

April 20th, 2012

Et → & → @ → & → &

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?

4

User interfaces



Data

A			B		
High-Level Requirement			Low-Level Requirement		
1	HRD 6.1.1		HRD 4.1.1.1.1		
2	HRD 6.1.1		(none)		
3	HRD 6.2.1		(none)		
4	HRD 6.3.1		(none)		
5	HRD 6.4.1		(none)		
6	HRD 6.5.1		HRD 4.5.1.4.1		
7	HRD 6.6.1		(none)		
8	HRD 6.7.1		(none)		
9	HRD 6.8.1		(none)		
10	HRD 6.9.1		(none)		
11	HRD 6.10.1		(none)		
12	HRD 6.11.1		HRD 4.11.1.1.1		
13	HRD 6.12.1		(none)		
14	HRD 6.13.1		(none)		
15	HRD 6.14.1		(none)		
16	HRD 6.15.1		(none)		
17	HRD 6.16.1		(none)		
18	HRD 6.17.1		(none)		
19	HRD 6.18.1		(none)		
20	HRD 6.19.1		(none)		
21	HRD 6.20.1		HRD 4.21.1.1.1		
22	HRD 6.21.1		(none)		
23	HRD 6.22.1		(none)		
24	HRD 6.23.1		(none)		
25	HRD 6.24.1		(none)		
26	HRD 6.25.1		(none)		
27	HRD 6.26.1		(none)		
28	HRD 6.27.1		(none)		
29	HRD 6.28.1		(none)		
30	HRD 6.29.1		(none)		
31	HRD 6.30.1		(none)		
32	HRD 6.31.1		(none)		
33	HRD 6.32.1		(none)		
34	HRD 6.33.1		(none)		
35	HRD 6.34.1		(none)		
36	HRD 6.35.1		(none)		
37	HRD 6.36.1		(none)		
38	HRD 6.37.1		(none)		
39	HRD 6.38.1		(none)		
40	HRD 6.39.1		(none)		
41	HRD 6.40.1		(none)		
42	HRD 6.41.1		(none)		
43	HRD 6.42.1		(none)		
44	HRD 6.43.1		(none)		
45	HRD 6.44.1		(none)		
46	HRD 6.45.1		(none)		
47	HRD 6.46.1		(none)		
48	HRD 6.47.1		(none)		
49	HRD 6.48.1		(none)		
50	HRD 6.49.1		(none)		
51	HRD 6.50.1		(none)		
52	HRD 6.51.1		(none)		
53	HRD 6.52.1		(none)		
54	HRD 6.53.1		(none)		
55	HRD 6.54.1		(none)		
56	HRD 6.55.1		(none)		
57	HRD 6.56.1		(none)		
58	HRD 6.57.1		(none)		
59	HRD 6.58.1		(none)		
60	HRD 6.59.1		(none)		
61	HRD 6.60.1		(none)		
62	HRD 6.61.1		(none)		
63	HRD 6.62.1		(none)		
64	HRD 6.63.1		(none)		
65	HRD 6.64.1		(none)		
66	HRD 6.65.1		(none)		
67	HRD 6.66.1		(none)		
68	HRD 6.67.1		(none)		
69	HRD 6.68.1		(none)		
70	HRD 6.69.1		(none)		
71	HRD 6.70.1		(none)		
72	HRD 6.71.1		(none)		
73	HRD 6.72.1		(none)		
74	HRD 6.73.1		(none)		
75	HRD 6.74.1		(none)		
76	HRD 6.75.1		(none)		
77	HRD 6.76.1		(none)		
78	HRD 6.77.1		(none)		
79	HRD 6.78.1		(none)		
80	HRD 6.79.1		(none)		
81	HRD 6.80.1		(none)		
82	HRD 6.81.1		(none)		
83	HRD 6.82.1		(none)		
84	HRD 6.83.1		(none)		
85	HRD 6.84.1		(none)		
86	HRD 6.85.1		(none)		
87	HRD 6.86.1		(none)		
88	HRD 6.87.1		(none)		
89	HRD 6.88.1		(none)		
90	HRD 6.89.1		(none)		
91	HRD 6.90.1		(none)		
92	HRD 6.91.1		(none)		
93	HRD 6.92.1		(none)		
94	HRD 6.93.1		(none)		
95	HRD 6.94.1		(none)		
96	HRD 6.95.1		(none)		
97	HRD 6.96.1		(none)		
98	HRD 6.97.1		(none)		
99	HRD 6.98.1		(none)		
100	HRD 6.99.1		(none)		
101	HRD 6.100.1		(none)		
102	HRD 6.101.1		(none)		
103	HRD 6.102.1		(none)		
104	HRD 6.103.1		(none)		
105	HRD 6.104.1		(none)		
106	HRD 6.105.1		(none)		
107	HRD 6.106.1		(none)		
108	HRD 6.107.1		(none)		
109	HRD 6.108.1		(none)		
110	HRD 6.109.1		(none)		
111	HRD 6.110.1		(none)		
112	HRD 6.111.1		(none)		
113	HRD 6.112.1		(none)		
114	HRD 6.113.1		(none)		
115	HRD 6.114.1		(none)		
116	HRD 6.115.1		(none)		
117	HRD 6.116.1		(none)		
118	HRD 6.117.1		(none)		
119	HRD 6.118.1		(none)		
120	HRD 6.119.1		(none)		
121	HRD 6.120.1		(none)		
122	HRD 6.121.1		(none)		
123	HRD 6.122.1		(none)		
124	HRD 6.123.1		(none)		
125	HRD 6.124.1		(none)		
126	HRD 6.125.1		(none)		
127	HRD 6.126.1		(none)		
128	HRD 6.127.1		(none)		
129	HRD 6.128.1		(none)		
130	HRD 6.129.1		(none)		
131	HRD 6.130.1		(none)		
132	HRD 6.131.1		(none)		
133	HRD 6.132.1		(none)		
134	HRD 6.133.1		(none)		
135	HRD 6.134.1		(none)		
136	HRD 6.135.1		(none)		
137	HRD 6.136.1		(none)		
138	HRD 6.137.1		(none)		
139	HRD 6.138.1		(none)		
140	HRD 6.139.1		(none)		
141	HRD 6.140.1		(none)		
142	HRD 6.141.1		(none)		
143	HRD 6.142.1		(none)		
144	HRD 6.143.1		(none)		
145	HRD 6.144.1		(none)		
146	HRD 6.145.1		(none)		
147	HRD 6.146.1		(none)		
148	HRD 6.147.1		(none)		
149	HRD 6.148.1		(none)		
150	HRD 6.149.1		(none)		
151	HRD 6.150.1		(none)		
152	HRD 6.151.1		(none)		
153	HRD 6.152.1		(none)		
154	HRD 6.153.1		(none)		
155	HRD 6.154.1		(none)		
156	HRD 6.155.1		(none)		
157	HRD 6.156.1		(none)		
158	HRD 6.157.1		(none)		
159	HRD 6.158.1		(none)		
160	HRD 6.159.1		(none)		
161	HRD 6.160.1		(none)		
162	HRD 6.161.1		(none)		
163	HRD 6.162.1		(none)		
164	HRD 6.163.1		(none)		
165	HRD 6.164.1		(none)		
166	HRD 6.165.1		(none)		
167	HRD 6.166.1		(none)		
168	HRD 6.167.1		(none)		
169	HRD 6.168.1		(none)		
170	HRD 6.169.1		(none)		
171	HRD 6.170.1		(none)		
172	HRD 6.171.1		(none)		
173	HRD 6.172.1		(none)		
174	HRD 6.173.1		(none)		
175	HRD 6.174.1		(none)		
176	HRD 6.175.1		(none)		
177	HRD 6.176.1		(none)		
178	HRD 6.177.1		(none)		
179	HRD 6.178.1		(none)		
180	HRD 6.179.1		(none)		
181	HRD 6.180.1		(none)		
182	HRD 6.181.1		(none)		
183	HRD 6.182.1		(none)		
184	HRD 6.183.1		(none)		
185	HRD 6.184.1		(none)		
186	HRD 6.185.1		(none)		
187	HRD 6.186.1		(none)		
188	HRD 6.187.1		(none)		
189	HRD 6.188.1		(none)		
190	HRD 6.189.1		(none)		
191	HRD 6.190.1		(none)		
192	HRD 6.191.1		(none)		
193	HRD 6.192.1		(none)		
194	HRD 6.193.1		(none)		
195	HRD 6.194.1		(none)		
196	HRD 6.195.1		(none)		
197	HRD 6.196.1		(none)		
198	HRD 6.197.1		(none)		
199	HRD 6.198.1		(none)		
200	HRD 6.199.1		(none)		

How does Ampersand represent an information system?

5

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

6

- 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 $\langle a, b \rangle \in R$ 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

7

- 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:
(let $S(a)$ be the semantics of script a , and $S(b)$ 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?

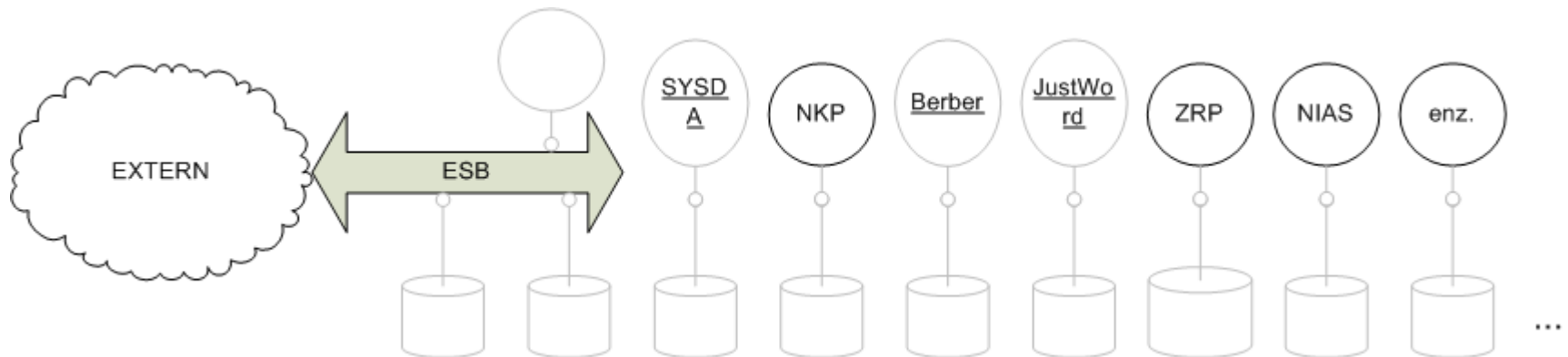
10

- 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:

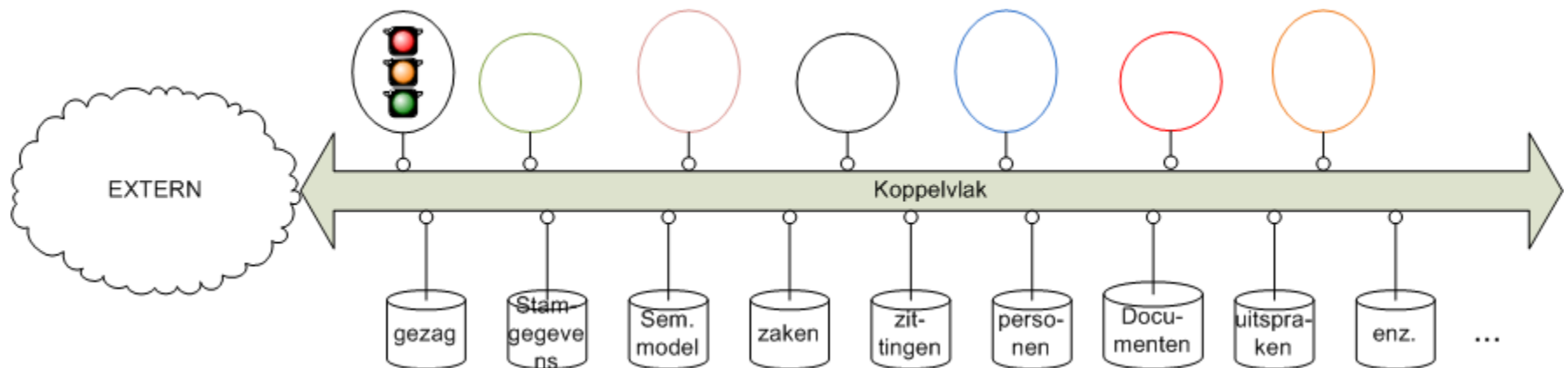
- 235 couplings between 133 systems, 75 of which are maintained by the judiciary. She communicates with 23 other institutions.
- Silo architecture dominates
- even small changes are becoming increasingly costly to make.



Approach:

12

- A service bus as an enabler for gradual transformation towards “a single truth”.



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

14

- 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

- 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”

CONNECTIVATE

www.ordina.nl