Haskell in practice: the Ampersand project
Dutch HUG day
April 20th, 2012
1. Ampersand is a declarative language, entirely built in Haskell, intended to automate the design of information systems and business processes (Stef Joosten)

2. A toy information system, generated by Ampersand, is demonstrated (Martijn Schrage)

3. Ampersand is used in the Dutch judiciary for the purpose of renewing the present information systems, 75 in total (Stef Joosten).
Ampersand, the language
What is an information system?

User interfaces

Data
An information system contains:

- data, which are meant to represent the truth according to its owner/patron;
- rules, which form an algebra;
- interfaces, which are meant to offer users a means to disclose and change the data, while maintaining the rules.
Relation algebra as a language

• Data: a set of binary relations whose contents may change over time
• Data is structured as a system of concepts, relations and rules that govern the relations.
• Every relation is given a type at define time, as in \( R :: A*B \).
• Every tuple \( <a,b> \in R \) implies \( a \in R \) and \( b \in R \).
• A rule is a term that must be true (by prescription).

• Rules are used in two different ways:
  • invariant, which means it must be true at all times
  • in a process, which means it must be true eventually.

• A rule may have violations, which can arise by changing the contents of relations
Characterization of Ampersand

- Relation algebra, because it has an interpretation as ontological language
- Representable algebra, because we want data in the database
- Heterogeneous algebra, because we want a strong type system
- Ampersand is declarative, meaning that a program consists solely of an unordered set of equations, which is sufficient to characterize the desired result.
- Ampersand is compositional language, meaning:
  \[(\text{let } S(a) \text{ be the semantics of script } a, \text{ and } S(b) \text{ the semantics of script } b)\]
  \[S(a \cup b) = S(a) \cap S(b)\]
A simple delivery system

Demonstrating a prototype generated by Ampersand
Application in practice
The judicial system in the Netherlands
Why Haskell?

- The Judiciary wants more control over her information systems at substantially lower cost.
- By the way, the process of jurisdiction may never be jeopardized.
- For this all information systems will need to behave as though it were one consistent collection of data.
- Contemporary ISDE’s cannot guarantee consistency
- Ampersand can guarantee consistency on the level of information system design
- Ampersand is built in Haskell
- In order to make Ampersand work for the Judiciary, some improvements, extensions, and corrections were (are are still) needed.
- So the Judiciary hired a Haskell expert…
Problem:

- 235 couplings between 133 systems, 75 of which are maintained by the judiciary. She communicates with 23 other institutions.
- Silo architecture dominates
- Even small changes are becoming increasingly costly to make.
Approach:

• A service bus as an enabler for gradual transformation towards “a single truth”.
Way of working

1. Make a semantic model of the law in Ampersand
2. Validate that model with legal experts of the judiciary
   a) by means of natural language, using the law as ultimate authority
   b) by means of prototypes, using stakeholders’ experience and intuition
3. Generate functional specifications from the semantic model, to feed the software factory
4. Use the prototypes as testing/acceptance device
Benefits for the judiciary

• Gradual improvement in small, manageable steps
• Ultimately 40% reduction of cost in IT maintenance.
• Improved quality of justice by reducing mistakes caused by erroneous or missing data
Summary
Summary

- Ampersand is a specification language,
- which is a (syntactically sugared) version of heterogeneous relation algebra,
- which has been used in practice several times,
- which helps requirements engineers get precise consistent functional requirements,
- which helps the business with working prototypes,
- which creates opportunities and hope for the “declarative movement”