INGENIEURBÜRO
GUTTENBERG & HÖRDEGEN

Amalienstraße 49a
80799 München

FON: +49 (89) 21 55 04 26 - 0
FAX: +49 (89) 21 55 04 26 - 9

kontakt@energiefluss.info
www.energiefluss.info

A Portrait of the Engineer as a Haskell Programmer

Dr.-Ing. Philipp Guttenberg

- Vortrag 30min, deutsch, Praxisbericht
- Zielgruppe sind Ingenieure und Haskell-Programmierer. Keine speziellen Vorkenntnisse sind nötig, weder in Haskell noch im Ingenieursbereich

Thema

Haskell ist bekannt für die reine Umsetzung des funktionalen Ansatzes. Oft wird aber bezweifelt, dass die mathematischen Ausrichtung der Sprache auch für den Produktiveinsatz geeignet ist.

Wir geben einen Einblick in unsere Eindrücke und Erfahrungen aus einem zweijährigen anspruchsvollen Entwicklungsprojekt im Ingenieursbereich.

Wir ermutigen Ingenieure sich Haskell als Produktivsprache anzuschauen. Die Haskell-Community fordern wir auf, Quereinsteigern unter die Arme zu greifen.

Energieflussanalyse

Wir haben eine Methodik zur Wirkungsgradanalyse in komplexen Netzwerken mit Energiefluss entwickelt. Die Anforderungen der Energieflussanalyse an die Implementierung sind hoch. Die Spezifika werden kurz erläutert.

Arbeitserfahrung eines Quereinsteigers

Wer das Programmieren in erster Linie mit Matlab gelernt hat, tut sich mit Haskell zunächst schwer. Dies trifft wohl für viele Ingenieure zu. Im Speziellen:

- Tupel versus algebraische Datentypen: Letztere sind ausserhalb der funktionelen Programmierung wenig bekannt. Zudem erinnert die Klammerung von Tupeln sehr an die gewohnte Schreibweise beim Aufruf von Funktionen.

- In unserer Anwendung arbeiten wir viel mit Zeitsignalen. Eine typsichere Handhabung ist wünschenswert. Eine korrekte physikalische und logische Typisierung wird jedoch schnell anspruchsvoll. Leider gibt es bisher keine didaktische Einführung in die (fortgeschrittenere) Typprogrammierung und Compilererweiterungen.
- Bei der Strukturierung von Code ist ein Umstellung vom omnipräsenten objektorientierten Paradigma hin zum datenflussorientierten Denken nötig.

Hat man die Hürden gemeistert, winkt eine rosige Zukunft:

- Haskell ist sehr gut als Spezifikationssprache geeignet, um eine erste Implementierung zu bekommen. Features können dann zu gegebener Zeit effizienter und eleganter reimplementiert werden.
- Der Code ist in allen Phasen der Entwicklung stabil. Bei Refaktorierungen hilft der Typchecker, Änderungen konsequent durchzuziehen, so dass man zu jedem Zeitpunkt ein funktionierendes Programm hat.
- Nebeneffektfreiheit macht Reimplementierung und Refaktorierung ebenfalls einfach.
- Logische Inkonsistenzen lassen sich nicht implementieren. Wenn man stecken bleibt, ist man gezwungen erneut über die Sachverhalte nachzudenken. Die Qualität des Endprodukts steigt.
- Laziness erleichtert die Arbeit mit großen Datenmengen.
- Ein Zertifizierung des Codes sollte mit einer typstarken Sprache einfacher sein. Leider ist uns bis dato keine Norm bekannt, bei der die Typstärke der eingesetzten Sprache als Nachweis für die Korrektheit einer Implementierung angeführt werden kann.

Zu einigen Punkten möchten wir Code-Beispiele vorstellen. Zu einer Demonstration der Software wären wir bereit, jedoch würde dies den Rahmen einer halben Stunde sprengen.

Neben diesen Punkten wollen wir unsere Erfahrung auch zu einigen profanen Aspekten äussern: Compilerstabilität, verfügbare Dokumentation und Bibliotheken, Verfügbarkeit von Arbeitskräften, Community.

Vision

Als Ingenieurbüro können wir uns zwei wirtschaftliche Modell für Haskell vorstellen:

- Das Ingenieurbüro selbst. Es profitiert von Haskell durch die oben genannten Punkte. Vom Ingenieur kann man jedoch im Allgemeinen nicht erwarten, sich vertieft mit Haskell's Typsystem auszukennen.
- Haskell-Consultancies: Bei manchen spezifischen Problemen vor allem mit dem Typchecker kann es deshalb sinnvoll sein, externe Hilfe in Anspruch zu nehmen. Hier könnten sich selbstständige Informatiker etablieren.

Beide Seiten sollten unserer Meinung nach ein wirtschaftliches Ökosystem bilden und stärker kooperieren. Hierbei ist vor allem die Bereitschaft der Ingenieure gefragt, Haskell anzunehmen.

Um Haskell im Ingenieurswesen als feste Größe zu etablieren, sollte die Verwendung typstarker Sprachen bei der Implementierung als Möglichkeit zum Korrektheitsnachweis bei der Zertifizierung in die Normen aufgenommen werden. Hier sind vor allem die Informatiker gefragt, Impulse zu setzen, da Ingenieure von sich aus nicht auf diese Idee kommen werden.