



ArgoUML Snabbguide

Komma igång med ArgoUML 0.34

**Kunle Odutola
Anthony Oguntimehin
Linus Tolke
Michiel van der Wulp**

ArgoUML Snabbguide : Komma igång med ArgoUML 0.34

av Kunle Odutola, Anthony Oguntimehin, Linus Tolke, och Michiel van der Wulp

Copyright © 2001 Kunle Odutola

Copyright © 2004-2010 Michiel van der Wulp

Copyright © 2008 Översättare: Linus Tolke

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Innehållsförteckning

1. Förord	v
1. Introduktion	1
1.1. Välkommen till ArgoUML	1
1.2.	1
2.	2
2.1.	2
2.2.	2
2.3.	2
2.4.	2
2.5.	2
2.5.1.	2
2.5.2.	2
3.	3
3.1.	3
3.2.	3
3.3.	3

Förord

1:e april 2001

I förra årtusendets sista decennium, gick objektorienterad teknologi (OO) från att vara ett lustigt fenomen i 1960-talets laboratoriemiljö till att bli den huvudsakliga mjukvaruutvecklingsparadigmen. Det har varit en lång och svår resa - och det var nära att det slutade i dunkel - framförallt för att OO-teknologi kräver en stor förändring i tankeprocessen hos de som designar systemet, utvecklarna, och andra inblandade i mjukvaruutvecklingens livscykel.

Det var inte förrän OO-blandspråken såsom C++, Klassisk Ada och Objekt Pascal, började komma på 80-talet som det stora flertalet fick kännedom och och möjlighet att experimentera med OO-teknologing. Dessa språk kombinerade stöd för OO-programmering med stöd för traditionell procedurell programmering. Avsaknaden av stöd för procedurell programmering i rena OO-språk hade i många år hindrat anammandet av OO-teknologin. Resan, som hade börjat med införandet av objekt i Simula på 1960-talet, var nu slutförd. OO-teknologin var tillgänglig för det stora flertalet, i alla fall OO-programmering.

Simulering är den grundläggande förutsättningen för att bygga upp OO-teknologin. Ett OO-system designas och byggs upp i allt väsentligt som en simulering av den riktiga världen med användandet av mjukvaruartefakter. Denna förutsättning är lika kraftfull som den är enkel. Genom att designa och bygga mjukvarusystem på detta sätt, kan man använda samma språk och idéer vid analysen, vid designen och vid implementationen av ett OO-system. Detta tillåter ett system att bli designat och testat, eller korrektare simulerat, utan att man behöver bygga systemet först. Denna möjlighet, parad med möjligheten att designa system på en mycket hög nivå gav erfarna OO-användare styrkan att designa och framgångsrikt implementera komplexare system än vad som tidigare hade varit möjligt.

Tillgängligheten av flera olika OO-metoder och det slutgiltiga anammandet av Unified Modelling Language (UML) som det standardiserade språket för att kommunicera OO-koncept fortsatte att införa OO-teknologin som norm. Framgången för det objekt-baserade språket Visual Basic och den samtidiga utvecklingen och blyt-snabba anammandet av det objekt-orienterade programspråket Java var katalysatorn som behövdes för de sista stegen av anammandet av OO-teknologin som norm.

ArgoUML skapades som ett verktyg för och en verktygsomgivning för att använda vid analys och design av objekt-orienterade mjukvarusystem. I detta avseende liknar det flera av de kommersiella programvaruutvecklingsverktygen som säljs som verktyg för att modellera mjukvarusystem. ArgoUML har några viktiga skillnader från många av dessa verktyg:

- ArgoUML har flera funktioner som stödjer de kognitiva behoven hos objektorienterade mjukvaruutvecklare och arkitekter.
- ArgoUML arbetar uteslutande med öppna standarder - UML, XMI, SVG, OCL med flera. I detta avseende är ArgoUML fortfarande (fem år efteråt) långt före många kommersiella verktyg.
- ArgoUML är till 100% utvecklat i Java. Detta gör att ArgoUML kör på alla plattformar där det finns en tillförlitlig portering av Java2.
- ArgoUML är Open Source. Tillgängligheten till källkoden gör att nya mjukvaruutvecklare och forskare har ett fungerande ramverk som de kan använda för vidare utveckling av CASE-verktygstekniker.

Genom skapandet av utvecklingsverktyget och omgivningen ArgoUML har Jason Robbins och resten av hans utvecklingsgrupp på UCL utnyttjat den särställning som UML har som objekt-orienterat modelleringsspråk och Java såsom den mest framgångsrika objekt-orienterade utvecklingsplattformen för att skapa ett stabilt verktyg för objekt-orienterad systemutveckling och att skapa förutsättningarna för

utveckling av OO CASE-verktygt och forskning.

Kunle Odutola

Kapitel 1. Introduktion

Kunle Odutola

1.1. Välkommen till ArgoUML

ArgoUML är en kraftfull men lättanvänd interaktiv grafisk mjukvarudesignomgivning som stödjer design, utveckling och dokumentation av objektorienterade mjukvaruapplikationer.

Om du är bekant med mjukvaruapplikationer som brukar kallas CASE-verktyg (verktyg för Datorstött mjukvaruutveckling - Computer Aided Software Engineering) kommer du genast att finna dig tillrätta i ArgoUML.

java -jar argouml.jar -help

java -jar argouml.jar -locale en
Ewan R. Grantham
Allan Todd
