



Software-Engineering-Werkzeuge zeichnen sich heute noch immer durch einen geringen Integrationsgrad und eine eingeschränkte Visualisierung aus. Verschiedene Werkzeuge sind in der Regel notwendig, um alle Phasen des Entwicklungsprozesses zu unterstützen. Verschiedene Notationen sind nötig, um alle Systemaspekte zu modellieren. Auch erfasst die Visualisierung in der Regel nicht die algorithmische Ebene. Der FracTool®-Modellierer besitzt diese Einschränkungen nicht. Mit seinen Leistungsmerkmalen ermöglicht er einen neuartigen Zugang zur Prozessmodellierung und Programmierung.



Äquivalent zu C++ und Java

FracTool® ist universell einsetzbar und unterstützt den imperativen Programmierstil. Er ist damit den Programmiersprachen C++ und Java in Leistung und Ausdrucksfähigkeit äquivalent. Die Codegenerierung erfolgt in die Zielsprachen C++, Java oder auch in die Hardware-Beschreibungssprache VHDL. Bestehende C++- und Java-Bibliotheken lassen sich leicht integrieren.

Vollständige Visualisierung

Die Visualisierung erlaubt die integrierte Darstellung von Operationen, Operanden, Flusskontrollstrukturen und Parallelität sowie von Zugriffen auf komplexe Operanden und Traversen darauf. Unter Verwendung derselben fraktalen Geometrie lässt sich die Visualisierung der Algorithmen in die Architekturebene fortsetzen. Damit ist es möglich, ein gesamtes Softwaresystem in einer einzigen integrierten und kompakten Graphik darzustellen.

Pragmatische Programmierung

Die Visualisierung trennt scharf die anwendungsbezogenen Aspekte eines Programms von den plattformbezogenen Aspekten. Sie verzichtet dabei gezielt auf semantische Details wie Parameterübergabeverfahren etc., die auf die Ebene der automatischen Codeerzeugung und der Konfiguration verlagert werden. Dies bedeutet für die Qualifikation des FracTool®-Programmierers, dass sie nahezu frei von computerspezifischem Wissen sein kann.

Durchgängiger Entwicklungsprozess

Die einzelnen Phasen des Entwicklungsprozesses entsprechen im Wesentlichen verschiedenen Vollständigkeitsgraden des zu entwickelnden Systems. Phasenübergänge erfolgen innerhalb derselben Notation, was einen fließenden Übergang von der Spezifikation zur Implementierung ermöglicht.

Prof. Dr. Johannes Reichardt
j.reichardt@fbi.fh-darmstadt.de
| Fachhochschule Darmstadt
Haardtring 100
D-64295 Darmstadt
Telefon +49(0)6151/16-8467
Telefax +49(0)6151/16-8935

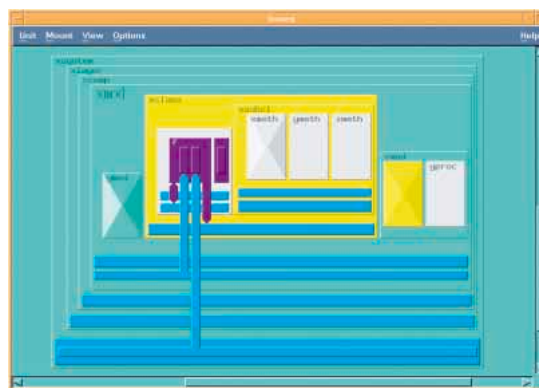
| FracTool GmbH
Telefon +49(0)6151/3919514
Mobil +49(0)173/3204236
| www.fractool.com

Datenbank-gestützte Integration

Die Objektdatenbank des Modellierers speichert den visuellen Programmcode, die Programm-Dokumentation sowie die automatisch generierte Dokumentation des Entwicklungsprozesses und ist vielfältigen Recherchen zugänglich. Die FracTool®-Datenbank stellt damit eine konsistente Grundlage für den gesamten Entwicklungsprozess dar.

Fraktale Programmstruktur

Die Visualisierung ist einheitlich und durchgängig vom Argument eines Methodenaufrufs bis zur Gesamtarchitektur. Fraktale Views zeigen außer dem fokussierten Modul die gesamte für die Funktion und das Verständnis relevante Umgebung des Moduls. Der dabei auftretende logarithmische Informationsreduktionseffekt erlaubt ein hohes Maß an Übersichtlichkeit.



Optimierbare Chipfläche

Die Bildfläche eines FracTool®-Programms ist algorithmisch auf ein Minimum reduzierbar. Die dadurch erhöhte Kompaktheit des Programms verbessert auch seine Verständlichkeit. Diese für textuelle Programmiersprachen undenkbare Möglichkeit der Optimierung soll die ingenieurmässige Software-Entwicklung unterstützen.

FracTool® – A Fractal Modeler

Software engineering tools today are still characterised by a low degree of integration and a restricted visualisation. In practice, different tools are necessary to support all of the steps of the development process. Different notations are necessary to model the different aspects of the software system. As a rule, visualisations on the algorithmic level cannot support the imperative programming style common in practice or are restricted to special application domains. These restrictions do not apply to the FracTool® Modeler. Its features listed below enable a truly innovative approach to process modeling and programming.

Equivalence to C++ and Java: FracTool® is a general-purpose modeler and supports the imperative programming style. Target languages for code generation are e.g. C++, Java, JVM Bytecode or the hardware description language VHDL. Existing C++ and Java libraries as well as GUI interfaces can be easily integrated.

Complete Visualisation: FracTool® visualisation integrates operations, operands, traverses and access on complex operands as well as flow-control and parallelism. Using the same fractal geometry the visualisation continues up to the architectural level. This enables integration and high-resolution display of an entire software system in form of a single compact graph.

Pragmatic Programming: The visualisation used concentrates on the application-oriented aspects of the program and hides platform-oriented aspects as e.g. details of parameter passing. Semantic aspects are widely transferred to the level of automatic code generation and configuration. This implies that the

qualification of the FracTool® programmer is no longer dominated by computer-specific knowledge.

Continuous Development Process: The different phases of the development process essentially correspond to different stages of completeness of the system under development so that the process amounts to a step-wise completion without leaving notation. This enables a smooth transition from specification to implementation. Particularly, the early phases of the development process are well supported.

Database Integration: The object database of the FracTool® Modeler stores the visual program code as well as the program documentation and the automatically generated documentation of the development process, thereby offering a consistent basis for the entire development process.

Fractal Program Structure: Fractal views permit focussing on a module while showing all of its visual environment essential for its working and understanding. The logarithmic information reduction effect of a fractal view together with the features of abstraction and coarsening enables a clear representation of a visual program even on standard-sized screens.

Optimised Chip Area: The area of a FracTool® program can be reduced algorithmically to a minimum. Minimal area implies increased compactness of the program and at the same time increased clearness. The possibility of optimising FracTool® programs – inconceivable for textual programs – supports a true engineering approach to software development.