EXPLOITING PAYROLL KNOWLEDGE

Supporting decisions from an end-user point of view

Challenges when modelling knowledge
TEAL PARTNERS BACKGROUND

• Partner in elaborating digitalization tracks
• Goal: Putting business in control when legislation changes

- Expat Tax & Payroll software for 86 countries
- Wage cost budgeting simulation tool for Large Enterprises
- Build Custom calculation engine
  - Stateless .NET service
  - No business process flow
- Optimization track with KUL
- Learned about DMN
  • Partner in elaborating digitalization tracks
  • Goal: Putting business in control when legislation changes
CONTEXT – PAYROLL/HR IN BELGIUM

• Complex domain
  • Six governments
  • Third place Global Payroll Complexity

• High taxes
  • Second highest taxed country in Europe
  • Multiple legal statutes
    • Different taxation
    • Different social security

=> Uncertainty about choosing most beneficial statute
OBJECTIVES

• Show setup simulation tool
  • Short demo
  • General architecture
  • Optimization solution

• Different route to similar destination

Started from process logic
End 2 end production focus

Process management
100% decision completeness
GENERAL ARCHITECTURE
EXPLOIT KNOWLEDGE

• Knowledge
  • Declarative set of rules
  • Structured data
    • Descriptive tables
    • Clear in-and outputs
  • Types with meta info

• Optimal Decision taking
  • How many days should I work to earn a certain income
  • What is the optimal day rate to minimize taxes and earn a certain income
OPTIMALIZATION SOLUTION (1/3)

• “Brute Force”
  • Simple scenarios are possible

• Started with miniZinc
  • Widely used for constraint programming
  • Has built-in optimization
  • No solver suited for our domain
    • Decimal numbers are not well supported
    • Non-linear constraints, not always possible to linearize.

```plaintext
variable dailyWage: Int;
variable workingDays: Int;
variable incomeNetto: Int;
variable incomeBrutto: Int;
constraint incomeBrutto = dailyWage * workingDays;
constraint incomeNetto =
    if incomeBrutto > X then incomeBrutto * p1
    else incomeBrutto * p2;
constraint incomeNetto > 50000;
constraint incomeNetto < 70000;
constraint workingDays < 150;
solve maximize incomeNetto;
```
OPTIMALIZATION SOLUTION (2/3)

Used Z3 constraint solver

- Fast in determining (un)satisfiability
- Compile model to SMT-LIB2
  - JIT compilation with input values
  - Limit search space
- Custom optimization algorithm
- Created by Microsoft

```plaintext
(declare-const PercentageForfait_BasisSocialeBijdragen Real)
(declare-const MaximumForfait_BasisSocialeBijdragen Real)
(declare-const Basis_BasisSocialeBijdragen Real)
(declare-const Forfait_BasisSocialeBijdragen Real)
(assert (= BasisForfaitaireBeroepskosten_BasisSocialeBijdragen
            (+ (+ JaarloonBedrijfsleider_BedrijfsleiderLoon 252.0) 2200.0)))
(define-fun min_Real ((x Real)(y Real)) Real
    (ite (< x y) x y))
(assert (= Forfait_BasisSocialeBijdragen
               MinimumForfait_BasisSocialeBijdragen
               (let ([PercentageForfait_BasisSocialeBijdragen
                      (assert (= BasisForfaitaireBeroepskosten_BasisSocialeBijdragen
                                 Forfait_BasisSocialeBijdragen))])))
```
• Multiple dynamic parameters => pareto front
SIGHTSEEING TIPS FROM OUR JOURNEY

Modeling
- Unified DRD and Decision tables
- Library blocks for re-use
- Decision can have multiple outputs

Platform
- Transparency
- Model specific code generation is super fast
- Cloud scalability

Versioning
- Simulations
- Auditing (functional & technical)
- Easy for test/acc/prod environments

Enterprise
- Draft/publish flow
- Multilanguage
- Reference data (dynamic forms)
CONTACT INFORMATION

Teal Partners

Van de Wervestraat 20 - bus 206
2060 Antwerpen

http://www.tealpartners.com
info@tealpartners.com

Your contact

Tim Stuyckens
Software Product Gardener

tim@tealpartners.com
BACKUP SLIDE: WHERE WE HEADING NEXT

• Comparing versions
  • Git like functionality to “merge” knowledge
  • “apply patches” to propagate new knowledge

• DMN compliancy
  • Decision tables: extra abstraction above blocks and tables
    • Auto convert them to switch block
    • Limit formula syntax to be in sync with FEEL