Schlussfolgerung und Ausblick

Die angestrebten Ziele wurden erreicht. Einerseits liefert die erzeugte Klassenbibliothek autokorrelierte Zufallszahlen, wobei der Grad der Autokorrelation übe die Korrelationskoeffizienten gesteuert werden kann, andererseits ist ein Simio-AddIn realisiert worden, welches es ermöglicht, die Klassenbibilothek in einer Simulationssoftware zu nutzen.

Mathematische Grundlagen/Auswertung

Durch unsere Auswertung ist ersichtlich geworden, dass Arta.Standard Zufallszahlen erzeugt, welche der JAVA-Implementation sehr ähnlich sind. Als einziger Unterschied können die ACFS und PACFS gesehen werden, welche sich in unserer Implementation langsamer Null annähern. Die generierten Zahlen liegen jeweils in den von der Verteilung definierten Grenzen und wiesen das angestrebte autokorrelierte Muster auf.

Simulation

Arta.Standard wurde im Rahmen dieser Arbeit bereits innerhalb eines Simulationsmodells getestet. Als nächsten Schritt sehen wir die Verwendung in komplexeren Modellen. In den Testsimulationen wurde klar ersichtlich, dass unsere Bibliothek autokorrelierte Zufallszahlen liefert und sich dies von den systemgenerierten Zahlen stark unterscheiden. Als nächstes Experiment sehen wir die Verwendung bzw. Gegenüberstellung eines komplexen Systems, welches einen realen Anwendungsfall abbildet. So können die Unterschiede in Bezug auf reale Daten klar ersichtlich gemacht werden.

Erweiterungspotential

Von der Implementationsseite her sehen wir ebenfalls Erweiterungspotential. Einerseits kann ArtaStatistics um eine grosse Funktionalität erweitert werden, um so einen umfassenderen, tieferen Einblick in den Arta-Prozess zu geben. Eine von uns wichtig erachtete Funktionalität sehen wir in der graphischen Darstellung bzw. der Datenausgabe. Durch eine Erweiterung könnten die Arta-Zahlen entweder direkt visualisiert oder formatiert ausgegeben werden. Weiter kann auch das Simio-Plugin in seiner Form weiterentwickelt werden. Der Ansatz liegt hierbei darin, dass die Arta-Zahlen nicht durch eine Function in ein System gespiesen werden, sondern dass dies über Simio-Events geschieht.

Abstaract

Die realisierte Klassenbibliothek ist im stande autokorrellierte Zufallszaheln zu erzeugen. Dabei ist der Grad der Autokorrrelation über die Korrealtionskoeffizienten steuerbar.

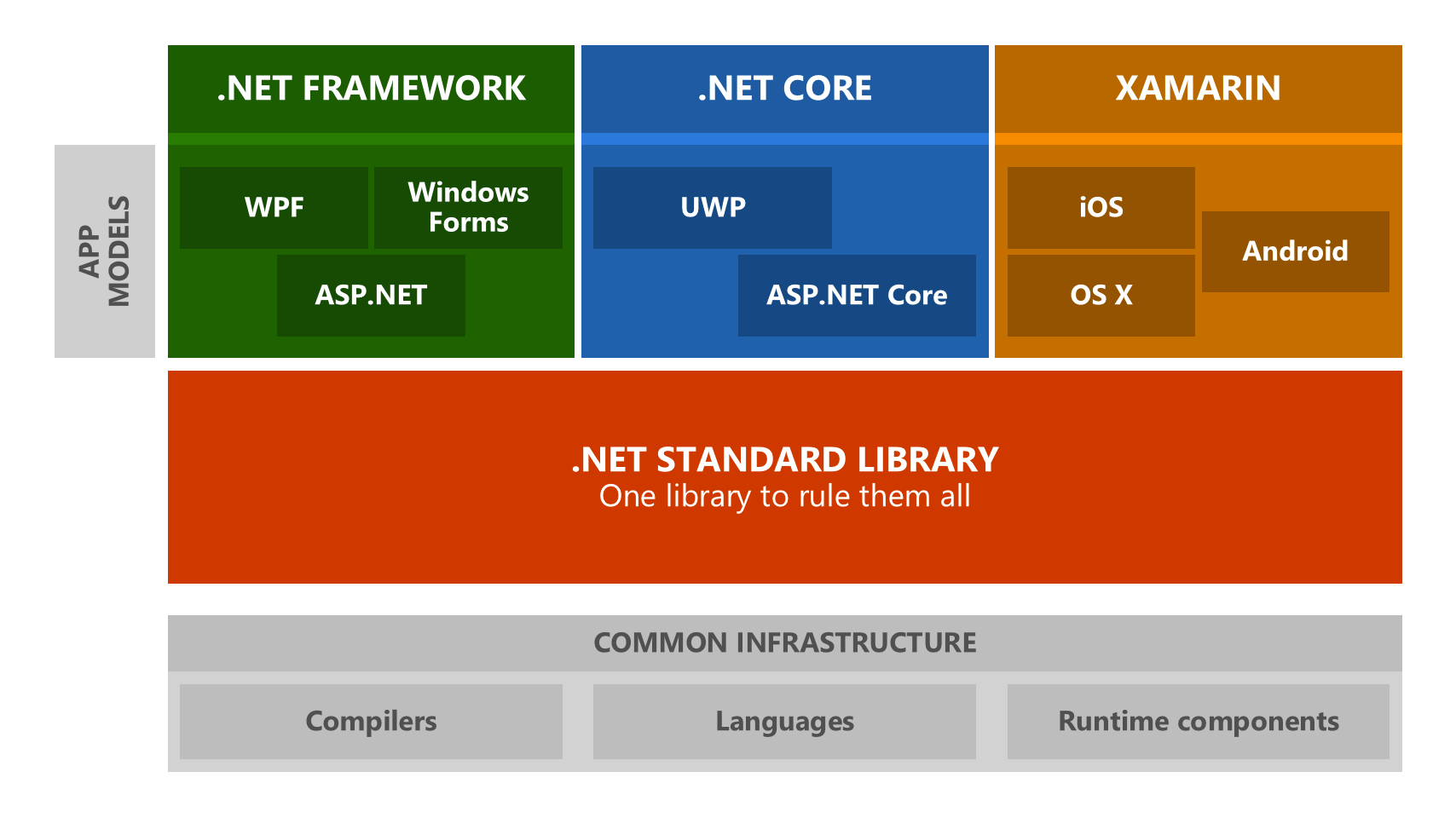
Die Auswertungen zeigen klar auf, dass die Zufallszahlen von Arta.Standard im gewünschten Bereich liegen. Als Referenz wurden Zufallszahlen mithilfe der Java-Implementation erzeugt und diese anschliessend mit den unseren Verglichen.

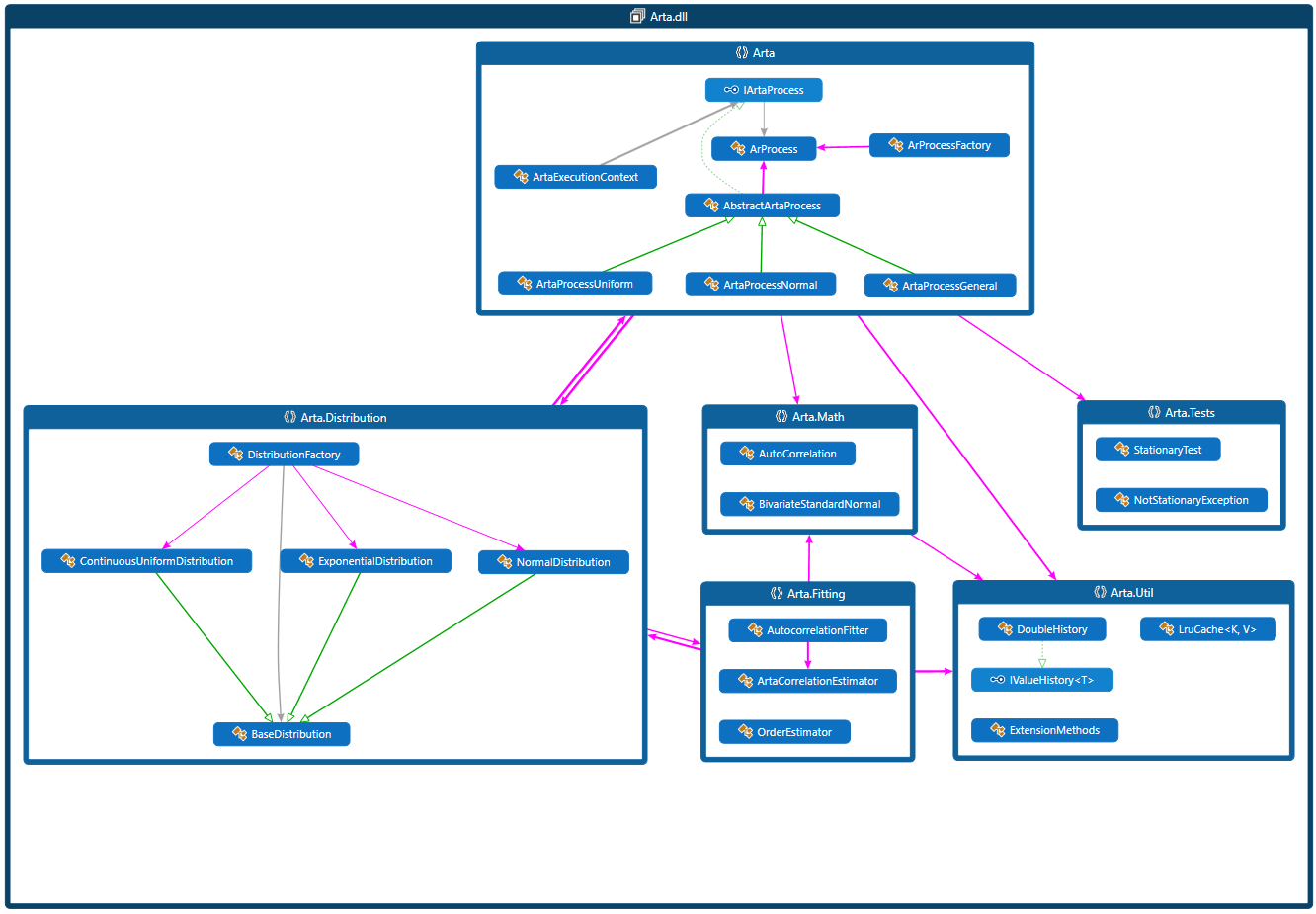
In Form von einfachen Simulationmodellen wurde einerseits die Funktionalität des Simio-AddIns getestet, andererseits wurde ein erster Vergleich zwischen simio-generierten Zahlen und Arta-Zahlen getätigt. Klar ersichtlich sind auch hier die Unterschiede zwischen den beiden Erzuegungsarten.

Implementation

Grundlegendes

ARTA.Standard basiert auf einer .Net Standard 1.6 Klassenbibliothek. Damit setzen wir auf dem untersten Layer auf, was Abhängigkeiten reduziert und die Bibliothek unabhängiger macht.





Arta

Dieser Namespace umfasst den Kern der Klassenbibliothek. Direkt darin sind die Klassen zur Erzeugung des AR- und ARTA-Prozesses angesiedelt. Erzeugt wird ein ARTA-Prozess durch einen ArtaExecutionProzess. Diesem werden die Korrelationskoeffizienten und die gewünschte Verteilung, per Enum-Typ, übergeben. Durch das Setzen des entsprechnden Vertielungstyp erzuegt die DistributionFactory einen entsprechenden ARTA-Prozess und gibt diesen zurück. Zuletzt wird durch die ArProcessFactory einen neuen AR-Prozess instanziert. Anschliessend werden die entsprechenden Objekte zurückgegeben und im ArtaExecutionContext gekapselt.



Arta.Distribution

Dieser Namespace beherbergt alle Verteilungsklassen. Einerseits sind werden hier Klassen für die eigentliche Verteilung abgebildet, andererseit erzeugt die DistributionFactory je nach gewählter Verteilung einen entsprechenden ARTA-Prozess.

Die Klasse BaseDistribution stellt die abstrakte Klasse für alle Verteilungen dar. Die konkreten Implementierungen der einzelnen Verteilungen findet in den gleichnamigen Klassen statt. Sie kapseln jeweils Objekte einer Verteilung und geben eine konkrete Verteilung aus Mathnet.Numerics zurück. Durch diese Kapselung und Generalisierung durch das Basisinterface können wir in der Implementation der gesamten Klassenbibliothek einen höheren Abstraktionslevel aufrecht erhalten. Weiter ist die Erweiterung um weitere Verteilungen einfach realisierbar.

Arta.Math und Arta.Fitting

Innerhalb dieses Namespaces werden die mathematischen Operationen zur Erzeugung des AR- und ARTA-Prozesses durchgeführt. Durch die Klasse AutoCorrealation wird eine Korrelationsmatrix auf der Basis der übergebenen Korrelationskoeffizienten generiert. Diese wird wiedrrum durch einen AutocorrelationFitter genutzt, um die Stuktur des darunterleigenden AR-Prozesses auf die des ARTA-Prozesses anzupassen.