

creating arbitrary objects with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application;

managing said arbitrary objects in an object library;

and deploying said arbitrary objects from said object library into a design framework to create said computer application.

These Arbitrary objects are deployed by building the complete application as described earlier, then,

-Moving that package file to the Android device, either by using Google's "Android Market" application (in the case of a SmartPhone), or using the built-in android application named "File Manager", clicking on the file, and Android will then install it on the device.

The Meier Book, **page XVII**, on its 5th Paragraph, describes that the Android Market can be used for distribution, or Deployment:

INTRODUCTION

Now is an exciting time for mobile developers. Mobile phones have never been more popular, and powerful smartphones are now a popular choice for consumers. Stylish and versatile phones packing hardware features like GPS, accelerometers, and touch screens, combined with fixed-rate, reasonably priced data plans provide an enticing platform upon which to create innovative mobile applications.

A host of Android handsets are now available to tempt consumers, including phones with QVGA screens and powerful WVGA devices like the Motorola Droid and the Google Nexus One. The real win though, is for developers. With much existing mobile development built on proprietary operating systems that restrict the development and deployment of third-party applications, Android offers an open alternative. Without artificial barriers, Android developers are free to write applications that take full advantage of increasingly powerful mobile hardware and distribute them in an open market.

As a result, developer interest in Android devices has exploded as handset sales have continued to grow. In 2009 and the early parts of 2010 more than 20 Android handsets have been released from OEMs including HTC, Motorola, LG, Samsung, and Sony Ericsson. Android devices are now available in over 26 countries on more than 32 carriers. In the United States, Android devices are available on all four major carriers: T-Mobile, Verizon, AT&T, and Sprint. Additionally, you can now buy the unlocked Google Nexus One handset directly from Google at <http://www.google.com/phones>.

Built on an open source framework, and featuring powerful SDK libraries and an open philosophy, Android has opened mobile phone development to thousands of developers who haven't had access

Using the Android Market for distribution, developers can take advantage of an open marketplace, with no review process, for distributing free and paid apps to all compatible Android devices.

This book is a hands-on guide to building mobile applications using version 2 of the Android software development kit. Chapter by chapter, it takes you through a series of sample projects, each introducing new features and techniques to get the most out of Android. It covers all the basic functionality as well as exploring the advanced features through concise and useful examples.

Google's philosophy is to release early and iterate often. Since Android's first full release in October 2008, there have been seven platform and SDK releases. With such a rapid release cycle, there are likely to be regular changes and improvements to the software and development libraries. While the Android engineering team has worked hard to ensure backwards compatibility, future releases are likely to date some of the information provided in this book.

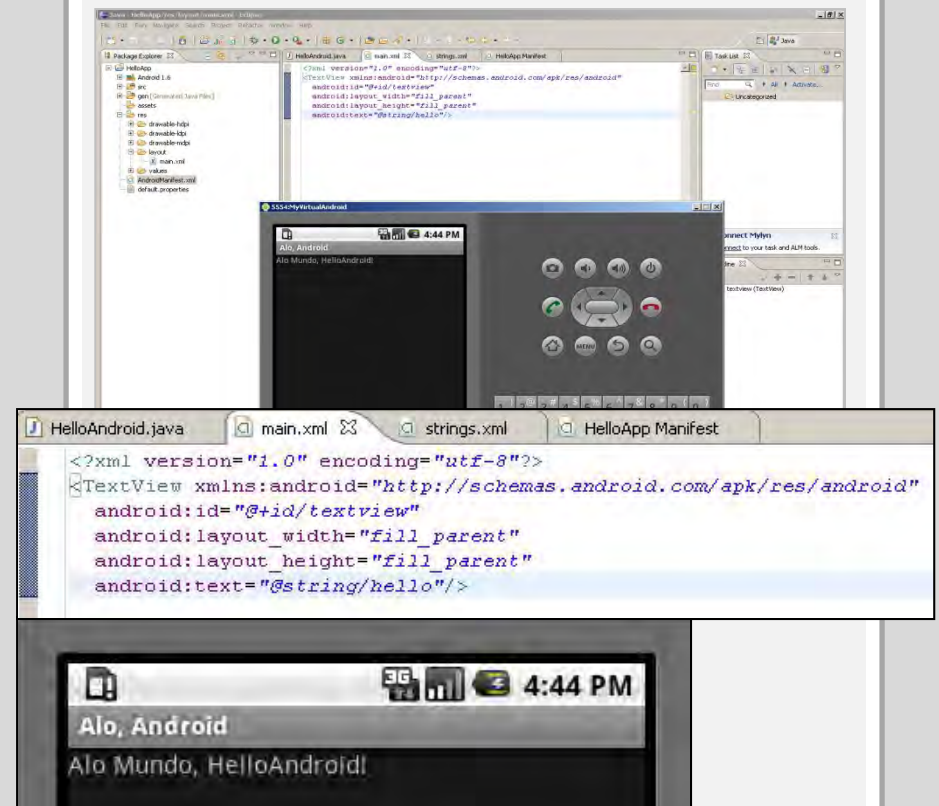
Nonetheless, the explanations and examples included here will give you the grounding and knowledge needed to write compelling mobile applications using the current SDK, along with the flexibility to quickly adapt to future enhancements.

Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 3.

The method of claim 1, wherein said various object types comprise text file pointers.

The Computer Application demonstrated on Claim 1, Element 1, makes use of text file pointers. Bellow Picture, shows the text file pointer "Alo Mundo, HelloAndroid!".



Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 4.

The method of claim 1, wherein said various object types comprise binary file pointers.

The Computer Application demonstrated on Claim 1, Element 1, makes use of binary file pointers. In the Meier Book, **Page 63** (on the paragraph labeled "Drawables") mentions images and bitmaps, which are binary file pointers:

Externalizing Resources | 63

```
<item name="@drawable/icon" value="@drawable/icon" />
</item>
</resources>
</resources>
```

Styles support inheritance using the parent attribute on the <style> tag, making it easy to create simple variations.

The following example shows two styles that can also be used as a theme: a base style that sets several text properties and a second style that modifies the first to specify a smaller font.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="BaseText">
    <item name="android:textSize">14sp</item>
    <item name="android:textColor">#111</item>
  </style>
  <style name="SmallText" parent="BaseText">
    <item name="android:textSize">8sp</item>
  </style>
</resources>
```

Drawables

Drawable resources include bitmaps and NinePatch (stretchable PNG) images. They also include complex composite Drawables, such as `LevelListDrawables` and `StateListDrawables` that can be defined in XML.

Both NinePatch Drawables and complex composite resources are covered in more detail in the next chapter.

All Drawables are stored as individual files in the `res/drawable` folder. The resource identifier for a Drawable resource is the lowercase file name without an extension.

Layouts

Layout resources let you decouple your presentation layer by designing user interface layouts in XML rather than constructing them in code.

The most common use of a layout is for defining the user interface for an Activity. Once defined in XML, the layout is "inflated" within an Activity using `setContentView`, usually within the `onCreate` method. You can also reference layouts from within other layout resources, such as layouts for each row in a List View. More detailed information on using and creating layouts in Activities can be found in Chapter 4.

Using layouts to create your screens is best-practice UI design in Android. The decoupling of the layout from the code lets you create optimized layouts for different hardware configurations, such as varying screen sizes, orientation, or the presence of keyboards and touchscreens.

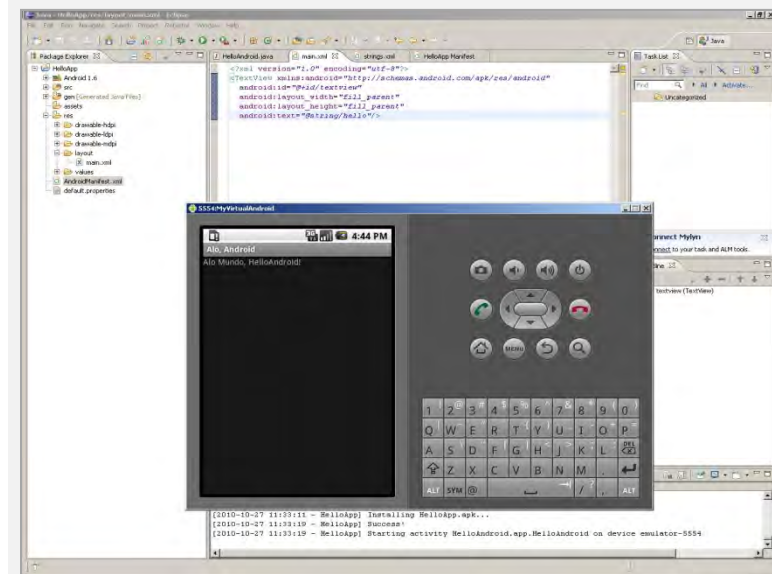
Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 5.

The method of claim 1, wherein said various object types comprise compiled executables.

The Computer Application demonstrated on Claim 1, Element 1, compiles into an executable.

Below Picture, shows in the bottom the program compiling into an executable.



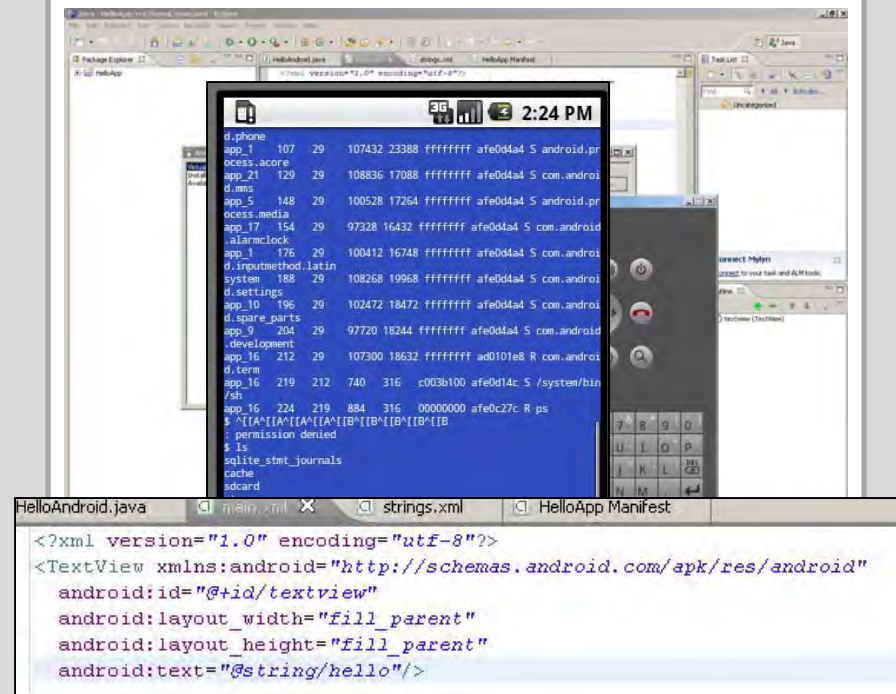
Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 9.

The method of claim 1, wherein said various object types comprise cached executables.

The Computer Application demonstrated on Claim 1, Element 1, compiles into an executable, which can be cached by the Android Framework. Like any operating system, every object that the framework hosts will be cached.

In this picture showing the Android terminal emulation application, the response to the command "ls" shows a subdirectory named "cache":



```
d.phone
app_1 107 29 107432 23388 ffffffff afe0d4a4 S android.pr
access.acore
app_21 129 29 108836 17088 ffffffff afe0d4a4 S com.androi
d.sms
app_5 148 29 100528 17264 ffffffff afe0d4a4 S android.pr
access.media
app_17 154 29 97328 16432 ffffffff afe0d4a4 S com.androi
.alarmclock
app_1 176 29 100412 16748 ffffffff afe0d4a4 S com.androi
d.inputmethod.latin
system 188 29 108268 19968 ffffffff afe0d4a4 S com.androi
d.settings
app_10 196 29 102472 18472 ffffffff afe0d4a4 S com.androi
d.spare_parts
app_9 204 29 97720 18244 ffffffff afe0d4a4 S com.androi
.development
app_16 212 29 107300 18632 ffffffff ad010e8 R com.androi
d.tor
app_16 219 212 740 316 c003b100 afe0d14c S /system/bin
/sh
app_16 224 219 884 316 00000000 afe0c27c R ps
$ ls
permission denied
$ ls
sqlite_stmt_journals
cache
sdcard
```

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

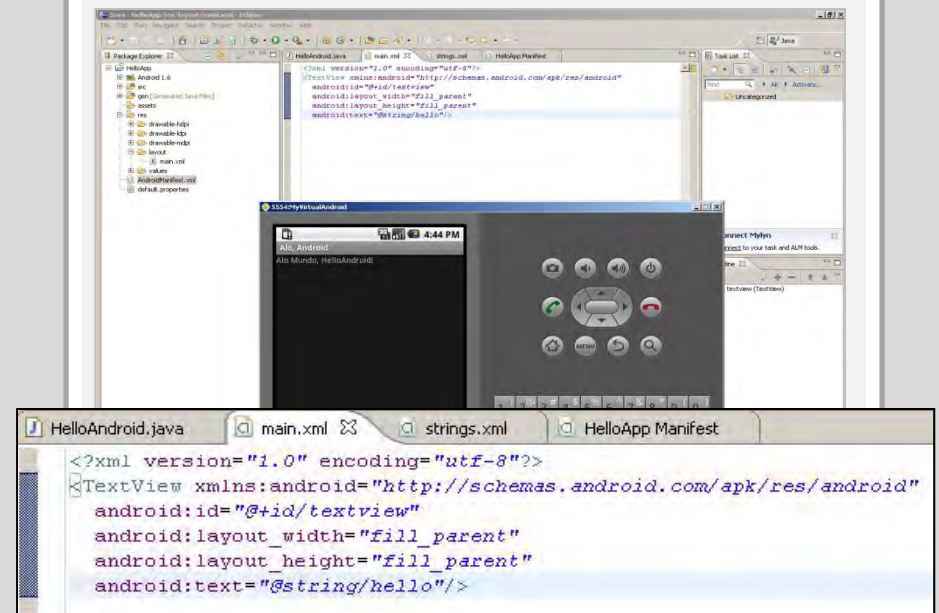
Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 17.

The method of claim 1, further comprising deploying arbitrary objects locally.

The Computer Application demonstrated on Claim 1, Element 1, makes use of arbitrary objects locally.

Below Picture, shows in the arbitrary object named textview being accessed locally within the mobile device.

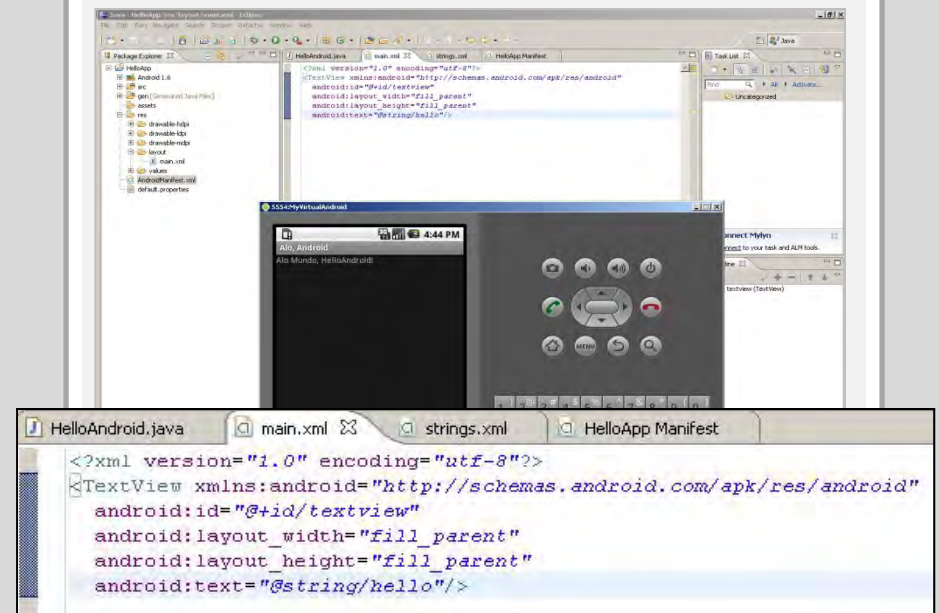


Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 21.

The method of claim 1, wherein said arbitrary objects can be accessed and deployed into said design framework using said corresponding arbitrary names.

The Android Framework enables access and deployment of said arbitrary names into said design framework. Bellow picture shows the deployment of said arbitrary objects (the arbitrary object named "textview") accessed and deployed into said design framework with the said arbitrary names (textview):

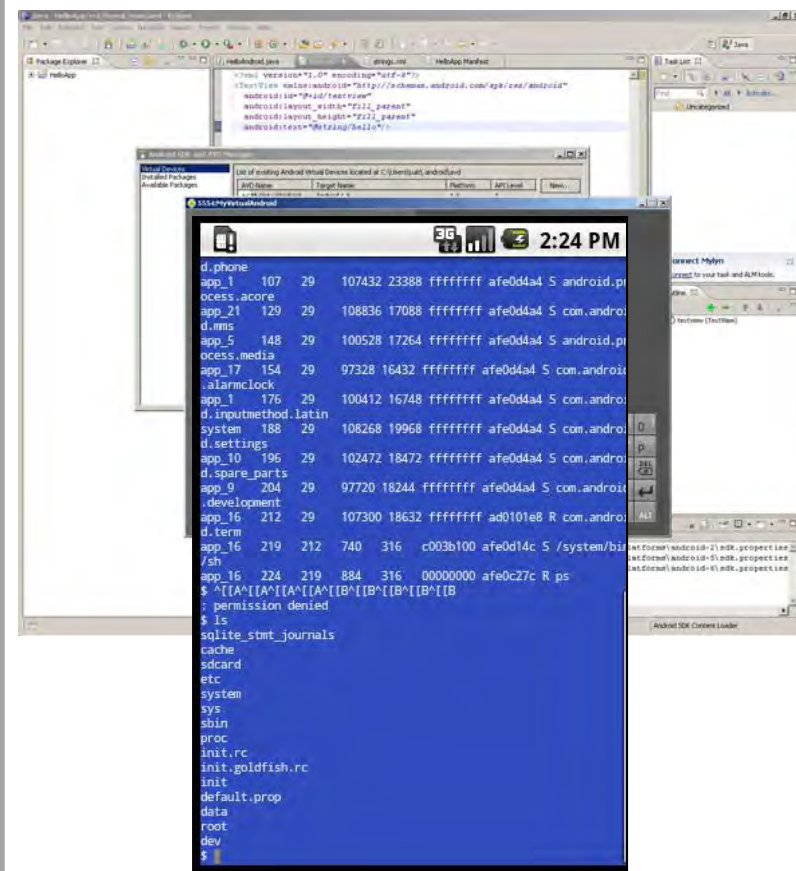


Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 23.

The method of claim 1, further comprising caching objects.

The Android Framework caches objects. In this picture showing the Android terminal emulation application, the response to the command "ls" shows a subdirectory named "cache":



```
d:phone
app_1 107 29 107432 23388 ffffffff afe0d4a4 S android.p
ocess.acore
app_21 129 29 108836 17088 ffffffff afe0d4a4 S com.androi
d.rms
app_5 148 29 100528 17264 ffffffff afe0d4a4 S android.p
ocess.media
app_17 154 29 97328 16432 ffffffff afe0d4a4 S com.androi
.alarmclock
app_1 176 29 100412 16748 ffffffff afe0d4a4 S com.androi
d.inputmethod.latin
d.settings
app_10 196 29 102472 18472 ffffffff afe0d4a4 S com.androi
d.spare_parts
app_9 204 29 97720 18244 ffffffff afe0d4a4 S com.androi
.development
app_16 212 29 107300 18632 ffffffff ad0101e8 R com.androi
d.term
app_16 219 212 740 316 c003b100 afe0d14c S /system/bi
/sh
app_16 224 219 884 316 00000000 afe0c27c R ps
$ ^[[A^[[A^[[A^[[A^[[A^[[B^[[B^[[B^[[B^[[B^[[B
; permission denied
$ ls
sqlite_stmt_journals
cache
sdcard
etc
system
sys
sbin
proc
init.rc
init.goldfish.rc
init
default.prop
data
root
dev
$
```


Infringement of United States Patent number 6,826,744 by Samsung's Android-Powered Smartphones and Tablet Computers

Claim 25.

The method of claim 1, further comprising generating arbitrary objects in a programming language that is compatible or supported by said host system.

The Android Framework allows such arbitrary objects to be developed in Java, which is a programming language supported by the host system. This can be shown in “Meyer Book”, **page 28** (below), the program named “LISTING 2-3”, written in Java:

28 | CHAPTER 2 GETTING STARTED

Listing 2-2 shows the UI layout defined in the main.xml file created by the Android project template.



LISTING 2-2: Hello World layout resource

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Hello World, HelloWorld"
    />
</LinearLayout>
```

Defining your UI in XML and inflating it is the preferred way of implementing your user interfaces, as it neatly decouples your application logic from your UI design.

To get access to your UI elements in code, you add identifier attributes to them in the XML definition. You can then use the `findViewById` method to return a reference to each named item. The following XML snippet shows an ID attribute added to the Text View widget in the Hello World template:

LISTING 2-3: Creating layouts in code

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    LinearLayout.LayoutParams lp;
    lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
                                      LinearLayout.LayoutParams.FILL_PARENT);

    LinearLayout.LayoutParams textViewLP;
    textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
                                              LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);

    TextView myTextView = new TextView(this);
    myTextView.setText("Hello World, HelloWorld");
```