

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Analýza strategií Formule 1: DMN modelovací přístup
Diplomová práce

Autor: Bc. Lukáš Janouch
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Ing. Hana Tomášková Ph.D.

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 7.8.2023

Lukáš Janouch

Poděkování:

Děkuji vedoucí diplomové práce doc. Ing. Haně Tomáškové, Ph.D. za odborné vedení, cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování diplomové práce.

Anotace

Diplomová práce se zaměřuje především na využití DMN, konkrétně jako nástroje pro tvorbu strategie ve Formuli 1. V teoretické části práce je popsán princip DMN a konverzačních a choreografických diagramů, které zahrnuje BPMN. Tato část také poskytuje srovnání některých nástrojů, v nichž lze vytvářet modely DMN a BPMN. Následně jsou uvedeny tři různé případy užití DMN a poté je rozebrána aplikační oblast diplomové práce. Praktická část popisuje modely, vytvořené pro zmíněnou aplikační oblast. Nejprve jsou popsány použité datové typy a elementy diagramu DMN a dále se tato část věnuje konverzačnímu a choreografickému diagramu, které byly vymodelovány pro dokreslení situace. V závěru jsou popsány výsledky spuštění modelu DMN.

Klíčová slova: BPMN, DMN, FEEL, KIE Sandbox, Formule 1

Annotation

Title: Formula 1 Strategies analysis: A DMN Modeling Approach

The diploma thesis focuses mainly on the use of DMN. Specifically, it applies it as a strategy tool in Formula 1. The theoretical part of the thesis describes the principle of DMN and conversational and choreographic diagrams. The last two diagrams mentioned are included in BPMN. This section also provides a comparison of some of the tools in which DMN and BPMN models can be created. Subsequently, three different cases of using DMN are presented and then the application area of the diploma thesis is analyzed. The practical part describes the models created for the mentioned application area. First, the used data types and elements of the DMN diagram are described, and then this part is devoted to the conversational and choreographic diagram, which were modeled to illustrate the situation more. In the conclusion, the results of running the DMN model are described.

Keywords: BPMN, DMN, FEEL, KIE Sandbox, Formula 1

Obsah

1	Úvod.....	1
2	Cíl práce a metodika zpracování	2
3	DMN	3
3.1	Vztah k BPMN	3
3.2	Úroveň požadavků rozhodování	4
3.3	Úroveň rozhodovací logiky	5
3.3.1	Datové typy jazyka FEEL	6
3.3.2	Funkce.....	8
3.3.3	Boxed expressions.....	8
4	Konverzační a choreografické diagramy.....	11
5	Vývojová prostředí	13
5.1	Enterprise architect.....	13
5.2	Camunda.....	13
5.3	Trisotech a další nástroje	14
5.4	KIE Sandbox.....	14
6	Využití DMN.....	16
6.1	Modelování variability a podnikový software	16
6.2	Big data a monitorování přírodních katastrof	18
6.3	DMN a PROMETHEE v medicíně	19
7	Strategie ve Formuli 1	22
7.1	Tvorba strategie před závodem.....	22
7.2	Tvorba strategie během závodu	23
7.3	Stratégové	24
7.4	Pneumatiky	24
8	Analýza strategie Formule 1 pomocí DMN.....	26

8.1	Datové typy	26
8.2	Elementy	27
8.2.1	Rozhodnutí a bussiness knowledge modely	27
8.2.2	Rozhodnutí <i>Pit when safety car deployed?</i>	28
8.2.3	Rozhodnutí <i>How much places will we drop?</i>	30
8.2.4	Rozhodnutí <i>Is current lap near planned pit stop?</i>	31
8.2.5	Rozhodnutí <i>Planned pit stops</i>	31
8.2.6	Rozhodnutí <i>Unexpected situation happened</i>	31
8.2.7	Rozhodnutí <i>Tyre used in the 1st stint</i>	32
8.2.8	Rozhodnutí <i>View actual strategy</i>	32
8.2.9	Rozhodnutí <i>View prefered strategies</i>	33
8.2.10	Rozhodnutí <i>Current lap percentage</i>	33
8.2.11	Bussiness knowledge model <i>get current lap percentage</i>	34
8.2.12	Rozhodnutí <i>Prefered strategies</i>	35
8.2.13	Bussiness knowledge model <i>choose prefered strategies</i>	35
8.2.14	Rozhodnutí <i>Actual strategy</i>	36
8.2.15	Bussiness knowledge model <i>get actual strategy</i>	36
8.2.16	Rozhodnutí <i>New strategy</i>	37
8.2.17	Bussiness knowledge model <i>make new strategy</i>	38
9	Doplnění modelu DMN konverzačním a choreografickým diagramem	40
10	Shrnutí výsledků	42
11	Závěry a doporučení	44
12	Seznam použité literatury	45
13	Přílohy	48

Seznam obrázků

Obrázek 1 DMN ve vztahu k BPMN a úrovně DMN [3].....	4
Obrázek 2 Konverzační diagram [8].....	11
Obrázek 3 Choreografický diagram [9]	12
Obrázek 4 Model BPMN procesu objednávky [16]	17
Obrázek 5 DRD procesu objednávky [16]	17
Obrázek 6 Model oDMN+ o zahájení varování [17].....	19
Obrázek 7 Model DMN procesu diagnózy mozkové příhody [18]	21
Obrázek 8 Očekávané strategie pro VC Monaka 2022 [20].....	23
Obrázek 9 Vyvolání bussiness knowledge modelu.....	28
Obrázek 10 Rozhodnutí <i>Tyre used in the 1st stint</i>	32
Obrázek 11 Rozhodnutí <i>View actual strategy</i>	33
Obrázek 12 Rozhodnutí <i>View prefered strategies</i>	33
Obrázek 13 Rozhodnutí <i>Current lap percentage</i>	34
Obrázek 14 Bussiness knowledge model <i>get current lap percentage</i>	34
Obrázek 15 Rozhodnutí <i>Prefered strategies</i>	35
Obrázek 16 Rozhodnutí <i>Actual strategy</i>	36
Obrázek 17 Rozhodnutí <i>New strategy</i>	37
Obrázek 18 Konverzační diagram tvorby závodní strategie	40
Obrázek 19 Choreografický diagram situace při poškození vozu	41
Obrázek 20 Ukázka vstupů a výstupů.....	43

1 Úvod

Při úspěchu v závodě hraje ve Formuli 1, ale i v dalších soutěžích motorsportu významnou roli zvolená strategie. Nečekané události, které nastanou během závodu, vedou k potřebě ji změnit. Strategie se také musí promyslet s velkým předstihem už před závodním víkendem. Ovlivnit ji lze výběrem pneumatik a určením kol, ve kterých se provedou zastávky v boxu k jejich výměně. Zejména při závodě je pak potřeba provádět správná rozhodnutí, jež musí být navíc rychlá, protože o rychlost jde především. Závod tedy není pouze o rychlosti vozů, ale také o rychlosti a správnosti rozhodování při tvorbě strategií. K tomu inženýrům v motorsportu pomáhají simulační systémy. Tato práce popisuje jeden ze způsobů, jakým lze systém pro automatické určování strategií vytvořit. Ukazuje, jak je možné, že podle popisu aplikační oblasti mohou vzniknout algoritmy, které budou samy řešit problémy, jež v ní nastanou.

Konkrétně je systém vytvořen pomocí standardu DMN (Decision Model and Notation). Pomocí něj lze vytvářet diagramy rozhodování. Standard definuje také programovací jazyk FEEL (Friendly Enough Expression Language), který byl vytvořen pro psaní výrazů v rozhodovacích uzlech DMN, aby byly tyto výrazy lehce čitelné pro co nejširší spektrum programátorů a analytiků. DMN se používá k tvorbě rozhodovacích systémů, které lze použít tam, kde je potřeba se rychle a zároveň správně rozhodnout. V teoretické části je využití více popsáno, aby o něm mohl čtenář případně získat konkrétnější představu. DMN se využívá často jako doplněk k procesnímu diagramu standardu BPMN, ale lze jej použít i samostatně. V této práci jde především o modelování rozhodování, a tak procesní diagram není tolik potřeba. Standard BPMN je tu ale také využit, a to v podobě dvou diagramů, které pod něj spadají – konverzačního a choreografického, aby byla znázorněna také komunikace aktérů rozhodování.

2 Cíl práce a metodika zpracování

Cílem práce je vytvořit rozhodovací model, který by mohl sloužit k vytváření strategií ve Formuli 1. Jeho účelem bude především navrhnout rozhodnutí s vysokou pravděpodobností správnosti během velice krátké reakční doby od vzniku neočekávané situace během závodu. Smyslem toho je také demonstrovat použití DMN a konverzačních a choreografických diagramů, které model DMN pro lepší představu o situaci doplní.

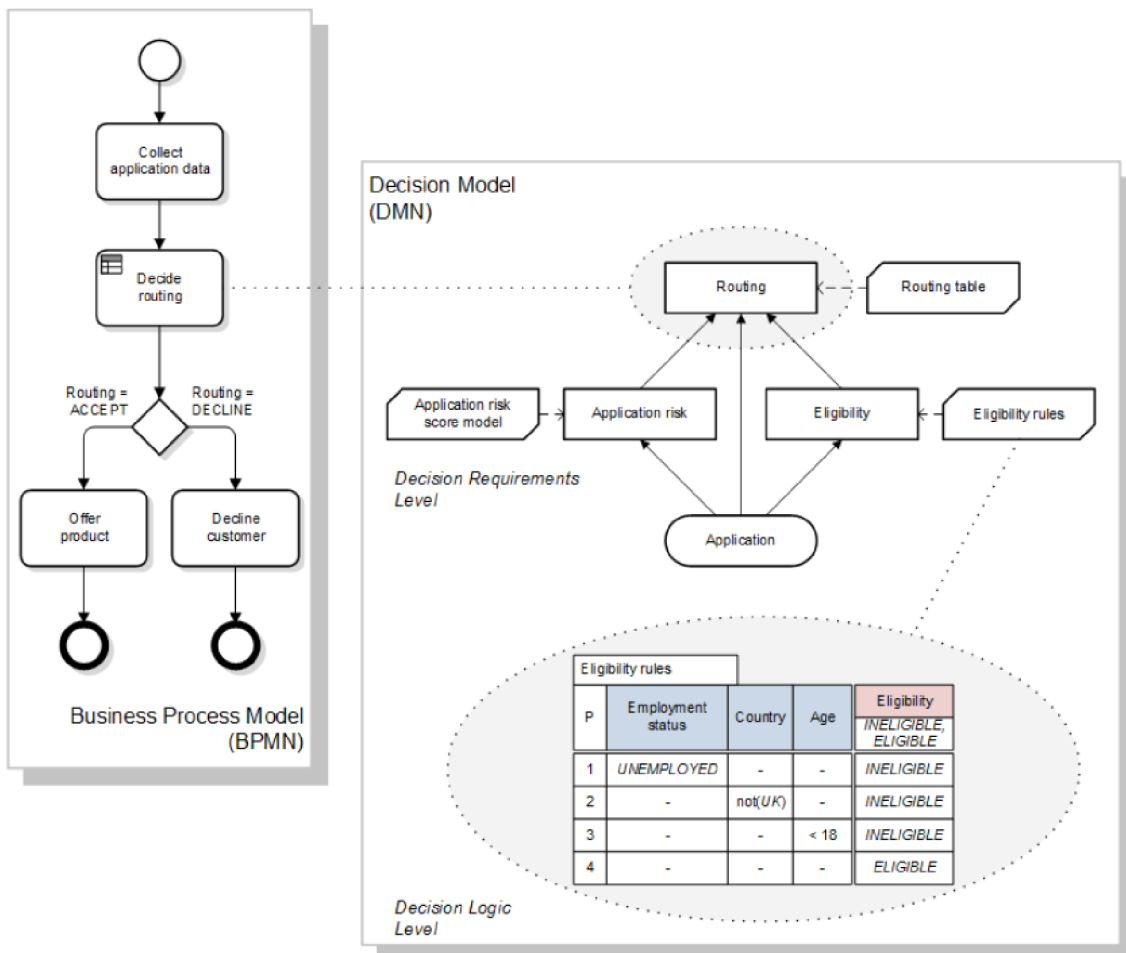
Cíle bylo dosaženo pomocí literární rešerše a modelování. Většina zdrojových článků byla napsána relativně nedávno a jsou především internetové. Jedná se o oficiální dokumentace, odborné články a důvěryhodné weby zabývající se Formulí 1. Po uchopení principů modelování ve zvoleném vývojovém prostředí a jevů pozorovaných v literatuře o Formuli 1 byl vytvořen model DMN pomocí nástroje KIE Sandbox, na kterém byla provedena simulace na reálných vstupech. Tento model byl doplněn konverzačním a choreografickým diagramem, vytvořenými v nástroji Enterprise Architect.

3 DMN

DMN je dle [1] standard oficiálně publikovaný skupinou OMG (Object Management Group) v září 2015. Dle [2] je OMG neziskové mezinárodní konsorcium, jehož členy je více než 230 organizací z 27 zemí. Vzniklo v roce 1989. K členům OMG patří prodejci, koncoví uživatelé, akademické instituce a vládní agentury. Cílem OMG je generovat technologické standardy, kterých bylo vytvořeno už přes 250. Standardy vznikají diskusí nad osvědčenými způsoby. Při ní má každá organizace bez ohledu na velikost jeden hlas. Standardy OMG jsou implementovány v oblasti zemědělství, autonomních vozidel, avioniky (vybavení letadel elektronikou), obchodu, energetiky, financí, zdravotnictví, vojenství, maloobchodu, telekomunikace, dopravy a vesmíru.

3.1 Vztah k BPMN

Oficiální dokumentace DMN [3] uvádí, že účelem tohoto standardu je poskytnout model, který dokáže podrobně znázornit rozhodování v procesech. Procesy mohou být namodelovány v procesním diagramu BPMN a DMN může tento diagram dále rozvíjet, ale nemusí, může se použít i samostatně. DMN znázorňuje rozhodování natolik detailně, že je pak možné provést validaci i automatizaci rozhodování. Model zahrnuje dvě úrovně: úroveň požadavků rozhodování a úroveň rozhodovací logiky. Vztah BPMN k DMN, strukturu DMN a příklady obou diagramů ukazuje obrázek 1.



Obrázek 1 DMN ve vztahu k BPMN a úrovně DMN [3]

BPMN (Business Process Model and Notation) bylo dle [4] vyvinuto institutem BPMI (Business Process Management Institute) v roce 2004. Tento institut se později sloučil s OMG a od roku 2006 je BPMN standardem OMG a dále se rozvíjí. Nejnovější specifikace 2.0.2 je z roku 2014.

3.2 Úroveň požadavků rozhodování

Základní jednotkou rozhodovacího modelu je dle [3] rozhodnutí. Tím je proces určení výstupu z určitého počtu vstupů pomocí vnitřní logiky. Často je vstup i výstupem jiného rozhodnutí. Pokud to tak není, nazývá se tento element „vstupní data“ (input data) a značíme ho oválem. Rozhodovací logika může být obsažena v elementu „rozhodnutí“ (decision) nebo v elementu „business knowledge model“. Rozhodnutí znázorňujeme obdélníkem a bussiness kowledge model obdélníkem se dvěma oříznutými rohy. Pravidla pro používání těchto elementů nejsou striktně

dána a jejich aplikování záleží na zvoleném stylu a metodologii autora modelu. Oba druhy elementů na sebe můžeme tedy libovolně větvit nebo také bussiness knowledge modely vynechat. Vnitřní logiku bussiness knowledge modelu lze znovupoužít vícero rozhodnutími. Dále lze do modelu přidat i element „knowledge source“, odkazující na osoby, které rozhodovací logiku vymyslely nebo ji vykonávají, či na dokumenty, z nichž je logika odvozena. Značíme ho obdélníkem, jehož spodní strana je vlnitá. Existuje i znovupoužitelný element „rozhodovací služba“ (decision service), který propojuje více rozhodnutí dohromady. Může ho interně využít jiné rozhodnutí, nebo dokonce externě task z diagramu BPMN. Znázornit ji lze obdélníkem s kulatými rohy rozděleným čarou na dvě poloviny. Horní polovina zahrnuje výstupní rozhodnutí, spodní zahrnuje vstupní.

Existují čtyři druhy vazeb, které mezi elementy mohou existovat. Vazba „informační požadavek“ (information requirement) propojuje vstup a rozhodnutí. Značíme ji plnou čarou se šipkou na konci. „Požadavek znalosti“ (knowledge requirement) vyjadřuje vyvolání funkce bussiness knowledge modelu vnitřní logikou rozhodnutí nebo rozhodovací služby. Je to přerušovaná čára se šipkou na konci. Třetí vazba – „požadavek autority“ (authority requirement) znázorňuje propojení mezi kterýmkoli elementem a zdrojem znalosti. Od předchozí vazby se liší tím, že má na konci kruh místo šipky. Elementy můžeme také opatřit textovou anotací, která se napojuje tečkovanou vazbou asociace.

Síť propojených rozhodnutí se nazývá graf požadavků rozhodování (Decision Requirements Graph) a nachází se v diagramu požadavků rozhodování (Decision Requirements Diagram).

3.3 Úroveň rozhodovací logiky

Tato úroveň umožňuje případnou automatizaci. Každé rozhodnutí či každý bussiness knowledge model je zde reprezentován hodnotovým výrazem, který se znázorňuje pomocí „boxed expression“. V elementu bussiness knowledge modelu je hodnotový výraz vyjádřen funkcí a vstupy elementu jsou její parametry. Funkce je pak vyvolána rozhodnutím. Boxed expressions obsahují tzv. „literal expressions“, což je text, který popisuje vznik výstupu ze vstupu. Mohou být spustitelné, ale nemusí. Tento text lze tedy zapsat například jazykem FEEL nebo Javou, ale

i angličtinou nebo češtinou. FEEL je v DMN výchozím jazykem. Byl inspirován jazyky, jako je Java, JavaScript, XPath, SQL, PMML, Lisp a další.

K vyjádření vnitřní logiky dále slouží také proměnné. Jsou to odkazy na vstupy, které mohou být, jak už bylo zmíněno, ve formě elementu vstupních dat nebo rozhodnutí. Proměnná je se vstupem propojena již zmíněnou vazbou informačního požadavku. Ve vnitřní logice svého elementu musí mít unikátní název. U bussiness knowledge modelu jsou proměnné parametry jeho funkce. Posledním elementem, který tvoří vnitřní logiku, je „definice položky“ (item definition). Ta určuje strukturu a typ dat. Mezi datové typy jazyka FEEL patří number, string, boolean, days and time duration, years and months duration, time a date and time. Struktury se skládají z těchto jednoduchých datových typů. K jednotlivým položkám struktury se ve výrazech přistupuje přes tečkový operátor. Všechna vstupní data, proměnné a hodnotové výrazy odkazují na svou definici položky.

3.3.1 Datové typy jazyka FEEL

Jak bylo uvedeno v úvodu této kapitoly, jazyk FEEL má osm datových typů. Datové typy popisují množinu hodnot. Ty následující jsou disjunktní.

Čísla jsou tedy v jazyce FEEL reprezentována typem number. Je založen na formátu Decimal 128 (čtyřnásobná přesnost) standardu IEEE 754-2008. Tento standard mimo jiné zahrnuje dle [5] ještě formáty Decimal 32 (základní přesnost) a Decimal 64 (dvojitá přesnost). Ve formátu Decimal 32 je například reprezentován javovský typ float. Datový typ number v jazyce FEEL vznikl omezením datového typu XML precisionDecimal a je ekvivalentem javovského typu BigDecimal s MathContextem DECIMAL128. BigDecimal se v Javě používá pro výpočty týkající se financí. Datový typ number nepodporuje $-INF$, $+INF$ ani NaN. Místo toho se používá null. Nerozlišuje ani mezi 0 a -0 . To, co bylo zmíněno v předchozích dvou větách, podporuje neomezený datový typ XML precisionDecimal. FEEL dále nepodporuje vědecký zápis čísel, to znamená, že např. $1.2e3$ je nutné zapsat jako $1.2*10^{**3}$. K oddělení celočíselné části a zlomku používá datový typ number tečku. String je sekvence znaků ve dvojitých uvozovkách. Například "abc". Proměnná typu boolean může dle [6] nabývat hodnot true, false nebo null.

Proměnná typu `date` je reprezentována pomocí stringu funkce `date()`. Parametr funkce má tvar "YYYY-MM-DD", kde YYYY je rok, MM měsíc a DD den např. `date("2017-06-23")`. Hodnota typu `date` je ekvivalentem hodnoty typu `date and time` s časem (00:00:00) UTC (koordinovaného světového času). Obdobně jsou proměnné typu `time` reprezentovány stringem a funkcí `time()`. Parametr funkce má tvar "hh:mm:ss[.uuu][(+)-hh:mm]". Jsou v něm tedy zahrnuty hodiny (hh), minuty (mm), sekundy (ss) a dobrovolně milisekundy (uuu). Druhá dobrovolná část parametru určuje časovou zónu. Pokud chceme čas UTC, zadáme -00:00 nebo z. Pokud tuto část parametru zcela vynecháme, uvažujeme místní čas. Místní čas a čas s uvedenou časovou zónou nelze mezi sebou porovnávat. Proměnné typu `date and time` se opět vytvářejí pomocí stringového parametru a funkce `date and time()`. Parametr má tvar "<date>T<time>", kde <date> a <time> jsou uvedeny v již zmíněných formátech a propojuje je T.

Dva poslední datové typy určují dobu trvání. Hodnoty těchto typů jsou opět vyjadřovány pomocí stringového parametru a funkce. Parametr funkce `date and time duration()` stejnojmenného datového typu zahrnuje počet dní, hodin, minut a sekund. Zápis potom může vypadat například následovně: `duration("P30DT11H10M30S")` `duration("P40D")` `duration("PT10H")` `duration("PT35M")`. Doby trvání lze mezi sebou sčítat, odčítat, násobit a dělit a zároveň ji lze vyjádřit i v sekundách. Např.: `duration("PT9M5S") + duration("PT2S") = 1090`. Pokud bychom sčítali pouze sekundy nebo pouze minuty či hodiny nebo dny, měl by výsledek tvar parametru funkce, ne jen samotného čísla. Obdobně můžeme použít funkci `duration` pro vyjádření doby trvání v jednotkách měsíců a dní u datového typu `years and time duration`. Sem lze ale zahrnout pouze roky a měsíce a časové jednotky dvou posledních datových typů nelze použít ve funkci `duration()` najednou. Příklady zápisu vypadají takto: `duration("P2Y4M")` `duration("P3Y")` `duration("P19M")`. Délky trvání lze opět sčítat, odčítat, násobit a dělit. Při sčítání a odčítání má výsledek formu parametru funkce – např.: `duration("P9M") + duration("P1Y2M") = P1Y11M`. Při násobení a dělení je výsledek vyjádřen v měsících – např.: `duration("P9M") * duration("P1Y2M") = 126`.

3.3.2 Funkce

Nad proměnnými lze provádět různé operace. Již byly zmíněny časové funkce a v této kapitole bude uvedeno několik dalších funkcí, které jsou použity v praktické části. Například funkce *not(proměnná, hodnota)* zjišťuje, zda se daná proměnná nerovná dané hodnotě. Vrací *true*, pokud se proměnná této hodnotě nerovná. Zaokrouhlovat čísla lze pomocí funkce *floor(číslo)*.

Na seznamech lze provádět následující operace. Funkce *sum(seznam)* dokáže například sečíst prvky seznamu. Funkcí *count(seznam, hodnota)* získáme počet prvků dané hodnoty v daném seznamu. Pomocí funkce *contains(seznam, prvek)* můžeme zjistit, zda daný seznam obsahuje zadaný prvek. Index daného prvku lze zjistit díky funkci *index of(seznam, prvek)*. Použitím funkce *sublist(seznam, index)* vznikne seznam, který obsahuje prvky původního seznamu od zadaného indexu včetně a vynechá prvky před indexem. V jazyku FEEL se indexuje od jedničky, takže pokud zadáme dvojku, bude vynechán první prvek seznamu. Připojit prvek nakonec seznamu dokáže funkce *append(seznam, prvek)*.

Na vytvoření seznamu, který vznikne z původního seznamu z prvků, které vyhovují dané podmínce neexistuje žádná funkce, ale lze ho dle [6] vytvořit pomocí operátoru `[]` – tedy *seznam[podmínka]*.

3.3.3 Boxed expressions

Dle [7] rozlišujeme sedm druhů boxed expressions. Rozhodovací tabulka (decision table) mapuje vstupy na výstupy tak, že pro každý název proměnné definuje podmínkami možné hodnoty. Horizontální tabulka má názvy proměnných v prvním řádku a výstupy ve sloupci vpravo. Ve vertikální tabulce jsou názvy proměnných umístěny do prvních řádků prvního sloupce, výstupy se nachází v prvním sloupci pod nimi. Řádky tabulky tvoří obchodní pravidla (business rules). Pokud vstupy odpovídají vícero pravidlům, pak o tom, která pravidla se použijí, rozhodují zásady přístupu (hit policy). U horizontální tabulky se první písmeno zásady přístupu zapisuje do buňky v levém horním rohu, u vertikální do nejspodnější buňky v prvním sloupci. Výchozí zásada přístupu je *unique (U)*. Pak podmínkám může vyhovovat pouze jedno pravidlo, protože jsou všechna disjunktní. Pokud je zásada přístupu *any (A)*, pak by měly být výstupy, které vyhovují podmínkám, shodné a lze

použít kterýkoliv z nich. Pokud to tak není, musíme použít jinou zásadu přístupu. Priority (P) povoluje více různých výstupů a platit bude vždy první podle priority. Priorita je definována ve zvláštním seznamu a neodvozuje se podle pořadí řádků, ale podle pořadí povolených hodnot výstupů, zapsaných v záhlaví sloupce (u horizontální tabulky) nebo řádku (u vertikální tabulky) výstupů. Na pořadí řádků závisí zásada přístupu first (F). Výsledkem je zde první výstup, který vyhovuje podmínkám. Pak se vyhodnocování zastaví. Při zmíněných zásadách přístupu se vrátí vždy pouze jeden výstup, při následujících je vráceno více výstupů najednou. Pokud použijeme zásadu output order (O), budou vráceny všechny výstupy, které odpovídají podmínkám, seřazené v seznamu podle priority. Rule order (R) funguje podobně, ale výstupy se vrátí seřazené podle pořadí pravidel (řádků) v tabulce. Collect (C) vrací výstupy v náhodném pořadí. Za symbol C lze přidat ještě operátor, pak tabulka vrací pouze jednu hodnotu. Při použití operátoru + je vrácen součet všech výstupů. U < se vrací nejmenší hodnota z výstupů a u > největší. Pokud použijeme #, dostaneme počet hodnot výstupů. V případě, že má tabulka více různých proměnných výstupů, nejsou operátory u zásady přístupu collect podporovány.

Dalším druhem boxed expression, který lze použít, je boxed literal expression. V [4] se nazývá také jako boxed FEEL expression a „je to boxed expression, který obsahuje literal expression“. Má podobu jedné buňky s tímto výrazem, kterou doplňuje buňka záhlaví s názvem proměnné a jejím datovým typem. V buňce pro hodnoty může být literal expression nebo jakýkoliv jiný boxed expression.

Kontext (boxed context) je soubor dvojic n (název proměnné, hodnota). Každá dvojice se nazývá „kontextové zadání“ (context entry). Kontext může, ale nemusí obsahovat výsledkové pole. V něm se může pracovat s hodnotami, které kontext obsahuje, a kontext pak tento výsledek vrací. Pokud výsledkové pole neobsahuje, je každá hodnota kontextu přístupná přes tečkový operátor. To se hodí, pokud chceme reprezentovat nebo vytvořit strukturu a dále s ní pracovat.

Invocation slouží rozhodnutím k získání výsledku funkce bussiness knowledge modelu nebo rozhodovací služby. V záhlaví obsahuje kromě názvu elementu rozhodnutí název vyvolávaného bussiness knowledge modelu a pod ním ve sloupečku vlevo názvy parametrů a ve sloupečku vpravo jejich hodnoty. Relace má

ve svém záhlaví názvy proměnných a v buňkách pod ním jejich hodnoty. Neobsahuje žádnou buňku s výsledkem. Má podobu tabulky relační databáze.

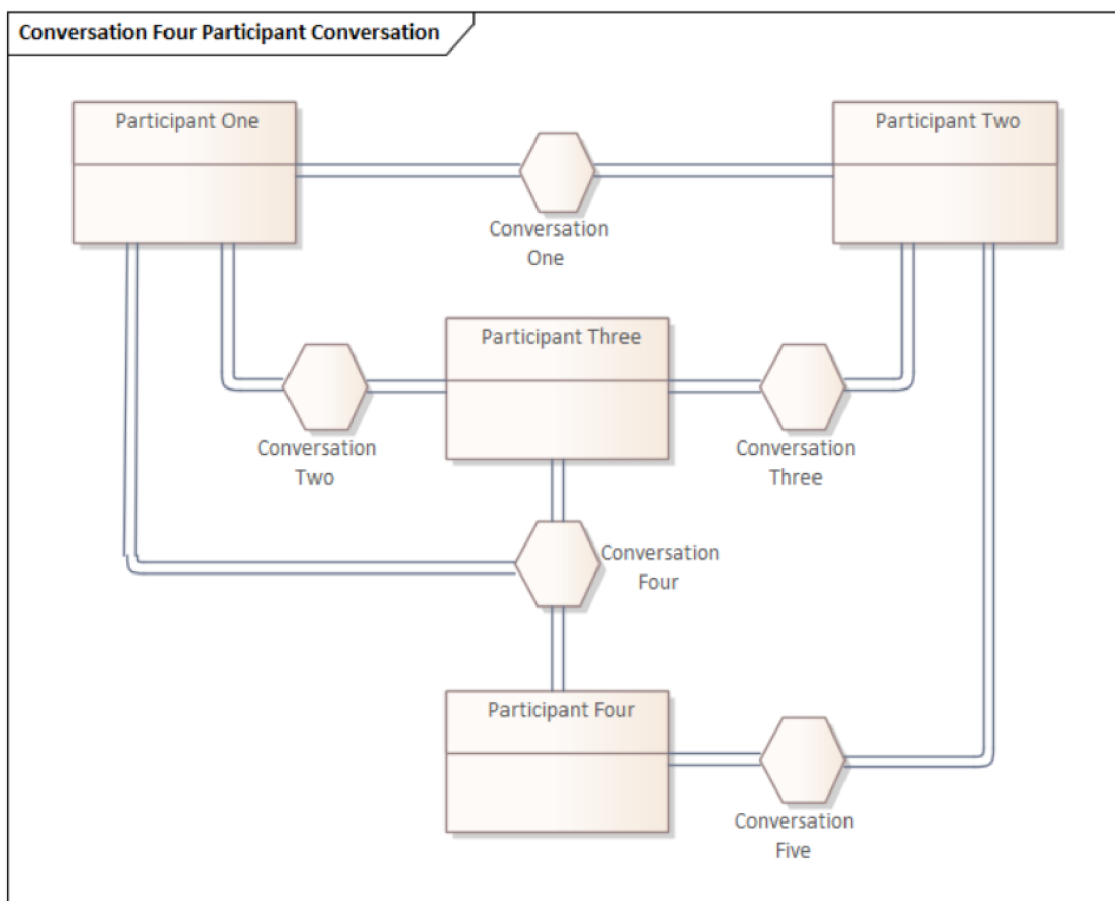
Funkce je parametrizovaný boxed expression. Ve výchozím stavu je každý bussiness knowledge model definován jako funkce, jejímiž parametry jsou vstupy modelu. Boxed function obsahuje tři buňky. V první je zadán začínajícím písmenem jazyk funkce – FEEL (F), JAVA (J), PMML (Predictive Model Markup Language) (P). Ve druhé jsou vypsané parametry a ve třetí je tělo funkce, kde může být jakýkoliv boxed expression nebo kontext obsahující funkci Javy nebo model PMML. Pokud se jedná o Javu, obsahuje kontext dvě hodnoty – třídu a funkci ze zadané třídy. Pokud použijeme PMML, zadáme jako první proměnnou kontextu dokument PMML a jako druhou můžeme zadat model ze zadaného dokumentu. Toto pole lze nechat také volné a pak se použije první model v dokumentu. Pro převod argumentů mezi jazyky FEEL a Javou nebo PMML se používá tabulka na mapování datových typů. Potřeba je zjistit i očekávané datové typy, což lze u Javy pomocí reflexe a u PMML pomocí mining schematu nebo datového slovníku.

Posledním druhem boxeded expressions je seznam. Obsahuje pouze buňky pro hodnoty bez názvů proměnných. To znamená, že se jedná o klasický seznam, jen ve formě boxed expression.

4 Konverzační a choreografické diagramy

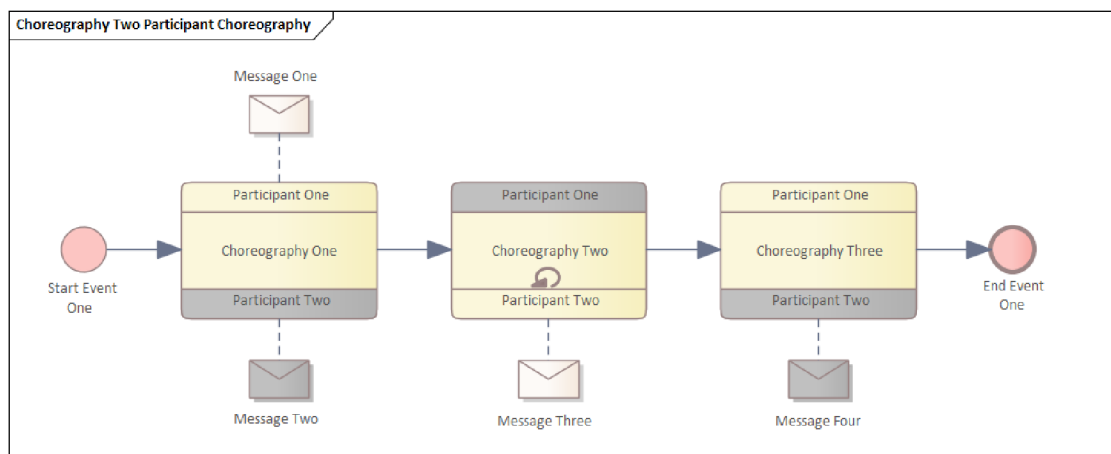
Procesní diagram BPMN nebo diagram DMN, který jde více do hloubky, mohou být doplněny například konverzačním (conversation) diagramem nebo choreografickým (choreography) diagramem, které spolu s procesním a kolaboračním (collaboration) diagramem zahrnuje BPMN 2.0.

Konverzační diagram znázorňuje dle [8] souhrny zpráv, které proudí mezi účastníky dané aplikační oblasti. Jedná se o zobrazení na vysoké úrovni abstrakce. Souhrn zpráv je vyjádřen jako konverzace a jednotlivé zprávy se neznázorňují. Diagram je přehledem o tom, jak spolu účastníci dané domény interagují. Účastníka lze vyjádřit elementem participant a konverzaci účastníků pomocí elementů conversation a conversation link. Tyto základní elementy a schéma konverzačního diagramu znázorňuje obrázek 2.



Obrázek 2 Konverzační diagram [8]

Choreografický diagram situaci dle [9] rozebírá více do hloubky, ve větším detailu. Vzorový diagram znázorňuje obrázek 3. Soustředí se na výměnu konkrétních zpráv, tedy informací mezi účastníky dané aplikační oblasti. Dá se použít k analýze jejich interakcí. Diagramy se podobají procesním diagramům, které jsou vidět například v kapitole 5 na obrázku 1, ale každá aktivita zde představuje interakci mezi dvěma a více účastníky. Na diagramech lze vizualizovat a plánovat, jak má kdo reagovat.



Obrázek 3 Choreografický diagram [9]

Diagram začíná počáteční událostí, která se znázorňuje elementem start event a končí koncovou událostí, která se značí elementem end event. Pro zprávy mezi účastníky interakcí se používá element message a účastníci jsou vyjádření elementem participant. Světle je označen účastník, který na dané aktivitě začíná konverzaci. Tmavě účastník, který obdrží zprávu od světle označeného účastníka, případně mu odpoví. Diagram může obsahovat také element gateway, který určuje rozhodovací bod v procesu. Nejčastěji se používá exclusive gateway. Pokud je pak podmínka splněna, pokračuje proces jedním směrem, pokud ne, pokračuje druhým. Tento element obsahuje diagram na obrázku 10 v kapitole 11.

5 Vývojová prostředí

Tato kapitola se zabývá srovnáním dostupných modelovacích nástrojů pro BPMN a DMN. U výběru modelovacího prostředí hraje roli cenová dostupnost, dostatečné možnosti nástroje pro modelování a vyhodnocování modelů a snadnost ovládání.

5.1 Enterprise architect

Známým softwarem pro systémovou analýzu je Enterprise Architect. Kvůli ceně jeho licence se s ním ale setkáme spíše ve společnostech než u jednotlivců. Standardní licence (pro jednoho uživatele) stojí dle [10] 499 \$ (asi 10 700 Kč) a floating licence (pro předem dohodnutý počet uživatelů) 649 \$ (asi 13 900 Kč). Enterprise Architect umožňuje dle [11] spustit simulaci modelu pro ověření jeho správnosti. Lze v něm také vygenerovat modul DMN v Javě, Java Scriptu, C++ nebo C#. Ten pak uživatel může použít v Enterprise Architect BPSim Execution Engine nebo v Executable StateMachine, což jsou nástroje pro komplexní simulaci a testování modelu, nebo ve vlastním softwaru. DMN je dostupné od verze 14 a doplňující nástroje od verze 15. Univerzita Hradec Králové má k dispozici verzi 13, ve které je již k dispozici BPMN a je v ní namodelován conversation diagram a choreography diagram v kapitole 11.

5.2 Camunda

Známým nástrojem pro tvorbu modelů DMN je také Camunda. Společnost Camunda Services GmbH dle [4] přispěla také k vývoji DMN. Online verzi lze využívat 30 dní zdarma a desktopová aplikace Camunda Modeler je zdarma zcela.

Rozhodovací tabulky dokáže vyhodnotit DMN engine, ke kterému lze přistupovat přes Spring bootovou aplikaci. Vyhodnocovat lze celý model nebo pouze jednotlivá rozhodnutí. Ta by třeba bylo možné vyhodnotit přes post request, dle [12] s url <http://localhost:8080/rest/decision-definition/key/{key}/evaluate>, například pomocí Postmanu. Místo {key} je potřeba zadat hodnotu atributu definition key, který má každé rozhodnutí. Tělo requestu pak obsahuje vstupy daného rozhodnutí ve tvaru: {"variables": {"var1": {"value": 0, "type": "integer"}}}. Název vstupní proměnné, v tomto případě var1, musí být uveden ve formulářovém okně, které se zobrazí po rozkliknutí záhlaví vstupů v rozhodovací tabulce.

Co se týče modelování, lze pomocí tabulky vyjádřit dva druhy boxed expression – rozhodovací tabulku a psaný výraz. Například kontext se už zapisuje do množinových závorek jako {název proměnné: hodnota}.výsledek => {a: 1, b: 2}.b. Tento výraz vypíše díky operátoru tečka hodnotu proměnné b, tedy číslo 2.

5.3 Trisotech a další nástroje

Trisotech nabízí krásný nástroj pro tvorbu analytických diagramů. Grafika tabulek DMN je barevně i proporčně vyvedená přesně podle dokumentace DMN od OMG. Užitečná je záložka „LEARN“, která obsahuje knihovničky funkcí, takže je uživatel nemusí hledat v dokumentaci. Lze si navíc vytvořit i vlastní knihovnu. Free verze softwaru bohužel neumožňuje testování diagramu, pouze validaci, takže si uživatel nemůže vyzkoušet zadávání vstupů a pozorování výstupů.

SAP Signavio má pěkné rozhraní pro analýzu a testování diagramu, verze zdarma je však pouze na 30 dní. Zcela zdarma je například nástroj od BPMN.io, který ale neumožňuje testování modelu. Boxed expressions se v elementech rozhodování zapisují stejným způsobem jako v nástroji Camunda.

5.4 KIE Sandbox

Pokud bychom hledali nástroj, který se snadno ovládá, umožňuje testování diagramu a je volně dostupný, nabízí se KIE (Knowledge Is Everything) Sandbox [13]. Umožňuje uživateli pracovat s BPMN, DMN a PMML. Lze v něm použít všechny elementy DMN i upravit jejich barvu a font. Elementy vnitřní logiky jsou k dispozici také všechny. V záložce „Data Types“ lze vytvořit vlastní jednoduché datové typy nebo strukturované z již předdefinovaných jednoduchých datových typů. Datovým typům lze nastavit i omezení. Můžeme je omezit číselníkem, výrazem nebo rozsahem. Během modelování editor sám vytváří dokumentaci modelu. Pokud si stáhneme rozšířené služby, můžeme model i testovat. Vždy je potřeba aktuální verze. Instrukce nalezneme pod záložkou „Run“. Poté lze zadávat vstupy a zjišťovat či kontrolovat výstupy. Simulace modelu běží na javovském frameworku Quarkus. Model je možné stáhnout ve formátu .dmn a dá se stáhnout i náhled modelu ve formátu .svg. Projekt lze také propojit s GitHubem. K tomu je potřeba se přihlásit účtem GitHubu a vygenerovat token. Přitom by se dle [14] mělo zaháčekovat „repo“

a „gist“. Po těchto krocích se objeví záložka „Sync“, kde je možné provádět pull a push. Do pravého horního rohu se dokonce přidá i tlačítko, díky kterému si můžeme hned prohlédnout či upravit kód modelu z repozitáře ve formátu .dmn pomocí vývojového prostředí Visual Studio Code. Do verze rozšířených služeb 0.30.0 bylo nevýhodou tohoto nástroje to, že na výstupu nedokázal zobrazit struktury. Tomu bylo potřeba uzpůsobovat model. Od této verze je však možné i toto, a i proto je nástroj velice vhodný k použití. Pro začátečníky v DMN nebo začátečníky v práci s tímto softwarem může být nápomocný tutoriál [15].

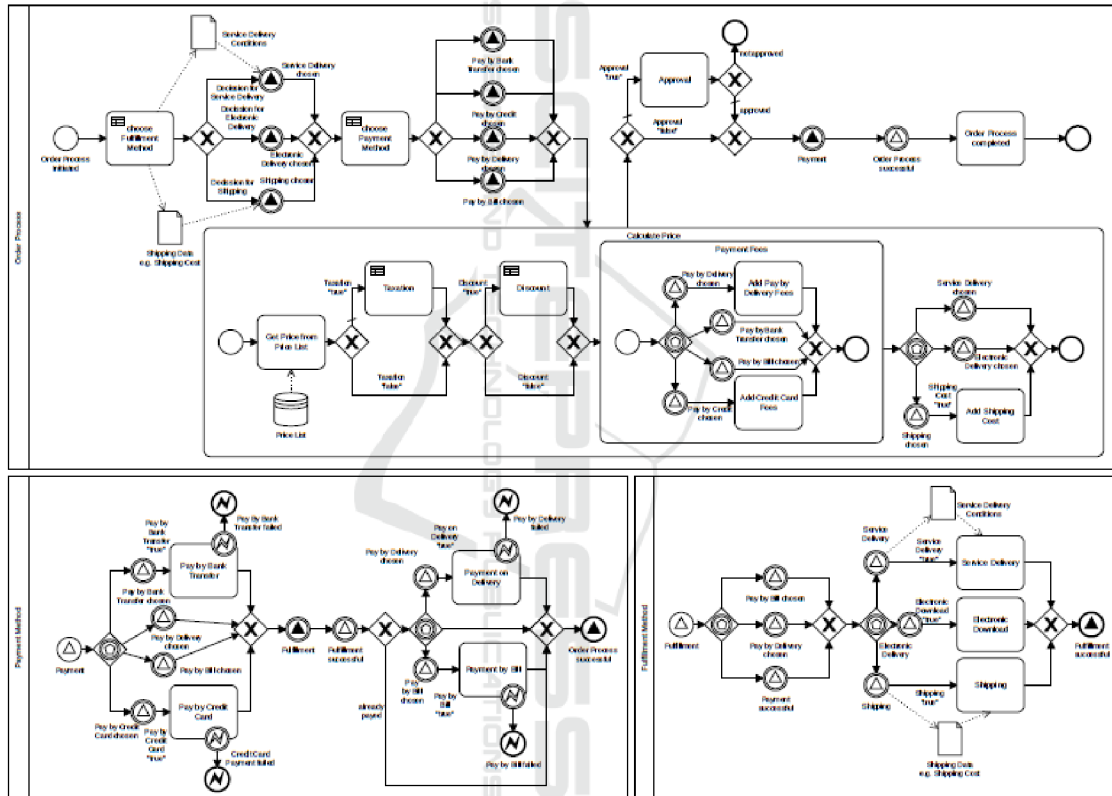
6 Využití DMN

Tato kapitola obsahuje tři příklady využití DMN. Byly vybrány tak, aby se poukázalo především na rozmanitost jeho aplikačních oblastí a také aby byl blíže vysvětlen jeho přínos. DMN lze aplikovat například ve firmách, při řešení krizových situací nebo v medicíně. Všude, kde je potřeba si rozhodování lépe rozvrhnout nebo ho urychlit a přitom zachovat dobré výsledky při rozhodování. Standard DMN lze také rozšiřovat nebo kombinovat s jinými technologiemi.

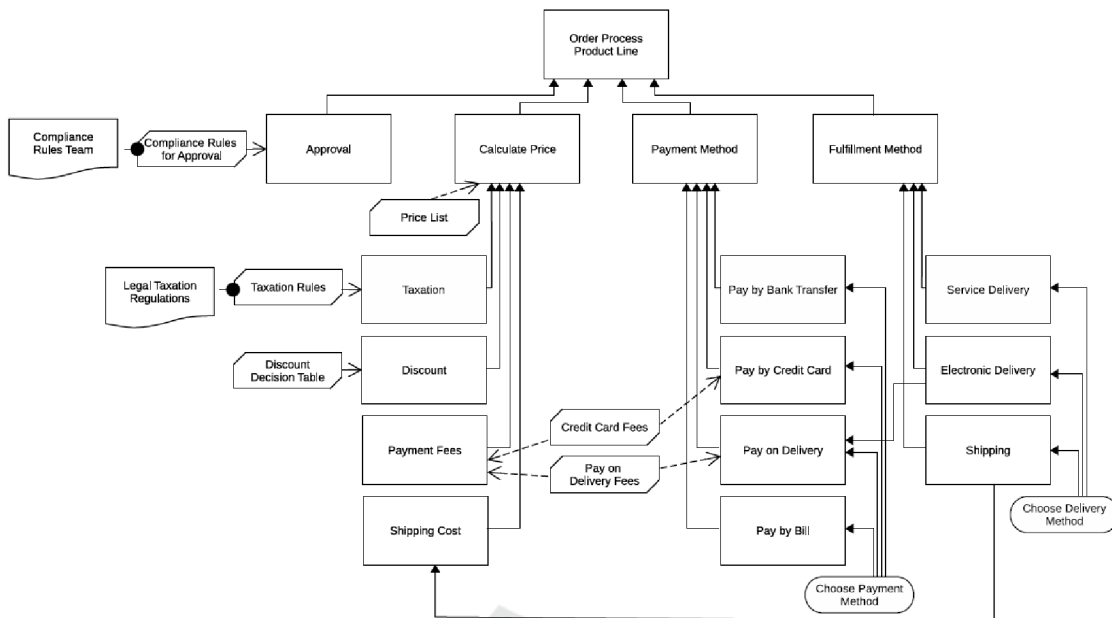
6.1 Modelování variability a podnikový software

[16] řeší, jak co nejefektivněji namodelovat variabilitu mezi znovupoužitelnými komponentami softwaru. Uvažuje nad podnikovými softwary, například systémy pro plánování zdrojů, elektronické obchodování nebo administrativními systémy. Variabilitou jsou myšleny situace, kdy může nastat vícero možností a tok softwaru může pokračovat do různých komponent. Mezi rodičovským elementem a dceřiným elementem mohou být tři typy závislostí. Pokud rodičovský element musí využít dceřiný, jedná se o povinnou závislost, v opačném případě je to dobrovolná závislost. Třetí možností je alternativní závislost, u které je povinné využití alespoň jedné z komponent, z nichž má rodičovský element na výběr.

K modelování variability zde byl použit jazyk BPMN, již zmíněný ve čtvrté kapitole, a ten byl dále rozvinut DMN. V BPMN lze variabilitu řešit větvíci operátory a také připojením nebo oddělením sekvencí procesů. Model BPMN však může být kvůli přílišnému používání větvících operátorů méně jasný. Proto je dobré vytvořit i model DMN. Výhoda modelu BPMN spočívá především v tom, že dobře znázorňuje umístění jednotlivých komponent v toku procesu. Tento přístup byl demonstrován na modelu BPMN procesu objednávky, který je vidět na obrázku 4. K němu vytvořený model DMN je ukázán na obrázku 5. Autoři použili modelovací nástroj Camunda.



Obrázek 4 Model BPMN procesu objednávky [16]



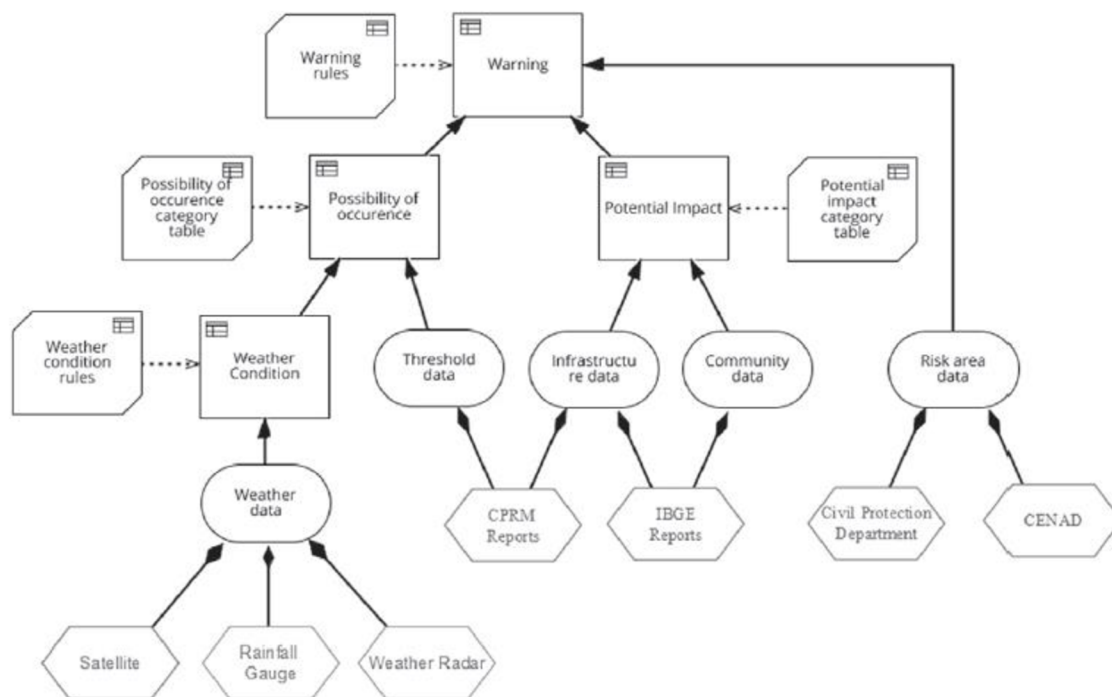
Obrázek 5 DRD procesu objednávky [16]

6.2 Big data a monitorování přírodních katastrof

[17] se zabývá tím, jak při rozhodování využít big data, která se objevila po rozšíření nositelné elektroniky a sociálních sítí. Je například dobré řešit, které informace budou vhodné pro které procesy, jak dodat data ve srozumitelné podobě člověku, jenž rozhoduje, nebo jaký vliv by měla na rozhodování změna dostupnosti dat.

Jako konkrétní případ užití pro rozhodování výzkumníci zvolili monitorování katastrof brazilským národním centrem Cemaden (Brazilian National Center for Monitoring and Early Warning of Natural Disasters). To monitoruje asi 1 000 obcí a je doplněno crowdsourcovými projekty, tedy projekty, které jsou jako výzva nabídnuty veřejnosti, aby bylo dostupných více druhů dat. Centrum spolupracuje i s dalšími institucemi. Cílem práce bylo pomoci Cemadenu zlepšit jeho rozhodování.

Analýza rozhodování i s uvažováním zdrojů informací je zde důležitá i proto, že je na rozhodování v dané chvíli málo času. Data mohou být navíc v různých formátech nebo se například se zakoupením nových senzorů může měnit jejich množství a původ. Pro zahrnutí zdrojů informací do diagramu autoři vyvinuli rozšířením původního jazyka standard oDMN+. Na obrázku 6 je vidět model vytvořený v tomto jazyce. Zdroje dat jsou znázorněny šestiúhelníky. DMN bylo v článku použito na doplnění diagramu BPMN, kde vždy jeden model DMN dopodrobna rozebírá jeden určitý proces modelu BPMN. Model na obrázku 6 konkretizuje proces „Open a warning“. Diagramy byly vytvořeny standardně pomocí rozhovorů s osobami s rozhodovacími pravomocemi.



Obrázek 6 Model oDMN+ o zahájení varování [17]

Analýza pomocí diagramů pomohla celkově urychlit rozhodování. Diagramy slouží jako vodítko a osoby s rozhodovacími pravomocemi pak lépe vědí, jaké aktivity mají provádět a jaké informace mají hledat. Diskuse nad diagramy pomohla například nalézt také alternativní zdroje dat, jako jsou informace ze sociálních sítí nebo oficiálních organizací, např. služeb civilní ochrany nebo policie. Informace ze sociálních sítí by se měly používat pouze pro doplnění, kvůli jejich menší důvěryhodnosti. Protože se data v různých místech liší, vznikl podnět k vytvoření referenční tabulky pro hodnoty různých lokalit, např. množství srážek. Tabulka je následně propojena s parametry rozhodování. Je tedy důležitá flexibilita systému ve vztahu k různým hodnotám a také vzhledem k tomu, že se data v čase mění. Analýza také pomohla určit, která rozhodnutí jsou kritická a která méně. Model také názorně ukazuje, které informace jsou pro jaká rozhodnutí důležité a zda náhodou nechybí.

6.3 DMN a PROMETHEE v medicíně

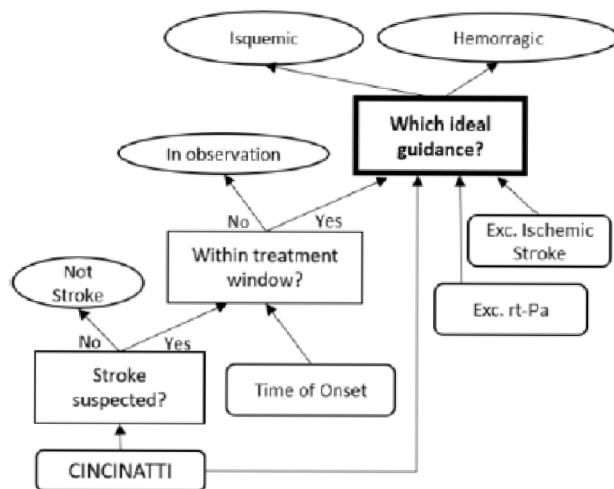
[18] kombinuje modelovací techniku DMN s metodou PROMETHEE II, která se zabývá multikriteriálními analýzami. Aplikační oblastí práce je diagnóza pacientů

s cévní mozkovou příhodou. Výsledný systém může zvýšit asertivitu a efektivitu při rozhodování, na které je málo času.

O mrtvici autoři uvádí, že je v současnosti druhou nejčastější příčinou úmrtí na světě. Cévní mozkové příhody vznikají v důsledku tvorby trombů (krevních sraženin) v cévě. Tím se zablokuje přísun krve do části mozku, což může způsobit postižení, nebo dokonce i smrt. Tyto následky lze minimalizovat včasným zásahem. Diagnóza je velice komplexní a ne každá nemocnice má zdroje a znalosti potřebné k poskytování péče o pacienty s mozkovou příhodou. I proto může být nápomocný rozhodovací model. Lékaři nebo sestry by do modelu vložili informace z dotazníku, který běžně používají při diagnóze této nemoci, a model by provedl rychlé rozhodnutí.

Model je vidět na obrázku 7. V tučně ohraničeném rozhodování je pak použita metoda PROMETHEE II pro vícekriteriální rozhodování. Ta dle [19] spadá pod MCDA (anglická zkratka vícekriteriální rozhodovací analýzy; multiple-criteria decision analysis). Tato metoda umožňuje do rozhodování zahrnout více protichůdných kritérií a vyhodnotit je na základě preferencí rozhodujícího a analytických principů. Například při koupi auta může být zvažována cena, komfort, bezpečnost nebo spotřeba paliva, což jsou kritéria, která se mohou navzájem vylučovat.

[18] používá tři kritéria. První určuje, zda má pacient cévní mozkovou příhodu, druhé, zda je způsobena krevní sraženinou (může být způsobena i prasknutím cévy), a třetí, zda pacient může podstoupit proceduru rt-Pa, jejíž název nese označení proteinu, který rozkládá krevní sraženiny.



Obrázek 7 Model DMN procesu diagnózy mozkové příhody [18]

7 Strategie ve Formuli 1

V této kapitole je rozebrán případ užití, na který bude DMN aplikováno v praktické části práce. Jedná se opět o odlišnou aplikační oblast oproti těm, které byly uvedeny v předchozí kapitole.

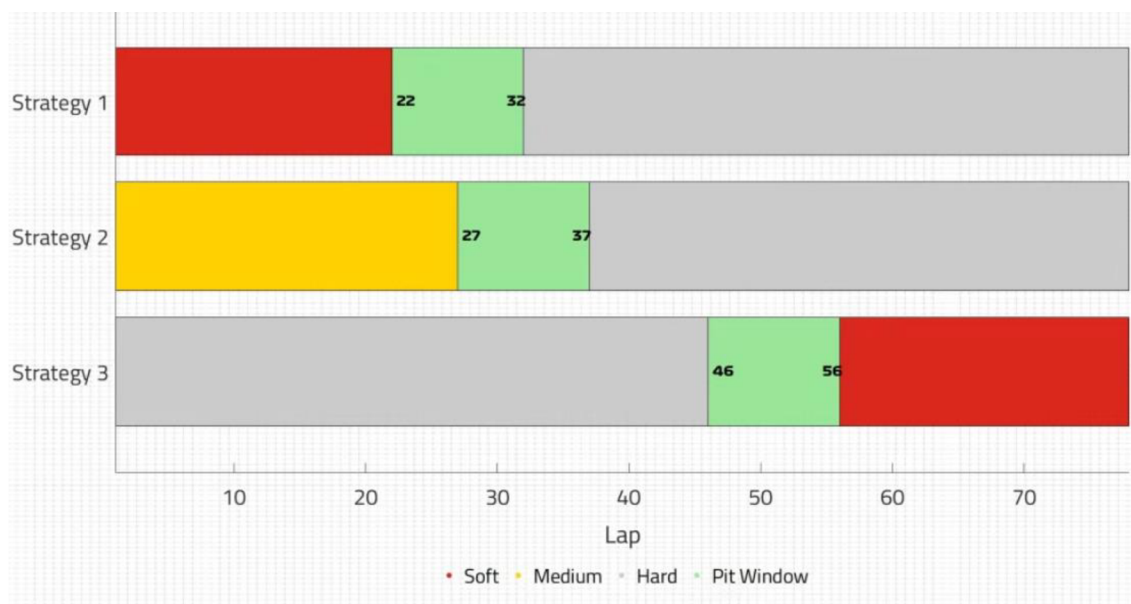
7.1 Tvorba strategie před závodem

Dobrá strategie je důležitou součástí úspěchu v závodě. Dle [20] se začíná připravovat asi šest týdnů před závodem. K jejímu určení inženýři používají i data z minulých závodů na stejné trati. Jestliže se na trati například špatně předjíždí, je dobré provést co nejméně pit stopů, protože by bylo těžké případně pozici vybojovat zpět předjížděním a dává se přednost neriskování. Hodí se také uvažovat o pravděpodobnosti, že vyjede safety car. Pokud je vysoká, může být dobré naplánovat strategii o více zastávkách v boxech, protože během něj nemusí jezdec pit stopem ztratit žádný čas. V Malajsii byla například v roce 2011 pravděpodobnost vyjetí safety caru 20 % (2 závody z 10). Strategie o více zastávkách je podporována i v případě, že pit stop netrvá na dané trati příliš dlouho. Boxová ulička má na různých tratích různou délku. Za velké zdržení se v boxech pokládá ztráta okolo třiceti sekund.

Dále jsou důležitá data od dodavatele pneumatik a data, která inženýři nasbírají během tří volných tréninků na daný závod. Při trénincích se mimo jiné především zjistí, jak dlouho která sada pneumatik na letošním voze vydrží. Z toho lze pak odvodit strategii pro kvalifikaci a v jaké části závodu jezdec použije jakou sadu.

Pro závod se tím vytvoří naplánované strategie, které lze během něj upravit. Neoptimálnější strategie se pracovníě označuje jako plán A a ostatní dalšími písmeny podle optimálnosti. Na grafu na obrázku 8 jsou však označeny jako „Strategy 1, 2, ...“. V případě, že by ani tak tyto strategie nevyhovovaly, vymyslí inženýři novou. Obrázek 8 znázorňuje naplánované strategie na VC Monaka 2022. Jsou zde vidět stinty a intervaly určující, kdy je možné provést pit stop. Stint je anglické označení pro časový úsek mezi zastávkami v boxech. Během jednoho stintu je tedy použita jedna sada pneumatik. Tento pojem bude při modelování také

několikrát využít. Závod v Monaku se jede na 78 kol. Při zastávce v boxech je důležité, aby jezdec pokud možno po pit stopu nevyjel za jezdci, kteří ještě nestavěli, protože by ho brzdili a zdržel by se tím.



Obrázek 8 Očekávané strategie pro VC Monaka 2022 [20]

S každou náhradní strategií je tedy spojena podmínka, kdy je vhodné ji použít místo aktuální strategie. Některá alternativní strategie může být například vhodná, pokud se jezdcí nepodaří kvalifikace, ale zároveň věří, že jeho pozici neodpovídá výkonnost jeho vozu. Jinou příčinou pro odlišnou strategii může být degradace pneumatik, která je vyšší, než se čekalo. Dalším důvodem pro změnu strategie může být i skutečnost, že soupeře není možné předjet na stejných pneumatikách. Tyto podmínky budou v modelu implementovány jiným způsobem než podmínky, po jejichž naplnění je nutné změnit strategii během závodu. Oba druhy řeší trochu něco jiného a u druhého typu není strategie, která se použije, předem známá, ale musí se nově vytvořit.

7.2 Tvorba strategie během závodu

Potřebu tvorby strategie během závodu tedy zapříčiní nějaká neočekávaná situace, například poškození vozu, změna počasí, zastavení závodu, vyjetí safety caru a další. Během fáze závodu, kdy je na trati safety car, jsou auta průměrně asi o 60 %

pomalejší. To znamená, že v této chvíli má jezdec vpředu o 60 % vyšší náskok než v průběhu normální fáze závodu. Náhle náskok například činí 25 s místo 15 s. Pokud by zastávka v boxech trvala méně než 25 s, je možné tento náskok využít k zastávce „zdarma“. Jestliže si je inženýr jistý, že bude mít jeho jezdec po pit stopu výrazně rychlejší pneumatiky než soupeř, může si dovolit ho povolát do boxu i v případě, že je jeho náskok na soupeře o něco nižší než ztráta v boxové uličce, a může ho nechat přijít i o pozici na trati. Do tohoto uvažování je dobré zahrnout také obtížnost předjíždění na dané trati a počet kol zbývajících do konce závodu. Pokud by hrozilo dojetí závodu pod safety carem, nevyplatilo by se vůbec ztrácet pozici.

7.3 Stratégové

Výše zmíněným se dle [21] zabývají stratégové, kteří mají svého šéfstratégu (chief strategist). Ti rozebírají strategii především z dlouhodobého hlediska – připravují ji na závod. Během závodu strategii podle potřeby mění závodní inženýři. Jejich počet je stejný jako počet jezdců. Každý jezdec má jednoho svého. K tomu, aby závodní inženýr mohl ovlivňovat strategii, s jezdce komunikuje a oba si vyměňují informace, které vidí ze svého pohledu. Jezdec svého inženýra například informuje o poškození vozu nebo o změně jeho jízdních vlastností a inženýr informuje jezdce o tom, co dělat proti následkům poškození, co dělat, aby se zlepšily jízdní vlastnosti vozu, kdy zastavit v boxu, a o dalších záležitostech. Kromě toho je úkolem závodního inženýra udržovat jezdce v psychické pohodě, aby podával co nejlepší výkon. Podle toho by s ním měl komunikovat. Mimo svého jezdce a šéfstratégu komunikuje závodní inženýr také s mechaniky, kterým dává pokyny o tom, kdy mají provést pit stop a co je během něj potřeba udělat. Komunikace účastníků rozhodování při tvorbě strategie je znázorněna conversation diagramem v kapitole 11.

7.4 Pneumatiky

Pro doplnění rozebírá tato podkapitola ještě pravidla týkající se pneumatik. Každý jezdec má na víkend k dispozici dvacet sad. Většinou se jedná o osm sad měkké směsi, tři sady střední, dvě sady tvrdé, čtyři sady do přechodných podmínek a tři sady do velkého deště. Pokud se očekává déšť, může se zvýšit počet sad do přechodných podmínek a snížit počet sad pneumatik do sucha. Jestliže se pro závod

předpovídá horké počasí, měli by jezdci během tréninků šetřit se suchými sadami, protože pneumatiky budou během závodu rychleji degradovat a bude potřeba je vícekrát vyměnit. Při deštivé předpovědi jezdci se suchými pneumatikami šetřit nemusí. V případě, že neprší, musí jezdec dle [22] použít alespoň dvě směsi, i kdyby měl k dispozici nějakou, která vydrží celý závod. Čím je pneumatika tvrdší, tím má menší přilnavost, ale zato delší výdrž a naopak. Po každém tréninku se vrací dvě použité sady dodavateli pneumatik – Pirelli. Na kvalifikaci a závod tak jezdci zbyde sedm sad. Jednu sadu měkkých pneumatik si jezdci šetří pro případ, že by postoupili do třetí, poslední části kvalifikace. Pokud se to stane, odevzdává tým dodavateli i tuto sadu. V případě, že je jezdec před třetí částí kvalifikace vyřazen, nechá si ji do závodu. Každý jezdec má své pneumatiky a týmoví kolegové je nemohou sdílet mezi sebou. Také platí pravidlo, že lze vždy použít pouze jednu sadu pneumatik a sady se nesmí míchat.

8 Analýza strategie Formule 1 pomocí DMN

Tato část popisuje vytvořený model DMN. Nejprve jsou rozebrány použité datové typy a poté elementy diagramu a jejich vnitřní logika. Model znázorňuje rozhodování při tvorbě strategie během závodu. Je zpracován na určitém stupni detailu a určitě by se dalo ho dále zpřesňovat. Vstupem jsou již naplánované strategie a vstupy vzniklé z průběhu závodu. Výstupem jsou výsledky rozhodnutí v dílčích situacích nebo nové strategie. Celý diagram je k vidění v příloze č. 1.

8.1 Datové typy

Ze základního typu string byly vytvořeny typy *tyre* a *strategyNameList*. Klíčovým slovem list na konci názvu datového typu jsou od ostatních typů odlišeny seznamy. Datové typy *lap*, *laps*, *percent* a *numberList* vznikly z typu number. Lap vyjadřuje číslo kola v závodě a laps počet kol. Rozsah datového typu lap je omezen na interval <0;78>, protože nejvyšší počet kol má VC Monaka, která se jede na 78 kol. Číslo mimo tento rozsah by nemělo smysl zadávat. Datový typ *percent* je omezen na rozsah <0;100>. V řadě pátý zmíněný typ je použit u pomocných proměnných. Základní datový typ boolean byl použit u vstupů *DriverCrashed*, *DriverChangedWetsForDries* a *SafetyCarDeployed*, což jsou konstanty určující situace, ve kterých je dobré vytvořit novou strategii. V modelu je také využit seznam booleanů *booleanList* u pomocné proměnné v rozhodnutí *Is current lap near planned pit stop?*. Posledním použitým základním datovým typem je date and time duration. Je využit při vyjádření délky pit stopu a ztráty ostatních vozů na vůz stratéga. Tyto údaje jsou v jednotkách sekund. Uživatel je zadá ve tvaru PTXS, kde X je počet sekund.

Další datové typy jsou vytvořeny z právě uvedených datových typů nebo jsou to struktury. Datový typ *tyre* byl odvozen od typu string a je číselníkem omezen na hodnoty „soft“, „medium“, „hard“ a „intermediate“. Datový typ *lapList* je seznam prvků typu lap. Může určovat například seznam kol, ve kterých se provede zastávka v boxech. Ten je využit například u časových rozestupů monopostů.

Nyní budou rozebrány struktury. Tou nejdůležitější je *Strategy*. Obsahuje proměnnou *conditionNumber* typu number určující podmínku, která by měla nastat, aby bylo danou strategií výhodné použít. Dále zahrnuje proměnnou *strategyName*

typu string určující název strategie, proměnnou *PitStopList* typu *PitStop*, určující seznam zastávek v boxu pro danou strategii a proměnnou *tyreList* typu *tyre* určující sady pneumatik pro danou strategii. Již zmíněný *PitStopList* je struktura obsahující dvě proměnné typu number – *fromLap* a *toLap*, ohraničující interval, ve kterém je výhodné na dané strategii provést zastávku v boxech. Struktura *Condition* obsahuje podmínku zmíněnou u struktury *Strategy*. Obsahuje dvě proměnné – *name* typu string, jež určuje název a zároveň obsah podmínky, a *fulfilled* typu boolean určující, zda podmínka nastala, nebo ne. Název podmínky může uživatel zadat libovolně i podle toho, které podmínky v daném závodě mohou nastat. Teorie těchto podmínek je rozebrána v podkapitole 9.1. Poslední strukturou, která byla vytvořena, je *RaceInfo*. Ta obsahuje proměnné určující obecné informace o závodě. Proměnná *wetWeather* je typu boolean a určuje, zda je deštivé počasí, či nikoli. Ostatní proměnné jsou již zmíněného typu laps. Celkový počet kol závodu vyjadřuje proměnná *totalLaps*. Ostatní proměnné určují výdrž směsí pneumatik v počtu kol. Ze struktur byly také vytvořeny a použity seznamy *ConditionList*, *strategyList* a *pitStopList*. Jejich význam je odvoditelný z jejich názvů.

8.2 Elementy

V této podkapitole jsou popsány všechny elementy modelu. Diagram obsahuje třináct vstupů, dvanáct rozhodnutí a čtyři bussiness knowledge modely. Ještě před popisem elementů je vysvětlen způsob propojení mezi rozhodnutími a bussiness knowledge modely. To odůvodňuje přítomnost některých přídavných rozhodnutí.

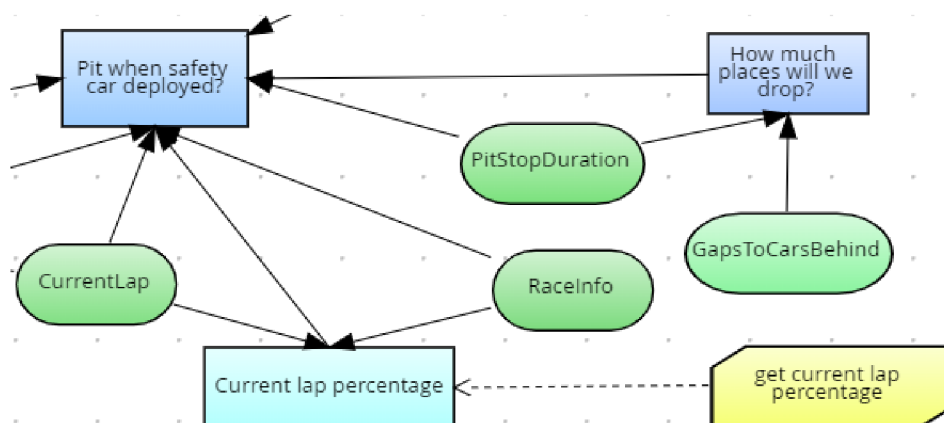
8.2.1 Rozhodnutí a bussiness knowledge modely

Ideální přístup, jak je propojit, je přes jiné rozhodnutí, jehož vnitřní logiku tvoří vyvolání. V rozhodnutích, kde je daný bussiness knowledge model použit je pak struktura vnitřní logiky přehlednější. Není nutné do nich posílat vstupy, které by byly potřeba pouze jako parametry bussiness knowledge modelů a dále by se s nimi v samotném rozhodnutí nepracovalo. Lepší je tedy předat parametry ve zvláštním rozhodnutí s vyvoláním bussiness knowledge modelu. Od rozhodnutí, kde chceme využít služeb modelu, je pak odstíněna a zapouzdřena nejen vnitřní logika modelu,

ale i jeho vyvolání. Toto rozhodnutí přistupuje pouze k výsledné hodnotě, kterou vrátí rozhodnutí s vnitřní logikou vyvolání bussiness knowledge modelu.

Jak bylo již uvedeno v páté kapitole, do verze rozšířených služeb 0.30.0 námi vybrané vývojové prostředí neumělo na výstupu zobrazovat struktury. Výstupy rozhodnutí jsou v rozhraní pro uživatele na rozdíl od výstupů bussiness knowledge modelů vždy zobrazovány, a protože jsou výsledkem tří bussiness knowledge modelů struktury strategií, bylo dřív možné použít rozhodnutí s vyvoláním pouze u rozhodnutí *get current lap percentage*, protože jeho výstupem je hodnota typu *percent*, což není struktura. Po aktualizaci na verzi 0.30.0 bylo tedy možné v průběhu psaní práce model přetvořit a propojit elementy efektivněji, výše uvedeným způsobem.

Rozhodnutí s vyvoláním je v modelu pro svou zvláštní funkci barevně odlišeno, aby bylo na první pohled jasné, že ostatní rozhodnutí, která pracují s výstupy tohoto rozhodnutí, využívají především výsledek vnitřní logiky daného bussiness knowledge modelu, a ne samotného rozhodnutí, jež tento model vyvolává. Konkrétně to ukazuje obrázek 9, který je výsečí našeho diagramu. Další kapitoly popisují jednotlivé elementy.



Obrázek 9 Vyvolání bussiness knowledge modelu

8.2.2 Rozhodnutí *Pit when safety car deployed?*

Jako první bude rozebráno rozhodnutí s názvem *Pit when safety car deployed?*. Má dva klasické vstupy. Vstup *OvertakingDifficulty* – obtížnost předjíždění je základního typu string. *TyreAge* určuje stáří sady pneumatik v počtu kol. Dále jsou vstupem do tohoto rozhodnutí výstupy tří dalších rozhodnutí. Výstupem rozhodnutí *Is current*

lap near planned pit stop? je hodnota proměnné typu boolean a určuje, zda je některá z naplánovaných zastávek blízko aktuálnímu kolu. Rozhodnutí *How much places will we drop?* vrací hodnotu proměnné typu number a určuje, o kolik pozic by se jezdec propadl, kdyby právě zajel do boxů. Z rozhodnutí *Current lap percentage* je také vrácena hodnota proměnné typu number a vyjadřuje odjetou část závodu v procentech.

Vnitřní logiku tohoto rozhodnutí tvoří rozhodovací tabulka. Zásada přístupu je zde nastavena na A, tedy any. To znamená, že výstupy, které budou vyhovovat podmínkám, budou mít stejnou hodnotu a lze použít kterýkoliv z nich. Výstupem rozhodovací tabulky je hodnota proměnné typu string, která určuje, zda při safety caru zajet do boxu, či nikoliv. Pro shrnutí jsou tedy v tabulce zapsány vstupy: *OvertakingDifficulty*, *TyreAge*, *Is current lap near planned pit stop?*, *CurrentLapPercentage* a *How much places will we drop?*.

Tabulka má celkem sedm řádků. V prvním je vyjádřeno, že kdykoliv je plánovaný pit stop blízko současnému kolu, je dobré zastavit. Na druhém řádku se testuje, zda je plánovaný pit stop daleko a jestli se pneumatiky opotřebovávaly pět kol a méně. Pokud toto platí, nevyplatí se zastavovat pro nové pneumatiky, protože opotřebení ještě není tak velké. Třetí řádek říká, že pokud je plánovaná zastávka v boxech daleko, opotřebení pneumatik vyšší než pět kol a zároveň by jezdec neztratil při bezchybné zastávce svou pozici, bylo by dobré zastavit. Tím jezdec snadno získá nové pneumatiky. Pokud by opotřebení pneumatik bylo nižší a zároveň by jezdec neztratil pozici, bylo by nejspíš na individuálním rozhodnutí, zda riskovat nepovedený pit stop, i když jsou pneumatiky skoro nové. Další řádek vrátí hodnotu „yes“, pokud je odjeto 90 % a méně z délky závodu, plánovaná zastávka v boxech je daleko, pneumatiky jsou starší než pět kol, jezdec by ztratil tři a méně míst a na trati se snadno předjíždí. Jezdec bude mít tedy nové pneumatiky a zároveň může vybojovat svou předešlou pozici zpět, pokud by jezdcí za ním nejeli do boxů, protože trať nabízí dobré možnosti k předjíždění. Pokud vstupy odpovídají podmínkám pátého řádku, bylo by lepší zůstat na trati. Testuje se zde to samé, co v předešlém řádku, ale jezdec by přišel o více než tři místa, což by mohlo být riskantní, a už by svou předešlou pozici nemusel vybojovat zpět, i kdyby se na trati dobře předjíždělo. Šestý řádek se oproti předchozím dvěma liší v tom, že by jezdec ztratil jakýkoliv

počet pozic a zároveň by se těžko předjíždělo. Pak by se vyplatilo zůstat na trati i v tomto případě, kdy má jezdec opotřebované pneumatiky, protože aktuální kolo není tak blízko plánovanému pit stopu, kdy by pneumatiky byly už na hranici opotřebovanosti, takže nějakou rychlost v sobě ještě mají a zároveň je pro útočícího soupeře předjetí skoro nemožné kvůli vysoké obtížnosti předjíždění. Poslední řádek řeší případ, kdy je odjeto 90 % závodu a závod se blíží ke konci. Současné kolo tím pádem není blízko žádné plánované zastávce a dále se také testuje, zda je stáří pneumatik větší než pět kol a zda by po zastávce jezdec ztratil pozici. Zde hrozí, že by závod byl už dojet pod safety carem a že se už nebude moci předjíždět a dohánět pit stopem ztracené pozice. Náhled na vnitřní logiku tohoto rozhodnutí se nachází v příloze č. 2.

8.2.3 Rozhodnutí *How much places will we drop?*

Jedním z rozhodnutí, které vstupuje do již popsaného rozhodnutí o zastávce v boxu ve chvíli, kdy na trať vyjede safety car, je, jak bylo již zmíněno, *How much places will we drop?*. Zjišťuje, o kolik pozic se jezdec propadne, zajede-li do boxu. Má dva klasické vstupy: *PitStopDuration* a *GapsToCarsBehind*. Vyjadřují dobu trvání pit stopu včetně projetí boxovou uličkou a ztráty vozů na vůz stratéga. Jak bylo již zmíněno v předchozí podkapitole, jde tu o datový typ *date and time duration*. Vnitřní logiku rozhodnutí tvoří kontext. Obsahuje čtyři kontextová zadání a jedno výsledkové pole. V prvních dvou se zapisují vstupy do proměnných kontextu. Ve třetím se vypočítává pomocná proměnná *counter*. Její datový typ je *numberList*. Pro každou časovou ztrátu ostatních jezdců se testuje, zda je rozdíl mezi ní a délkou pit stopu menší než nula. Tedy zda by se jezdec za jezdce s danou ztrátou propadl, či nikoliv. Pokud je rozdíl menší než nula čili by se propadl, přidá se do *counter* jednička. Pokud ne, přidá se nula. Aby bylo porovnání s nulou možné, je zapsána také v sekundách jako *duration("PT0S")*. Obě porovnávané hodnoty jsou pak stejného datového typu. V dalším kontextovém zadání se do proměnné *places* typu *number* sčítají položky seznamu *counter*. Ve výsledkovém poli se vrací hodnota proměnné *places*, tedy počet vozů, za které by se jezdec propadl. Náhled na vnitřní logiku tohoto rozhodnutí se nachází v příloze č. 3.

8.2.4 Rozhodnutí *Is current lap near planned pit stop?*

Dále bude popsáno druhé rozhodnutí vstupující do rozhodnutí, které bylo popsáno jako první. Jak již bylo zmíněno, má název *Is current lap near planned pit stop?* a zjišťuje, zda není na nejbližší kola naplánována zastávka v boxech. Vstupem do tohoto rozhodnutí je klasický vstup *CurrentLap* typu lap a rozhodnutí *Planned pit stops*, které vrací seznam kol, v blízkosti nichž by bylo dobré provést zastávku v boxech. Vnitřní logiku rozhodnutí tvoří opět kontext. Má tři kontextová zadání a výsledkové pole. V prvních dvou se opět zapisují vstupy do proměnných. Ve třetím se vypočítává seznam hodnot typu boolean. Pro každé kolo ze seznamu se zjišťuje, zda je kolo plánovaného pit stopu ve vzdálenosti pěti kol od současného kola. Pokud ano, zapíše se do seznamu *booleanList* hodnota true, jinak false. Ve výsledkovém poli je testováno, zda seznam *booleanList* obsahuje alespoň jednu hodnotu true. Pokud ano, vrací rozhodnutí true, jinak false. Náhled na vnitřní logiku tohoto rozhodnutí se nachází v příloze č. 4. Do rozhodnutí *Pit when safety car deployed?* vstupuje ještě rozhodnutí *Current lap percentage*, ale to bude popsáno později u bussiness knowledge modelů, protože jeden z nich toto rozhodnutí vyvolává.

8.2.5 Rozhodnutí *Planned pit stops*

Dalším rozhodnutím, které bude rozebráno, je rozhodnutí *Planned pit stops*. Má jeden vstup, kterým je rozhodnutí s vyvoláním bussiness modelu *Actual strategy*. Logiku rozhodnutí popisovaném tímto odstavcem tvoří kontext a účelem rozhodnutí je zobrazit kola, ve kterých jsou naplánovány pit stopy při aktuální strategii. Ty se získají ve druhém kontextovém zadání ze seznamu *PitStopsList* operátorem tečka aplikovaném na vstup rozhodnutí. Zastávky v boxu se neplánují na jedno kolo, ale na rozmezí kol, a tak výsledkové pole vrací průměr krajních bodů intervalů, k němuž je přičtena hodnota 0,5, aby byl výsledek zaokrouhlen na celé kolo. Výsledkové pole vrací seznam kol naplánovaných pit stopů *pitsList*. Náhled na vnitřní logiku tohoto rozhodnutí se nachází v příloze č. 5.

8.2.6 Rozhodnutí *Unexpected situation happened*

Rozhodnutí *Unexpected situation happened*, které, jak již název napovídá, rozhoduje o tom, zda nastala nějaká neobvyklá situace, má tři klasické vstupy – *DriverCrashed*,

SafetyCarDeployed a *DriverChangedWetTyresToDryTyres*. Všechny jsou typu boolean. Vnitřní logiku znovu tvoří kontext. Má tři kontextová zadání a ve všech se zapisují vstupy do proměnných kontextu. Ve výsledkovém poli je testováno, zda je některý z booleanovských vstupů roven true. Pokud ano, vrací výsledkové pole true, pokud ne, vrací false. Náhled na vnitřní logiku tohoto rozhodnutí se nachází v příloze č. 6.

8.2.7 Rozhodnutí *Tyre used in the 1st stint*

Rozhodnutí *Tyre used in the 1st stint* slouží především pro kontrolní výpis první použité sady pneumatik během závodu. Má jeden vstup, kterým je rozhodnutí *Preferred strategies*. Vnitřní logika je jako u většiny ostatních rozhodnutí tvořena kontextem. Obsahuje dvě kontextová zadání. V prvním se zapisuje vstup do proměnné a ve druhém je získána první preferovaná strategie, tedy strategie, která byla použita jako první. Z ní je operátorem tečka přistoupeno k seznamu pneumatik, jež se na této strategii použijí, a vybere se první sada pneumatik. Ta se vrátí ve výsledkovém poli. Vnitřní logiku rozhodnutí lze vidět na obrázku 10.

#	Tyre used in the 1st stint (string)	
1	preferredStrategies (strategyList)	Preferred strategies
2	tyre (string)	preferredStrategies[1].tyresList[1]
	<result>	tyre

Obrázek 10 Rozhodnutí *Tyre used in the 1st stint*

8.2.8 Rozhodnutí *View actual strategy*

Podobně jako předešlé rozhodnutí vypadá rozhodnutí *View actual strategy*. Hodilo se především před aktualizací rozšířených služeb vývojového prostředí na verzi 0.30.0, aby se uživatel nějak dozvěděl, na jakou strategii jezdec právě jede. Po aktualizaci lze zobrazit celou strategii a není potřeba zvlášť zobrazovat název, ale v modelu ho můžeme ponechat. Název strategie tedy rozhodnutí už nezískává

z bussiness modelu, ale z rozhodnutí *Actual strategy*, které bussiness model vyvolá. Náhled na vnitřní logiku tohoto rozhodnutí se nachází na obrázku 11.

#	View actual strategy (string)	
1	actualStrategy (Strategy)	Actual strategy
	<result>	Actual strategy.strategyName

Obrázek 11 Rozhodnutí *View actual strategy*

8.2.9 Rozhodnutí *View preferred strategies*

Rozhodnutím, které využívá jiné rozhodnutí vyvolávací bussiness knowledge model je i toto. Jeho účelem bylo, podobně jako u rozhodnutí *View actual strategy*, zobrazit uživateli názvy strukturovaných prvků výstupního seznamu bussiness knowledge modelu. Názvy získává ze vstupního rozhodnutí *Preferred strategies*. Náhled na vnitřní logiku tohoto rozhodnutí se nachází na obrázku 12. Dále budou popsány bussiness knowledge modely.

#	View preferred strategies (strategyNameList)	
1	PreferredStrategies (strategyList)	Preferred strategies
2	StrategiesNames (strategyNameList)	for element in PreferredStrategies return element.strategyName
	<result>	StrategiesNames

Obrázek 12 Rozhodnutí *View preferred strategies*

8.2.10 Rozhodnutí *Current lap percentage*

Toto rozhodnutí bylo zmíněno již v podkapitole 8.2.1. Vstupuje do rozhodnutí *Pit when safety car deployed?*. Díky němu si můžeme usnadnit přístup k bussiness knowledge modelu. Vyvolání bussiness knowledge modelu, jež tvoří jeho vnitřní logiku, předává parametry *CurrentLap* typu *number* a *RaceInfo* typu *RaceInfo*

bussiness kowledge modelu *get current lap percentage*, které bude blíže rozebráno později. Vnitřní logika rozhodnutí je vidět na obrázku 13.

#	Current lap percentage (percent)	
	get current lap percentage	
1	CurrentLap (number)	CurrentLap
2	RaceInfo.totalLaps (laps)	RaceInfo.totalLaps

Obrázek 13 Rozhodnutí *Current lap percentage*

8.2.11 Bussiness knowledge model *get current lap percentage*

Bussiness kowledge model *get current lap percentage* již byl zmíněn na začátku této kapitoly. Jak bylo naznačeno, jeho výstup je použit v rozhodnutí o pit stopu ve chvíli, kdy na trať vyjede safety car, a určuje, kolik procent ze závodu bylo již odjeto. Parametry funkce modelu, a tedy jeho vstupy, mu jsou předány rozhodnutím *Current lap percentage* a jedná se o *CurrentLap* a *RaceInfo*. V těle funkce se vypočítává procento jako: $CurrentLap / RaceInfo.TotalLaps * 100$. Výstupem je hodnota typu number. Zde prakticky vidíme přístup k položce struktury přes tečkový operátor, když počítáme s proměnnou *TotalLaps*. Vnitřní logika tohoto bussiness knowledge modelu je vidět na obrázku 14.

F	get current lap percentage (percent)
	(CurrentLap, RaceInfo.totalLaps)
	$CurrentLap / RaceInfo.totalLaps * 100$

Obrázek 14 Bussiness knowledge model *get current lap percentage*

8.2.12 Rozhodnutí *Preferred strategies*

V podkapitole 8.2.10 zmíněné rozhodnutí předává parametry bussiness knowledge modelu skrz vyvolání a zobrazuje preferované strategie jako celé struktury. Jeho vnitřní logika je vidět na obrázku 15. Další odstavce popisují bussiness knowledge modely. Jako první je popsán ten, který je napojen na toto rozhodnutí.

#	Preferred strategies (strategyList)	
	choose preferred strategies	
1	RecommendedStrategies (strategyList)	RecommendedStrategies
2	AvailableTyres (tyreList)	AvailableTyres

Obrázek 15 Rozhodnutí *Preferred strategies*

8.2.13 Bussiness knowledge model *choose preferred strategies*

Úkolem bussiness knowledge modelu *choose preferred strategies* je vybrat strategie, na které jezdec má dostupné pneumatiky. Do modelu vstupují dva parametry – *RecommendedStrategies* typu *strategiesList* a *AvailableTyres* typu *tyreList*. Tělo jeho funkce je tvořeno kontextem. Nejprve jsou v kontextových zadáních spočítány počty jednotlivých dostupných směsí pro jezdce. Čtvrté kontextové zadání po jedné zpracovává strategie ze vstupního seznamu. Daná strategie je pro jezdce vhodná, pokud má k dispozici alespoň tolik daných sad pneumatik, kolik jich strategie vyžaduje. To se zjišťuje operátorem [], ve kterém se nachází podmínka určující, která říká, že se má vybrat strategie, ve které je alespoň tolik pneumatik, kolik má jezdec k dispozici. V této podmínce pak ještě operátor dále určuje, kterou směs spočítat. Výsledkové pole vrátí seznam vhodných strategií. Náhled na vnitřní logiku tohoto bussiness knowledge modelu se nachází v příloze č. 7.

8.2.14 Rozhodnutí *Actual strategy*

Toto rozhodnutí zobrazuje celou strategii jako strukturu a předává pomocí vyvolání parametry bussiness knowledge modelu, které bude popsáno později. Vnitřní logiku tohoto rozhodnutí ukazuje obrázek 16.

#	Actual strategy (Strategy)	
	get actual strategy	
1	Unexpected situation happened (boolean)	Unexpected situation happened
2	New strategy (Strategy)	New strategy
3	Prefered strategies (strategyList)	Prefered strategies
4	ConditionsToChangeStrategy (ConditionList)	ConditionsToChangeStrategy

Obrázek 16 Rozhodnutí *Actual strategy*

8.2.15 Bussiness knowledge model *get actual strategy*

Aktuální strategii zjišťuje bussiness knowledge model *get actual strategy*. Má čtyři parametry: jeden jednoduchý – *ConditionsToChangeStrategy* typu *ConditionList* a tři rozhodnutí – *Prefered strategies*, *New strategy* a *UnexpectedSituationHappened*. Tělo funkce modelu tvoří opět kontext. Ten má čtyři kontextová zadání. V nich probíhají výpočty a vytvářejí se pomocné proměnné. Ke zjišťování, zda je jiná strategie vhodnější než aktuální, jsou všechny strategie kromě první zapsány do seznamu *OtherStrategiesList* typu *strategiesList*. To je provedeno pomocí funkce *sublist()*, která z původního seznamu vytvoří nový seznam obsahující prvky z původního seznamu od daného indexu včetně. Ve třetím kontextovém zadání se testují strategie po jedné. Jak již bylo uvedeno, strategie je struktura, jež obsahuje i číslo podmínky, určující, jaká podmínka by musela nastat, aby bylo vhodné danou strategii použít. Podmínka je struktura, která zahrnuje název typu string a proměnnou *fulfilled* typu boolean, která určuje, zda je či není splněna. Pro každou strategii je zjišťováno, zda některá podmínka ze seznamu *ConditionsToChangeStrategy* o indexu čísla

podmínky u strategie nemá svou proměnnou *fulfilled* nastavenou na true. Pokud ano, je daná strategie přidána do seznamu ActualStrategy0 pomocí funkce *append()*, která přijímá jako parametry seznam, na jehož konec bude prvek přidán, a samotný prvek, který přidáme. ActualStrategy0 je v předešlém kontextovém zadání inicializován jako prázdný seznam. Pokud žádná strategie nemá svou podmínku nastavenou na true, vrací toto kontextové zadání první strategii ze strategií, které jezdcí vyhovují i podle jeho dostupných pneumatik, tedy první strategii, již vrátí rozhodnutí *Prefered strategies*. Protože může nastat více podmínek najednou, je proměnná třetího kontextového zadání seznamem. Ať už je v tomto seznamu jeden prvek, nebo více, použije se jako aktuální strategie první prvek tohoto seznamu, který obdržíme v posledním kontextovém zadání. Ve výsledkovém poli je navíc testováno, zda nenastala nějaká neobvyklá situace. Pokud ne, je vrácena strategie ze čtvrtého kontextového zadání. Pokud ano, bude vytvořena nová strategie pomocí rozhodnutí *New strategy*. Náhled na vnitřní logiku tohoto bussiness knowledge modelu se nachází v příloze č. 8.

8.2.16 Rozhodnutí *New strategy*

Poslední rozhodnutí, které zbývá uvést má obdobnou funkci, jako předchozí tři a jeho vnitřní logiku ukazuje obrázek 17. Jeho výstupem je nová strategie, kterou vytvoří následující bussiness knowledge model.

#	New strategy (Strategy)	
	make new strategy	
1	Tyre used in the 1st stint (string)	Tyre used in the 1st stint
2	AvailableTyres (tyreList)	AvailableTyres
3	RaceInfo (RaceInfo)	RaceInfo
4	CurrentLapToMakeStrategy (<Undefined>)	CurrentLapToMakeStrategy

Obrázek 17 Rozhodnutí *New strategy*

8.2.17 Bussiness knowledge model *make new strategy*

Bussiness knowledge model *make new strategy* je posledním a možná i nejsložitějším elementem našeho modelu. Má čtyři parametry: jedno rozhodnutí – *Tyre used in the 1st stint* a tři jednoduché vstupy – *RaceInfo*, *CurrentLapToMakeStrategy* a *AvailableTyres*. Ve druhém kontextovém zadání je pomocí podmínek vytvářen seznam sad pneumatik pro novou strategii. První skupina podmínek testuje, zda bude do konce závodu stačit už jen jedna sada pneumatik. Nejprve je vždy zjištěno, zda prší, nebo je sucho, pak se porovnává trvanlivost odpovídající sady s počtem kol zbývajících do konce závodu. Pokud jedna sada stačit nebude, přejde se k dalším podmínkám, které testují dvě sady. Zde se navíc ověřuje, zda jsou obě dané sady pneumatik pro jezdce dostupné. Pokud by náhodou ani jedna podmínka nevyhovovala, vypsal by se hláška „More than two sets needed.“.

Ve třetím kontextovém zadání je řešeno kolo zastávky v boxech nové strategie, pokud při ní budou použity dvě sady pneumatik. Jestliže by se použila pouze jedna sada, už by se zastavovat nemuselo.

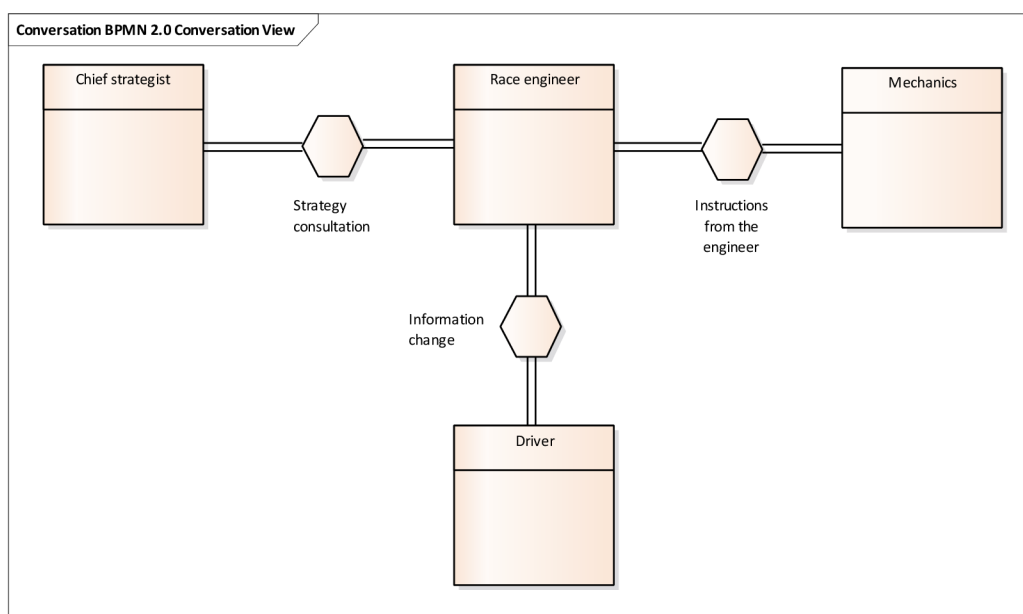
Pro výpočet je zde použita hodnota aktuálního kola, tedy kola, ve kterém nastala nečekaná událost a začala se počítat strategie. Aby se vypočítané kolo pit stopu v průběhu závodu neměnilo, nepoužívá se zde vstup *CurrentLap*, jenž by se měl měnit s každým kolem, ale *CurrentLapToMakeStrategy*, aby kolo, ze kterého se vypočítává pit stop, zůstalo stejné a měnilo se pouze aktuální kolo, které slouží k jiným výpočtům. Je název tohoto parametru je ale v prvním kontextovém zadání zkrácen na *CurrentLap*.

Pokud jsou názvy obou pneumatik stejné, naplánuje se zastávka na polovinu kol zbývajících do konce závodu. Pokud se v každém stintu použijí jiné pneumatiky, určí se kolo pit stopu poměrem zbývajících kol, v němž je trvanlivost první sady vůči trvanlivosti sady druhé. Menší část poměru se přičte k aktuálnímu kolu a dostaneme kolo pit stopu. Takto je určeno pouze jedno kolo a pit stop je možné jako běžně provést v rozmezí několika kol. Proto se v dalším kontextovém vstupu u proměnné *fromLap* odečítá od vypočítaného kola dvojka a u proměnné *toLap* se přičítá. Obě vypočítané hodnoty kol jsou zaokrouhleny na celá čísla. Zde mimochodem

používáme další zanořený kontext a ten je bez výsledkového pole, které není vždy povinné, a zde se to hodí. Kontext bez výsledkového pole využíváme i v posledním kontextovém vstupu, kde skládáme dříve zjištěné proměnné dohromady a vytváříme z nich strategii. Následně jí vrací výsledkové pole kontextu tohoto bussiness kowledge modelu. Náhled na vnitřní logiku tohoto bussiness knowledge modelu se nachází v příloze č. 9 a z důvodu její rozsáhlosti pokračuje v příloze č. 10.

9 Doplnění modelu DMN konverzačním a choreografickým diagramem

Model DMN, který je popsán v předchozí kapitole, lze pro lepší dokreslení situace doplnit konverzačním diagramem, který se nachází na obrázku 18. Znárodnuje účastníky rozhodování a konverzaci, jež mezi nimi při tvorbě strategie probíhá. To je podrobněji rozebráno v podkapitole 9.3. Jak je v ní řečeno, tvorby strategie se účastní šéfstratég, závodní inženýr a jezdec. Jezdcův inženýr navíc komunikuje s mechaniky a dává jim pokyny ohledně zastávek v boxu.

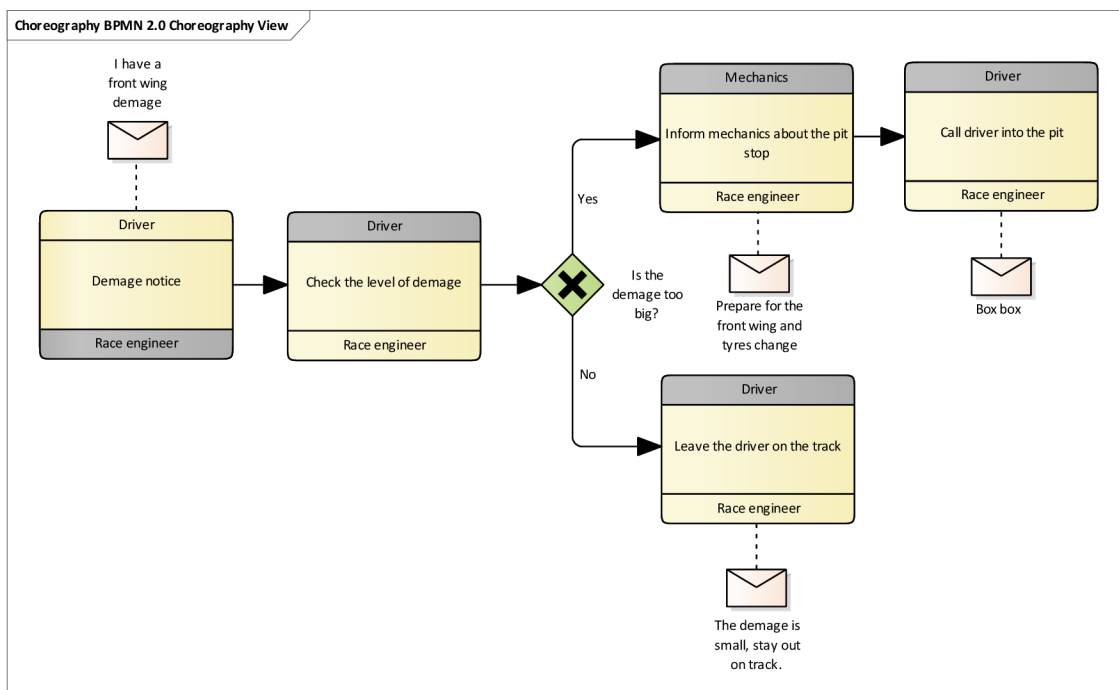


Obrázek 18 Konverzační diagram tvorby závodní strategie

Vytvořený choreografický diagram, který je vidět na obrázku 19, popisuje jednu z mnoha možných komunikací mezi jezdcem, závodním inženýrem a mechaniky při situaci, kdy u jezdce dojde k poškození předního spojleru. Jezdec o poškození informuje svého inženýra a ten na datech zkontroluje, zda je poškození velké a mohlo by vážně ovlivnit rychlost vozu, či nikoliv. Pokud není poškození velké, oznámí jezdcí, že může zůstat na trati. Pokud je velké, informuje mechaniky o tom, že se mají připravit na zastávku jezdce v boxech k výměně pneumatik a předního přítláčného křídla. Pak řekne jezdcí, že může zajet do boxů.

Tento diagram doplňuje v modelu DMN vstup *DriverCrashed*, rozhodnutí *Unexpected situation happened* a bussiness knowledge model *make new strategy*.

Poškození předního spojleru je nečekaná událost, po které závodní inženýr musí přehodnotit strategii a znovu určit, kdy a zda budou provedeny další zastávky v boxech.



Obrázek 19 Choreografický diagram situace při poškození vozu

10 Shrnutí výsledků

Výsledkem práce je model DMN o dvaceti šesti elementech, který je doplněn konverzačním a choreografickým diagramem. Dokáže provádět základní rozhodnutí při tvorbě strategie v závodě.

Do modelu lze zadat zhruba jedenáct vstupů, z nichž některé jsou dále strukturované. Jedná se o dostupné pneumatiky pro jezdce, doporučené strategie pro závod; současné kolo závodu; délku pit stopu včetně projetí boxovou uličkou; náskoky na vozy za jezdce; stáří pneumatik v počtu kol; informace o závodě, které zahrnují počet kol; indikátor určující, zda je mokré, nebo suché počasí, a dále jsou zde zahrnuty trvanlivosti jednotlivých směsí. Dalšími vstupy jsou obtížnost předjíždění; tři události, které mohou nastat během závodu (nehoda, vyjetí safety caru a změna pneumatik ze suchých na mokré), a podmínky, jež se pojí k dané doporučené strategii. Ty si může uživatel sám zvolit a pojmenovat.

Vzniklými výstupy jsou pak výsledky elementů rozhodnutí. Konkrétně se jedná o pneumatiky použité v prvním stintu; názvy preferovaných strategií; název aktuální strategie; výstup určující, zda by se mělo zastavit v boxech, pokud nyní vyjede safety car; kola naplánovaných zastávek v boxu při současné strategii; výstup indikující, zda je aktuální kolo blízko plánovaného pit stopu; výstup, který říká, o kolik míst by se jezdec propadl, kdyby teď zastavil a zároveň by jezdci za ním nezastavili; odjetá část závodu v procentech a výstup upozorňující na to, zda nastala neočekávaná situace.

Názorná ukázka vstupů a výstupů je k vidění na obrázku 20. Vstupy se v použitém vývojovém prostředí zadávají vlevo a výstupy se ukazují napravo. Zde je například vidět na výstupu vytvořená nová strategie. Pokud by byl vstup *CurrentLapToMakeStrategy* vyšší, obsahoval by seznam *PitStopsList* dvě nuly, protože by už nebylo potřeba stavět v boxech.

Doplňující diagramy – konverzační a choreografický ilustrují komunikaci mezi aktéry rozhodování. Jsou jimi šéfstratég, závodní inženýr, mechanici a jezdec. Choreografický rozpracovává situaci a rozhodování při poškození vozu více do detailu.

The image shows a user interface for a data processing tool, divided into two main sections: "Inputs (Row 1)" and "Outputs".

Inputs (Row 1):

- CurrentLapToMakeStrategy:** A text input field containing the value "20".
- CurrentLap:** A dropdown menu with "20" selected.
- PitStopDuration:** A text input field containing "PT25S".
- GapsToCarsBehind:** A list of text input fields, each with a red minus icon to its right. The values are: "PT1S", "PT2S", "PT4S", "PT10S", and "PT30S". A blue plus icon is located above the first item.

Outputs:

The output section is titled "Actual strategy" and displays the following data:

- strategyName:** New strategy
- StintsList-0:** soft
- StintsList-1:** medium
- conditionNumber:** 0
- PitStopsList-fromLap:** 42
- PitStopsList-toLap:** 46

At the bottom of the output section, there is a status message: **Evaluated with success**, accompanied by a checkmark icon.

Obrázek 20 Ukázka vstupů a výstupů

11 Závěry a doporučení

Podářilo se vytvořit model, který dokáže urychlovat a usnadňovat rozhodování při tvorbě strategií ve Formuli 1. Tím bylo demonstrováno další možné využití DMN a ukázáno, jak může vzniknout systém pro tvorbu strategií v motorsportu.

Jedním ze zdrojů, který při tvorbě modelu DMN hodně pomohl, byl [14], který obsahuje jednoduché příklady pro použití základních konstrukcí jazyku FEEL a především interaktivní tutoriál, díky němuž se uživatel naučí používat nástroj KIE Sandbox a získá praktické povědomí o tvorbě modelů DMN. Při psaní složitějších výrazů byla nápomocná hlavně oficiální dokumentace DMN [4] a dá se říci zestručnělá dokumentace [6], které obsahovaly i konkrétní vzorové diagramy (každá jiný), u nichž bylo možné se inspirovat. Zdroje [15–17] pomohly lépe si vytvořit obecnou představu o využití DMN.

Při práci v nástroji KIE Sandbox občas působily problém aktualizace, kvůli kterým některé funkce prostředí přestaly fungovat, jak by měly, například zobrazování textových polí pro strukturované datové typy ve formuláři vstupů. Po další aktualizaci ale zadávání dat strukturovaných datových typů do formuláře pro vstupy opět začalo fungovat a navíc, jak je v práci uvedeno, zobrazuje prostředí po aktualizaci rozšířených služeb na verzi 0.30.0 strukturovaná data i na výstupu.

Model je zpracován na základní úrovni detailu a šlo by ho dále zpřesňovat až do míry, která se podobá skutečnému systému pro tvorbu strategií. Pro reálné použití by se tedy hodil detailnější model, nicméně tento dokáže určité rozhodovací pochody také urychlit a usnadnit. Podobný systém DMN by bylo možné vytvořit také pro tréninky a kvalifikaci. Algoritmy, které byly v modelu vytvořeny, by mohly být využitelné také například v závodních počítačových hrách.

12 Seznam použité literatury

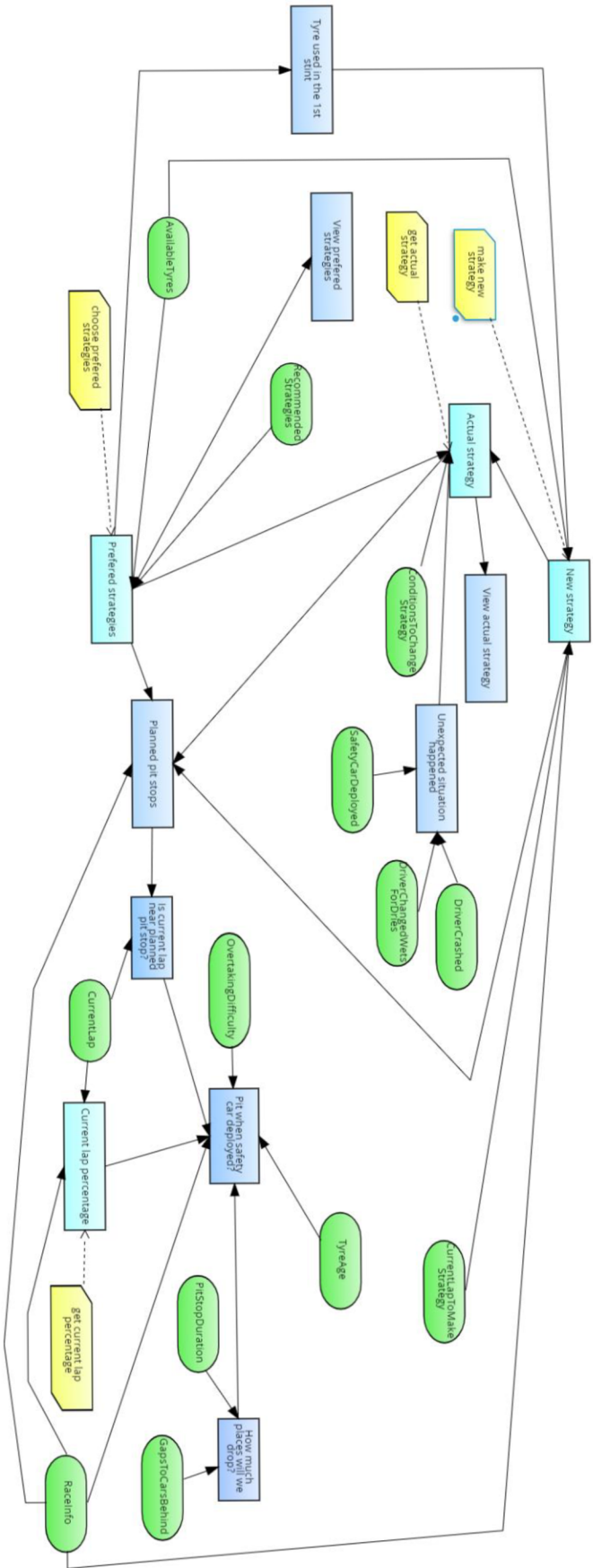
- [1] SILVER, Bruce. What is DMN – Method and Style [online]. 2016, [cit. 2023-04-26]. Dostupné na World Wide Web: <<https://methodandstyle.com/what-is-dmn/>>.
- [2] Mission and Vision [online]. 2023, [cit. 2023-04-26]. Dostupné na World Wide Web: <<https://www.omg.org/about/index.htm>>.
- [3] About the Decision Model and Notation Specification Version 1.4 [online]. 2019, [cit. 2023-03-29]. Dostupné na World Wide Web: <<https://www.omg.org/spec/DMN>>.
- [4] WHITE, Stephen A.; BOCK, Conrad. BPMN 2.0 handbook second edition: methods, concepts, case studies and standards in business process management notation. Future Strategies Inc., 2011.
- [5] IEEE Standard for Floating-Point Arithmetic [online]. 2008, [cit. 2023-04-29]. Dostupné na World Wide Web: <<https://ieeexplore.ieee.org/document/4610935>>.
- [6] Drools Documentation [online]. 2020, [cit. 2023-04-01]. Dostupné na World Wide Web: <https://docs.jboss.org/drools/release/7.41.0.Final/drools-docs/html_single/index.html#dmn-data-types-ref_dmn-models>.
- [7] List expressions | Camunda Platform 8 Docs [online]. 2023, [cit. 2023-08-06]. Dostupné na World Wide Web: <<https://docs.camunda.io/docs/components/modeler/feel/language-guide/feel-list-expressions/>>.
- [8] Conversation Diagrams | Enterprise Architect User Guide [online]. 2023, [cit. 2023-05-23]. Dostupné na World Wide Web: <https://sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/bpmn_2_0_conversation.html>.
- [9] Choreography Diagrams | Enterprise Architect User Guide [online]. 2023, [cit. 2023-05-25]. Dostupné na World Wide Web: <https://sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/bpmn_2_0_choreography.html>.
- [10] Unified Edition of Enterprise Architect | Sparx Systems [online]. 2023, [cit. 2023-04-04]. Dostupné na World Wide Web: <<https://sparxsystems.com/products/ea/editions/unified.html>>.
- [11] Decision Model and Notation (DMN) | Enterprise Architect User Guide [online]. 2023, [cit. 2023-04-04]. Dostupné na World Wide Web: <https://sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/dmn_modeling_and_simulation.html>.

- [12] Evaluate Decision | docs.camunda.org [online]. 2022, [cit. 2023-06-06]. Dostupné na World Wide Web: <<https://docs.camunda.org/manual/7.11/reference/rest/decision-definition/post-evaluate/>>.
- [13] KIE Sandbox [online]. 2019, [cit. 2023-04-06]. Dostupné na World Wide Web: <<https://sandbox.kie.org/#/>>
- [14] CAPONETTO, Guilherme. Deploy your KIE Sandbox to Openshift [online]. 2022, [cit. 2023-04-07]. Dostupné na World Wide Web: <<https://blog.kie.org/2022/01/deploy-your-kie-sandbox-to-openshift.html>>.
- [15] GUILHERME, Carreiro. Learn DMN in 15 minutes | Decision logic [online]. 2022, [cit. 2023-04-09]. Dostupné na World Wide Web: <<https://learn-dmn-in-15-minutes.com/learn/decision-logic>>.
- [16] SPECK, Andreas; JAGENOW, Aljoscha; WINDRICH, Melanie. Decision Model and Notation for Describing Variability in Business Process Product Lines. In: ENASE [online]. 2021, [cit. 2023-04-09]. Dostupné na World Wide Web: <<https://www.scitepress.org/Papers/2021/104972/104972.pdf>>.
- [17] HORITA, Flávio EA, et al. Bridging the gap between decision-making and emerging big data sources: An application of a model-based framework to disaster management in Brazil. Decision Support Systems [online]. 2017, [cit. 2023-04-09]. Dostupné na World Wide Web: <<https://www.sciencedirect.com/science/article/pii/S0167923617300416>>.
- [18] OTERO, P., et al. Decision-Making Support in Stroke Diagnosis Process: An Approach Based on the PROMETHEE Method and Decision Model Notation. In: MEDINFO 2021: One World, One Health—Global Partnership for Digital Innovation: Proceedings of the 18th World Congress on Medical and Health Informatics. IOS Press, 2022. p. 321.
- [19] 16 Different decision making techniques to improve business outcomes | Changeboard [online]. 2019, [cit. 2023-06-04]. Dostupné na World Wide Web: <<https://www.changeboard.com/article-details/16961/16-different-decision-making-techniques-to-improve-business-outcomes/>>.
- [20] MEDLAND, Chris. STRATEGY GUIDE: What are the possible race strategies for the 2022 Monaco Grand Prix? [online]. 2022, [cit. 2023-04-10]. Dostupné na World Wide Web: <<https://www.formula1.com/en/latest/article.strategy-guide-what-are-the-possible-race-strategies-for-the-2022-monaco.5mqY812SIG0kd3pqCAFcxn.html>>.
- [21] OPONG, Rich. What Do F1 Race Engineers Do? (Full Explanation) - FLOW RACERS [online]. 2022, [cit. 2023-05-24]. Dostupné na World Wide Web: <<https://flowracers.com/blog/what-do-f1-race-engineers-do/>>.

[22] ASHER, Richard. F1 tyres: What are the compounds and what do they mean? [online]. 2022, [cit. 2023-04-09]. Dostupné na World Wide Web: <<https://www.autosport.com/f1/news/f1-tyres-what-are-the-compounds-and-what-do-they-mean/10344284/>>.

13 Přílohy

- 1) Diagram DMN
- 2) Rozhodnutí „Pit when safety car deployed“
- 3) Rozhodnutí „How much places will we drop?“
- 4) Rozhodnutí „Is current lap near planned pit stop?“
- 5) Rozhodnutí „Planned pit stops“
- 6) Rozhodnutí „Unexpected situation happened“
- 7) Bussiness knowledge model „choose prefered strategy“
- 8) Bussiness knowledge model „get actual strategy“
- 9) Bussiness knowledge model „make new strategy“ – 1. část
- 10) Bussiness knowledge model „make new strategy“ – 2. část



A	Current lap percentage <i>(percent)</i>	Is current lap near planned pit stop? <i>(boolean)</i>	TyreAge <i>(laps)</i>	How much places will we drop? <i>(number)</i>	OvertakingDifficulty <i>(string)</i>	Pit when safety car deployed? <i>(string)</i>	annotation-1
1	-	true	-	-	-	"yes"	Kolo, kdy vyjel SC je blízko plánovanému
2	-	false	<= 5	-	-	"no"	Pneu je ještě nové, jezdec byl v boxu hned před SC
3	-	false	> 5	0	-	"yes"	Jezdec má opotřebenou pneu a dostatečný náskok na výměnu
4	<= 90	false	> 5	<= 3	"easy"	"yes"	Jezdec nemá dostatečný náskok, ale lehce se předjíždí a nejspíš pozici vybojuje zpět
5	<= 90	false	> 5	> 3	"easy"	"no"	Jezdec by ztratil víc než 3 pozice, pokud by se musel zastavit, i když se předjíždí
6	<= 90	false	> 5	> 0	"hard"	"no"	Jezdec má opotřebenou pneu a nedostatečný náskok, ale je těžké předjíždět a tak zůstane na dráze
7	> 90	false	> 5	> 0	-	"no"	Bude konec závodu, nejspíš bude dojet pod SC

#	How much places will we drop? <i>(days and time duration)</i>	
1	GapsToCarsBehind <i>(dateTimeDurList)</i>	GapsToCarsBehind
2	PitStopDuration <i>(days and time duration)</i>	PitStopDuration
3	counter <i>(numberList)</i>	for element in GapsToCarsBehind return if element - PitStopDuration < duration("PT0S") then 1 else 0
4	places <i>(number)</i>	sum(counter)
	<result>	places

#	Is current lap near planned pit stop? <i>(boolean)</i>	
1	PlannedPitStops <i>(lapList)</i>	Planned pit stops
2	CurrentLap <i>(lap)</i>	CurrentLap
3	booleanList <i>(booleanList)</i>	<pre>for element in PlannedPitStops return if (CurrentLap - 5) <= element and (CurrentLap + 5) >= element then true else false</pre>
	<result>	if list contains(booleanList, true) then true else false

#	Planned pit stops <i>(pitStopList)</i>	
1	ActualStrategy <i>(Strategy)</i>	Actual strategy
2	pitsList <i>(lapList)</i>	for pitStop in ActualStrategy.PitStopsList return (pitStop.fromLap + pitStop.toLap)/2 + 0.5
	<result>	pitsList

#	Unexpected situation happened (boolean)	
1	DriverCrashed (boolean)	DriverCrashed
2	DriverChangedWetsForDries (boolean)	DriverChangedWetsForDries
3	SafetyCarDeployed (boolean)	SafetyCarDeployed
	<result>	if DriverCrashed = true or DriverChangedWetsForDries = true or SafetyCarDeployed = true then true else false

choose preferred strategies <i>(strategyList)</i> (RecommendedStrategies, AvailableTypes)	
F	
1 availableSoftsCount <i>(number)</i>	count(index of(AvailableTypes, "soft"))
2 availableMediumsCount <i>(number)</i>	count(index of(AvailableTypes, "mediums"))
3 availableHardCount <i>(number)</i>	count(index of(AvailableTypes, "hard"))
4 PreferredStrategies <i>(strategyList)</i>	RecommendedStrategiesCount(item.StatsLastItem = "soft") <= availableSoftsCount and count(item.StatsLastItem = "medium") <= availableMediumsCount and count(item.StatsLastItem = "hard") <= availableHardCount
<result>	PreferredStrategies

get actual strategy (Strategy)		
(Unexpected situation happened, ConditionsToChangeStrategy, New strategy, Preferred strategies)		
1	OtherPreferredStrategies (strategyList)	sublist(Preferred strategies, 2)
2	ActualStrategy0 (strategyList)	[]
3	ActualStrategyList (strategyList)	for strategy in OtherPreferredStrategies return if ConditionsToChangeStrategy[strategy.conditionNumber].fulfilled = true then append(ActualStrategy0, strategy) else Preferred strategies[i]
4	ActualStrategy (Strategy)	ActualStrategyList[1]
	<result>	if(Unexpected situation happened = false) then ActualStrategy else New strategy

4	<p>PicStop <i>(picstop)</i></p>	<table border="1"> <tr> <td data-bbox="965 761 1077 940">1</td> <td data-bbox="965 940 1077 1265"> <p>fromLap <i>(number)</i></p> </td> <td data-bbox="965 1265 1077 1960"> <p>if PicstopLap = 0 then 0 else PicstopLap - 2</p> </td> </tr> <tr> <td data-bbox="901 940 965 1265">2</td> <td data-bbox="901 1265 965 1960"> <p>toLap <i>(number)</i></p> </td> <td data-bbox="901 1265 965 1960"> <p>if PicstopLap = 0 then 0 else PicstopLap - 2</p> </td> </tr> <tr> <td colspan="3" data-bbox="901 1960 1077 1986"> <p><result></p> <p>Select expression</p> </td> </tr> </table>	1	<p>fromLap <i>(number)</i></p>	<p>if PicstopLap = 0 then 0 else PicstopLap - 2</p>	2	<p>toLap <i>(number)</i></p>	<p>if PicstopLap = 0 then 0 else PicstopLap - 2</p>	<p><result></p> <p>Select expression</p>								
1	<p>fromLap <i>(number)</i></p>	<p>if PicstopLap = 0 then 0 else PicstopLap - 2</p>															
2	<p>toLap <i>(number)</i></p>	<p>if PicstopLap = 0 then 0 else PicstopLap - 2</p>															
<p><result></p> <p>Select expression</p>																	
5	<p>NewStrategy <i>(Strategy)</i></p>	<table border="1"> <tr> <td data-bbox="837 761 901 940">1</td> <td data-bbox="837 940 901 1265"> <p>conditionNumber <i>(number)</i></p> </td> <td data-bbox="837 1265 901 1960"> <p>0</p> </td> </tr> <tr> <td data-bbox="774 761 837 940">2</td> <td data-bbox="774 940 837 1265"> <p>strategyName <i>(string)</i></p> </td> <td data-bbox="774 1265 837 1960"> <p>if RaceInfo.weather = true then "Rac strategy" else "New strategy"</p> </td> </tr> <tr> <td data-bbox="710 761 774 940">3</td> <td data-bbox="710 940 774 1265"> <p>PicStopList <i>(picstopList)</i></p> </td> <td data-bbox="710 1265 774 1960"> <p>PicStop</p> </td> </tr> <tr> <td data-bbox="646 761 710 940">4</td> <td data-bbox="646 940 710 1265"> <p>StimList <i>(pvelist)</i></p> </td> <td data-bbox="646 1265 710 1960"> <p>TYPESTAKE</p> </td> </tr> <tr> <td colspan="3" data-bbox="614 1960 901 1986"> <p><result></p> <p>Select expression</p> </td> </tr> </table>	1	<p>conditionNumber <i>(number)</i></p>	<p>0</p>	2	<p>strategyName <i>(string)</i></p>	<p>if RaceInfo.weather = true then "Rac strategy" else "New strategy"</p>	3	<p>PicStopList <i>(picstopList)</i></p>	<p>PicStop</p>	4	<p>StimList <i>(pvelist)</i></p>	<p>TYPESTAKE</p>	<p><result></p> <p>Select expression</p>		
1	<p>conditionNumber <i>(number)</i></p>	<p>0</p>															
2	<p>strategyName <i>(string)</i></p>	<p>if RaceInfo.weather = true then "Rac strategy" else "New strategy"</p>															
3	<p>PicStopList <i>(picstopList)</i></p>	<p>PicStop</p>															
4	<p>StimList <i>(pvelist)</i></p>	<p>TYPESTAKE</p>															
<p><result></p> <p>Select expression</p>																	
<p><result></p>		<p>NewStrategy</p>															



Zadání diplomové práce

Autor: Bc. Lukáš Janouch

Studium: I2000711

Studijní program: N1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název diplomové práce: **Analýza strategií Formule 1: DMN modelovací přístup**

Název diplomové práce AJ: Formula 1 Strategies analysis: A DMN Modeling Approach

Cíl, metody, literatura, předpoklady:

analýza strategií rozhodování pro formule 1 a jejich namodelování prostřednictvím DMN

- About the Decision Model and Notation Specification Version 1.4 [online]. 2019. Dostupné na World Wide Web: .
- MEDLAND, Chriss. STRATEGY GUIDE: What are the possible race strategies for the 2022 US Grand Prix? [online]. 2022. Dostupné na World Wide Web:
- Drools Documentation [online]. 2020. Dostupné na World Wide Web:
- ASHER, Richard. F1 tyres: What are the compounds and what do they mean? [online]. 2022. Dostupné na World Wide Web:

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.

Oponent: Ing. Karel Mls, Ph.D.

Datum zadání závěrečné práce: 15.10.2021