Functional-based modeling
White paper
June 2009

Rational. software

UML for the C programming language.

This document requires Adobe Reader version 6.0.1 or later. To view this document, please download Adobe Reader from: http://www.adobe.com/

Contents

Executive summary

Unified Modeling Language (UML) has been highly successful in the modeling of software-intensive systems, including systems-oriented models and realtime
or these models
and, this is
ge for realtime
e other hand,
velopment, and
d.

ltime and

Object-
object-b
functio
in how c
power a

resent the system
nd the only tricky
res of UML into C
mation simple:
s to "member"
structs
nerated by
er functions; and

Object-based designs are even simpler because they don't use the full richness of UML; specifically, they disallow the use of generalization and polymorphism. This results in designs that more directly and obviously map to the C language features but don't take advantage of all the conceptual tools available within UML.

Highlights

The last approach, functional-based modeling, is the focus of this paper. There are many reasons developers may want to avail themselves of the power of UML to represent the different aspects of their systems (e.g., functional, behavioral, atures completely,

hout requiring its

UML ca
develop
feature

concepts of files,
l attributes.
critical designs are
Considerations in

The Fun
subset
of funct
system

or extends UML
onalC profile uses
ased systems. The
able 1.

			e system with relations
			cts constructed from the executables and libraries
			eir sequences among
	File diagram	Class diagram	Shows the .cpp and .h files and their relations, including
	Source code diagram	<none>	Shows the generated source code as editable text
Behavior	Message diagram	Sequence diagram	Shows sequences of calls and events sent among a set of files, including passed parameter values
	State diagram	State diagram	Shows the state machine for files and how their included functions and actions are executed as events (whether synchronous or asynchronous) are received
	Flowchart	Activity diagram	Details the flow of control for a function or use case

Table 1: FunctionalC profile diagrams.

This document requires Adobe Reader version 6.0.1 or later. To view this document, please download Adobe Reader from: http://www.adobe.com/

Highlights

Functional development within UML

Developers can program functionally with UML diagrams by using a UML stereotype called a file, which is simply a graphical representation of a source

developers are used

file is added to

n much the same

O) programming.

A file is

as a gra

source f

familiar

variable

e structural

ofile represents .c

er as one element

upling in your

iagram. This

gn, but can just

vel of abstraction,

d functions in the

ram and generate,

phics. In addition,

on the host PC by

t the behavior and

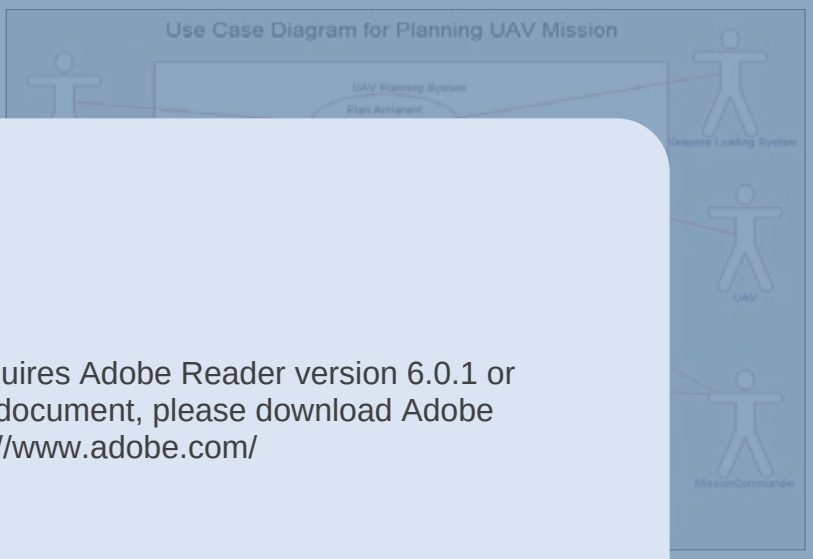
Let's take the example of a timer file. This file could have the responsibility to keep track of time. It could have variables such as minutes and seconds, and perhaps functions such as reset and tick. The reset function could initialize the ime by one second. n such as the ables minutes and eset().

ypical C program

, except it was .

ons contained in her a state n be used in the same model, and files can be converted to objects, if desired. This enables developers who wish to migrate to an OO approach to do it at their own pace and doesn't force an all-or-nothing switch.

Highlights

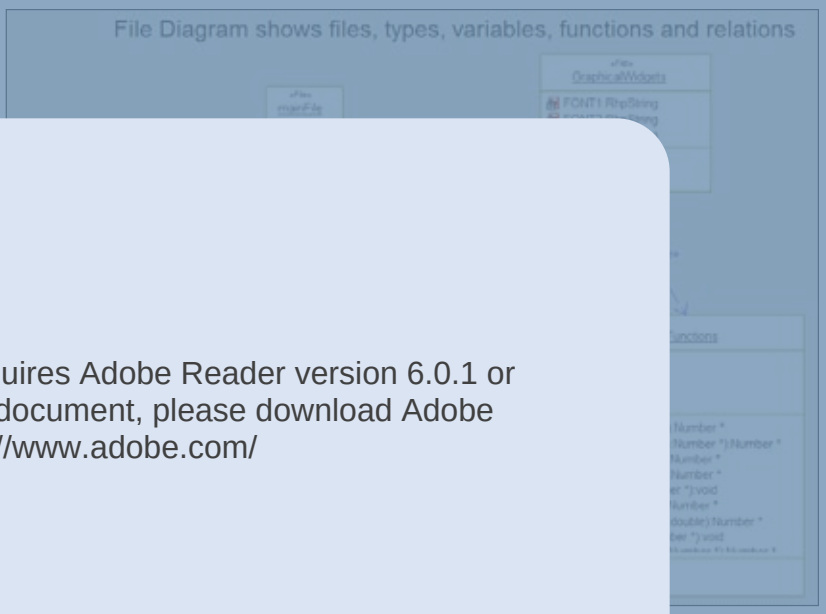


Use cas ments (called use
way to o This is the same
system' ge a system's

functions,
ng source or

Highlights

A file dia
a file's c
to other
build dia
relation



This document requires Adobe Reader version 6.0.1 or later. To view this document, please download Adobe Reader from: http://www.adobe.com/

e compilation and

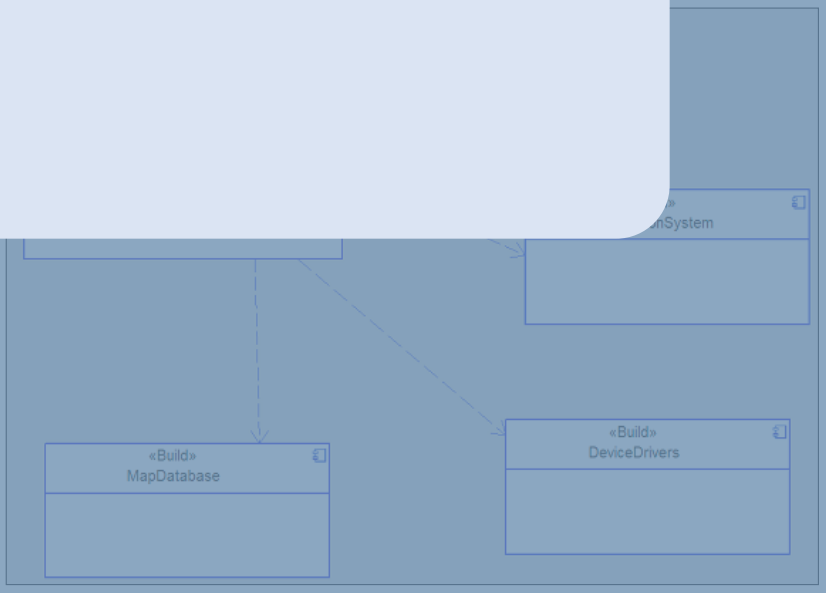
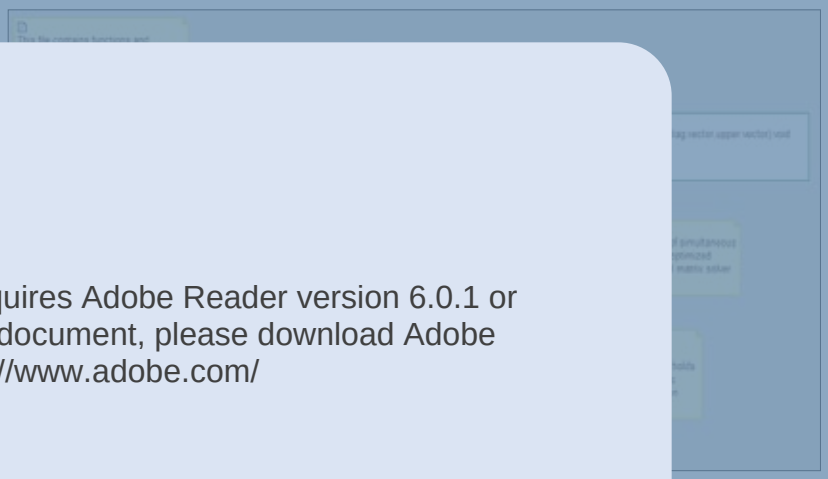


Figure 4: Build diagram.

Highlights

A call gra
relation
and vari
diagram
messag

The call graph shows the relations among functions and variables.



specific set of
in a particular
or per

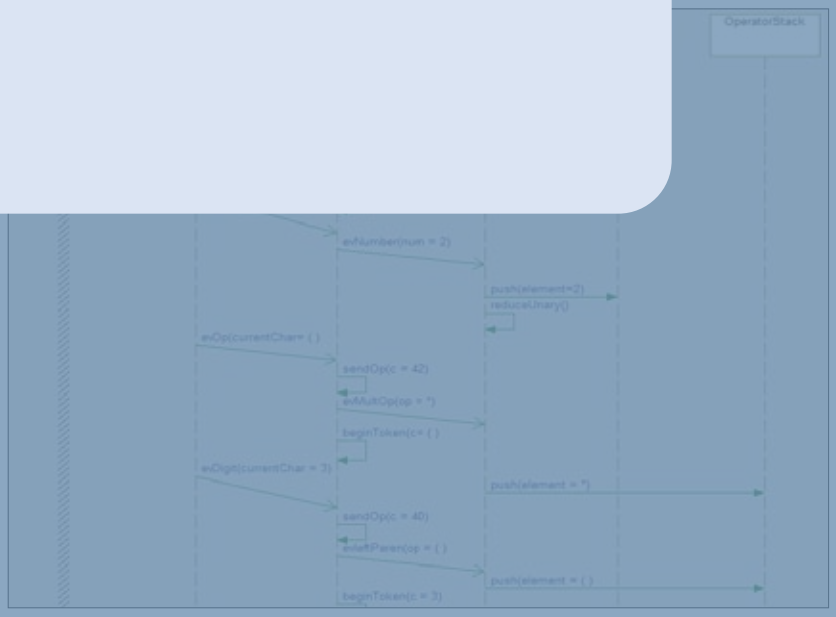


Figure 6: Message diagram.

Highlights

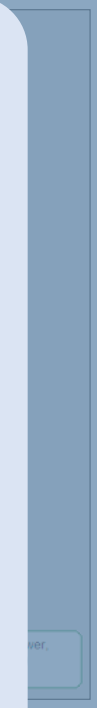
A state diagram represents the set of sets achievable by a file (or use case), the events it receives and the transitions that occur as a result of those event receptions.

A state d
sets ach
case the
transac

Highlights

A flowchart is used primarily to represent an algorithm or the detailed functional flow of control within a function.

Flowcha
the func



Highlights

The sam… describ… validate…

With rea… model p… amount… generat…

Validating the design using the model

In a typical design, it is never clear that the model is correct until it has been executed. Technology is available today to perform graphical back-animation …m the model with …en uses this …el concepts. This …l can be used to …e value of the …ile is in, trace the … step through a …roject and allows …and doing design …okkeeping …th the model and …and correct them

… from the model; …nerated …nd embedded …ode can be …rammer writes, …e generated …tions, variables, …eeds to specify the functions and actions on the state charts.

Ensuring that the model and the code are always synchronized

In the traditional use of computer-aided software/system engineering (CASE) tools, programmers spend time creating a model and then generate the co
ever has the ti
models get mo
less useful. Ag
to the code to
that the code
during the ma
to a new versi

Conclusion

The introduct
variables into
of model-driv
to which they
now possible t
without chang
code (IP), eith
quality struct
further by all
existing proce
adopting thes
immensely. M
UML-based M
development cycle substantially in some cases.

For more information

To learn more about UML for the C programming language, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/rational

IBM Corporation
Software Group
Route 100

erica

are trademarks or reg-
l Business Machines
other countries, or
emarked terms are
n this information with
ese symbols indicate
ademarks owned by
as published. Such
d or common law
current list of IBM
eb at "Copyright and

ce names may be
thers.

IBM products and
tends to make them
IBM operates.

documentation is
es only. While efforts
ness and accuracy
s documentation, it is
of any kind, express or
on is based on IBM's
y, which are subject
IBM shall not be
sing out of the use of,
or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

RAW14058-USEN-01