Ruleflow or Decision Model?

Kayla Martens

Software Engineer

Guilhem Molines

Decision Product Architect

Agenda

- Introduction speakers
- Business Context
- Ruleflow
- Decision Model
- Combining them
- Conclusion
- Q&A

Kayla Martens

- **Software Engineer** for Principal
  - Decision Management
  - Decision Modeling
  - Rules Developer
  - Prior Business Analyst

www.linkedin.com/in/kaylaemartens

Guilhem Molines

- **Product Architect** for IBM Decision
  - Decision Management
  - Decision Modeling
  - Knowledge Representation
  - Rule Languages
- Associate Teacher at Polytech Sophia - Engineering School

@GuilhemMolines

linkedin.com/in/guilhemmolines

Business Context

- Principal Financial Group
  - A retirement services, insurance solutions, and asset management services company headquartered in Des Moines, Iowa, USA
- Application for Life Insurance Coverage
  - Application Part A: Requested coverage details and details about the applicant
  - Application Part B: Activity, health habits, and medical history details about the applicant used for underwriting risk class determination
- Decision usage
  - Using IBM Operational Decision Manager since 2011, and on Cloud since 2017
  - Used Sapiens Decision from 2014-2018
  - Decision Model in Visio/Gliphy and Excel for initial research

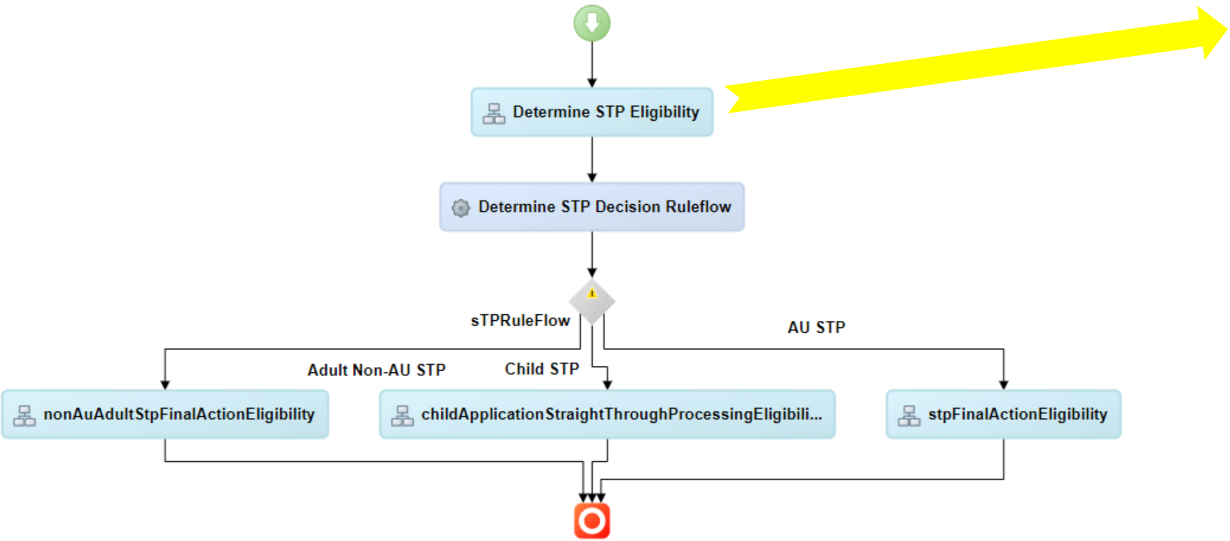
Ruleflow

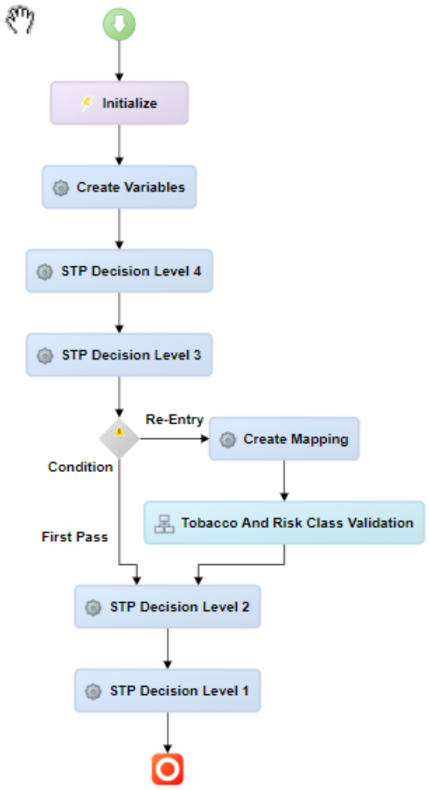
- Well understood concept in the industry
- Bottom-up approach
  - requires developer to define object model and RF structure
  - then rule writer can jump in for authoring rules
- Pros:
  - well structured, guides the user in a procedural approach
    - (do *this*, then do *this*, then do *that*)
  - Easy to know where to add new logic
  - Branches / conditions to exit or skip steps
  - Flexible: can adapt easily to various organizations
- Cons:
  - Can't model on-the-fly while having conversation with business users
  - Requires more technical skill set to get started

Ruleflow

PFG use case

- Deciding if an insurance application is eligible to be reviewed by the system or needs be routed to a human underwriter


Determine STP Eligibility
Determine STP Decision Ruleflow
sTPRuleFlow
AU STP
Adult Non-AU STP
Child STP
nonAuAdultStpFinalActionEligibility
childApplicationStraightThroughProcessingEligibili...
stpFinalActionEligibility



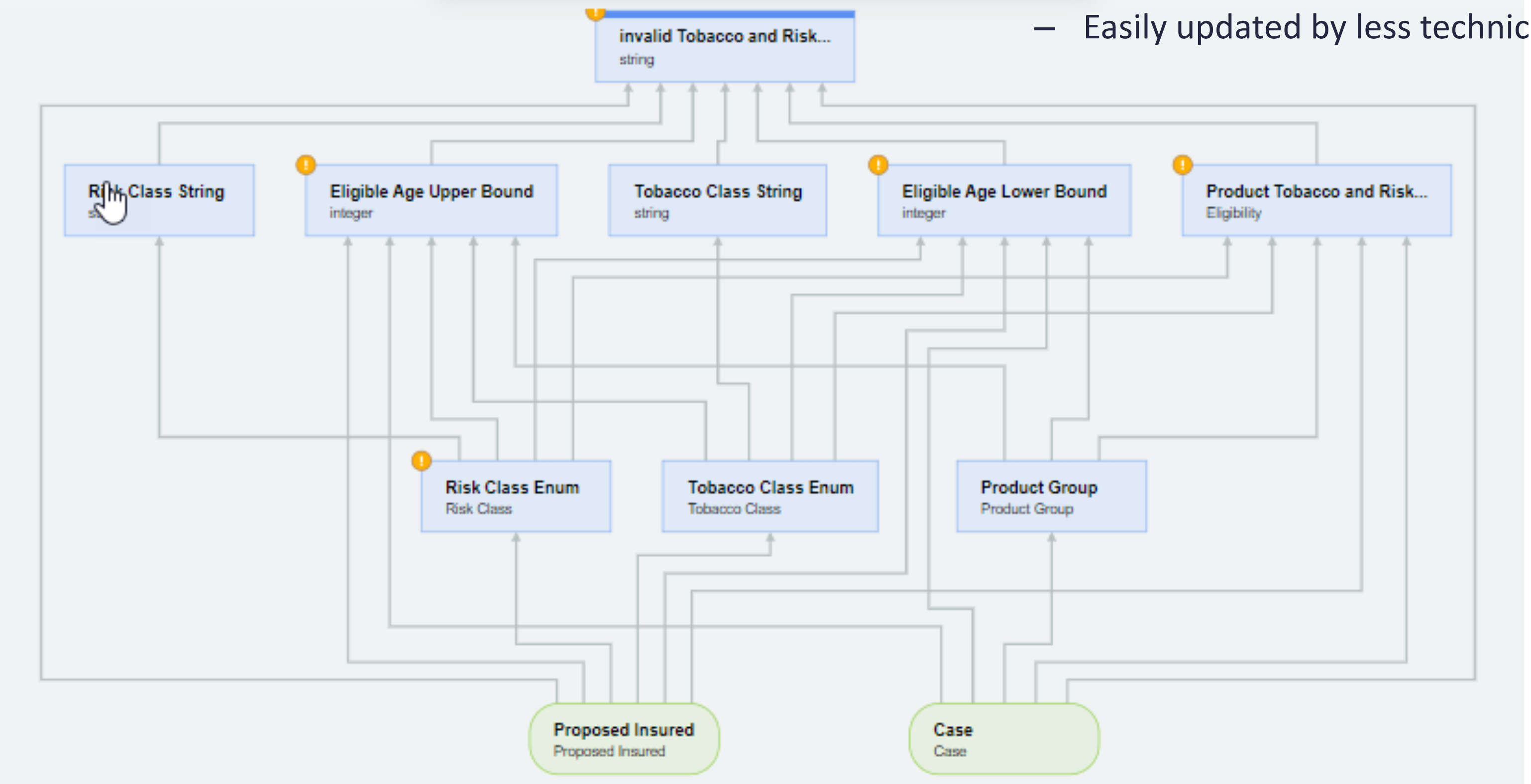
Decision Model

- Top-down approach
- Requires "functional" thinking
  - "what data does this decision depend on", rather than "how do I compute this data"
- Pros:
  - Can be used while talking with business analysts
  - Lower reliance on technical rule developer
- Cons:
  - decomposition as a DRD not very natural/intuitive, requires some getting used to

Decision Model

PFG use case

- Compares selected product, risk class, tobacco class, and proposed insured age to make sure combination is allowed
- Updated frequently with new products
- Easily updated by less technical users

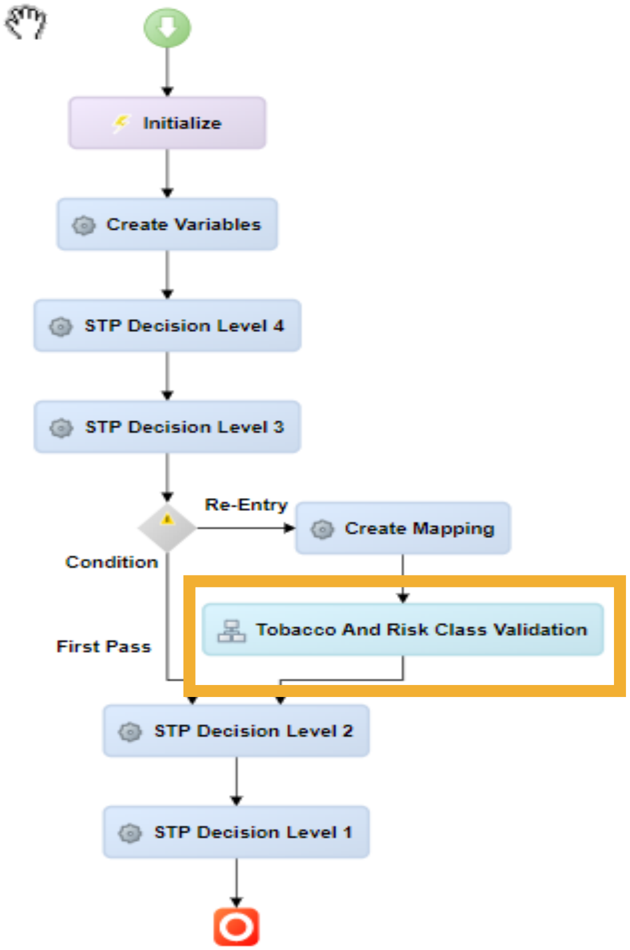


Combining them

- Extending a "legacy" service with a new part, designed top-down
- DM used for a "small" decision, inserted within several large and complex ruleflows
- Pros:
  - Decision Model reused and shared into several existing ruleflows
  - Easier for involving non-technical users
- Cons:
  - Added complexity, need to have one call the other at runtime
  - Need to manually link the lifecycles of the ruleflow and the decision model

Combining them

- Tobacco and Risk Class Validation logic is used here and elsewhere in PFG systems



Conclusion

- Ruleflows and Decision Models are two possible, complementary tools in a toolbox, each with pros and cons.

- Lessons learned:
  - Ruleflow risk growing too large and complex, and without componentization become too large as a communication vehicle with the business
  - Great value in modeling small decisions, that can be reused across the system. Makes it easier to start new projects by assembling already existing decisions
  - Componentization of decisions makes it easier to amend the overall logic when new life insurance products are released

Q&A

Thank you

Principal®