Schlussbewertung - Galapagos

Thomas Grottker, Kai Nessig

February 8, 2012

Contents

1 Technik

1.1 Ada

Ada wurde zur Implementierung des Frameworks genauso wie zur Implementierung der Verteilungsfunktion im LunarlanderAda Beispiel.

1.2 Ahven

Testframework für Ada, adaptiert von JUnit

1.3 Box2D/JBox2D

Mächtiges Physik-Framework, verwendet für die evaluate()-Methode des Lunarlanders und für die Darstellung im LLMonitor

1.4 Scala

Scala wurde zur Implementierung des Frameworks und zur Implementierung der Darstellung und Berechnung des LunarlanderScala Beispiels.

1.5 Specs2

Testframework für Scala.

1.6 XML/Ada

Bibliothek, die zwecks Parsing im LunarlanderAda Beispiel verwendet wird.

2 Werkzeuge

2.1 IntelliJ

Verwendet für die Entwicklung der Scala Implementierung.

2.2 GNAT Programming Studio (GPS)

Verwendet für die Entwicklung der Ada Implementierung.

2.3 SVN

Versionskontrolle, verwendet mit TortoiseSVN unter Windows und dem nativen Client unter Linux.

2.4 Visual Paradigm for UML

Verwendet zur Erstellung der UML-Klassendiagramme.

3 Fazit

Die Unterschiedlichen Konzepte der Anwendung von generischen Parametern in Ada (Paketorientiert) und Scala (Klassenorientiert) verhalten sich fast schon konträr zu einander. Beides hat Vor- und Nachteile, die klassenorientierte Modellierung in UML neigt dazu, viele implizite Abhngigkeiten zu verstecken und führt damit den Vorteil der Modellierungsphase ein wenig ad absurdum. Die momentane Implementierung in Ada ist dadurch z. B. in eine Instanziierungsschlacht ausgeartet, deren Einhalt man nur durch noch tiefere Verschachtelung der Pakete zur Folge hätte.