Modellgetriebene SW-Entwicklung

* iceCream.use bearbeiten
  + OCL-Constraints definieren
  + [Durchschnittswerte für aktuelle Werte und Varianz]
* nach Generierung:
  + Constraints mit „Leben“ füllen
  + andere View für Varianzabweichung
  + jeweils Beurteilung, wie schwierig das ist: z.B. wie schwierig ist der generierte Code zu verstehen, wird das MVVM-Paradigma entsprechend berücksichtigt?

roter Faden für den Artikel:

1. Einleitung:
   1. Hintergrund: Projektarbeit im Fach Modellgetriebene SW-Entwicklung im Masterstudiengang Informatik
   2. Entwicklung einer mobilen MVVM Applikation mittels des Tools JUSE4Android
   3. Die Anwendung ermöglicht eine Überwachung der Ice-Cream Partikel in der Luft und orientiert sich stark an dem von Martin Fowler verwandten Beispiel zur Demonstration von GUI Architekturen.
   4. Die generierte Anwendung wird funktional erweitert und es wird eine Anpassung einer View-Komponente vorgenommen.
   5. Durch dieses Vorgehen sollen die Stärken und Schwächen des Tools analysiert werden.

1. Vorstellung des Tools
   1. Ergebnis einer Masterarbeit von Silva
   2. Die Generierung erfolgt auf Basis eines PIM-Modells/Klassendiagramms
2. Vorstellung der USE-Syntax

1. Beschreibung der Anwendung
   1. 2-3 Sätze
   2. Klassendiagramm mit OCL-Constraints
2. Untersuchte Fragestellungen
   1. Wie bilden sich Relationen aus dem Klassendiagramm auf den Programm-Code ab?
   2. Wie werden die im Klassendiagramm festgelegten OCL-Constraints umgesetzt?
   3. Wird das MVVM Paradigma berücksichtigt?
   4. Wie schwierig gestaltet sich das nachträgliche Einfügen von Programm-Code?
      1. Kann einfache Berechnungslogik (durchschnittliche Varianzen und durchschnittliche gemessene Werte) evt. durch OCL-Contraints eingefügt werden? Wie schwierig ist das?
      2. Ist der generierte Code kommentiert?
      3. Sind Code-Konventionen eingehalten worden?
      4. Ist die Generierung nachvollziehbar/ Würde ein Entwickler es genauso oder ähnlich umsetzen?
   5. Wie schwierig gestaltet sich das Anpassen einer View-Komponente?
      1. Einfügen einer alternativen View für die Varianz-Abweichung
   6. Was passiert bei einer erneuten Generierung mit geändertem/ergänztem Code?
      1. Welche Möglichkeiten werden eingesetzt/kann man hier einsetzen?  
         Z. B. mögliche Lösungsansätze: Geschützte Bereiche (Protected Regions)  
         Trennung von manuellem und generiertem Code durch separate Dateien,  
         Vererbung/Entwurfsmuster wie Fabrikmethode, Schalbonenmethode,  
         Dependency Infjection ...
   7. Können unterschiedliche Berechnung durch Entwurfsmuster eingebracht  
      werden (z. B. Strategy)?
3. Ergebnisse und Diskussion
   1. Minus-Punkt: JUSE4Android wird nicht mehr gepflegt (letzten Stand github erwähnen)
   2. Im Zuge dieser Arbeit wurde festgestellt, dass die Persistenzkomponente DB4-bla nicht länger unterstützt wird, sodass die generierten Anwendungen nicht auf Devices mit einer Android-Version >Api Level 17 lauffähig sind