Engenharia do Conhecimento 2017/2018

Primeiro Projecto
v1.1
Um sistema de regras de negócio para uma empresa de aluguer de automóveis

Março 2018

1 Introdução **2**
1.1 O conceito de regra de negócio . . . 2
1.2 Objectivo do projecto . . . 2

2 Descrição detalhada do problema **2**
2.1 Pedido de reserva . . . 3
2.2 Afectação de um carro a uma reserva . . . 3
2.3 Concretização do aluguer . . . 4
2.4 Conclusão do aluguer . . . 4
2.5 Devoluções em falta . . . 4

3 Funcionamento do Sistema **4**
3.1 Funcionamento genérico . . . 4
3.2 Tarefas específicas . . . 5
3.2.1 Pedido de reserva . . . 5
3.2.2 Afectação de automóveis . . . 6
3.2.3 Aluguer / início do dia . . . 6
3.2.4 Conclusão do aluguer / devolução do carro . . . 7
3.2.5 Fecho do dia / verificar faltas . . . 7

4 Trabalho a Realizar **7**
4.1 Recursos disponibilizados . . . 8

5 Entrega **8**

6 Avaliação **8**

7 Datas **8**

8 Versões **9**

Bibliografia **9**

1 Introdução

Este projecto consiste na implementação, em CLIPS, de um sistema de regras de negócio que permita facilitar (e regular) o funcionamento de uma empresa do ramo de aluguer de automóveis.

O modelo a representar resulta de um conhecido caso de estudo na área, o da empresa fictícia **EU-Rent** (Hay and Healy [2000]; Frias *et al.* [2003]), sendo o objectivo deste projecto a concretização de uma sua versão simplificada.

1.1 O conceito de regra de negócio

Uma *regra de negócio* é uma afirmação que de algum modo restringe um aspecto concreto de um negócio, quer estabelecendo a sua estrutura quer influenciando o seu comportamento[1].

De acordo com Boyer and Mili [2011], as regras de negócio podem assim dividir-se em duas categorias principais:

estruturais Quando têm o objectivo de caracterizar um conceito do negócio em causa. Por exemplo: *"A informação sobre uma* ***venda*** *deve incluir: o comprador, o produto, a quantidade, o preço e um possível desconto".*

operacionais Quando estabelecem como se deverá comportar o sistema perante determinados eventos. Por exemplo: *"Uma encomenda feita online, para entrega no dia seguinte, só poderá ser aceite se for realizada antes das 15:00."*

Uma característica importante destas regras é a sua natureza *declarativa*, tornando-as facilmente programáveis numa linguagem como o CLIPS.

1.2 Objectivo do projecto

O negócio do aluguer de automóveis envolve várias vertentes: gestão de reservas, gestão de clientes, política de preços e descontos, afectação de automóveis, levantamento e entrega dos automóveis, gestão das várias agências empresa, etc. Todas estas vertentes podem ser caracterizadas por algumas regras de negócio que permitem descrever os conceitos relevantes e estabelecer a forma de funcionamento da empresa em cada uma das situações.

O objectivo deste projecto é concretizar em CLIPS a aplicação de regras de negócio a situações concretas, em alguns dos cenários indicados acima, nomeadamente:

- processamento de um pedido de reserva por parte de um cliente;
- afectação de um carro a um pedido de reserva bem sucedido;
- concretização do aluguer de um automóvel;
- recebimento de um carro (conclusão do aluguer);
- determinação de devoluções em falta;

2 Descrição detalhada do problema

Como referido acima, vamos focar em apenas alguns aspectos do negócio da **EU-Rent**, no que diz respeito a alugueres de curta duração. Mas antes, comecemos por indicar algumas informações genéricas:

[1] Pode consultar [Boyer and Mili, 2011, Cap.1] para uma introdução ao conceito.

- Uma restrição que não depende da empresa é a de que um condutor só pode alugar um carro se tiver um carta de condução válida (considerando a data de validade da carta relativamente ao período de reserva);

- São consideradas as seguintes classes de veículos, por ordem de preço: *mini*, *economico*, *compacto*, *familiar*, *executivo*, *suv*, *luxo*;

- existe uma *lista negra de* clientes, na qual são registados os clientes que incorreram numa das seguintes falhas: entregaram o carro danificado, tiveram algum problema com o pagamento, ou entregaram o carro mais tarde do que o estabelecido.

Comecemos por ver o processamento de um pedido de reserva.

2.1 Pedido de reserva

Um pedido de reserva, é regulado pelas seguintes regras:

- Um pedido de reserva deverá especificar, no mínimo: a identificação do cliente e o período em que o cliente pretende o carro (datas de levantamento e entrega).

- Sendo alugueres de curta duração, o período de cada reserva não poderá exceder 30 dias consecutivos.

- Se um cliente está na *lista negra* o seu pedido de reserva deverá ser rejeitado.

- Se, à data do início do aluguer, o cliente tiver carta de condução há menos de um ano, ou menos de 25 anos de idade, o pedido de reserva deverá ser rejeitado.

- Se um pedido de reserva não especifica um tipo de carro concreto, por omissão é assumido o tipo mais barato (*mini*).

- Só podem ser aceites reservas que possam ser satisfeitas considerando a disponibilidade de carros da agência.

- Um cliente pode ter múltiplas reservas, mas apenas um carro alugado em cada momento.

2.2 Afectação de um carro a uma reserva

Assim que um pedido de reserva é aceite, é necessário afectar um carro de acordo com as seguintes regras:

- Se o pedido de reserva referir um modelo de carro concreto, deverá, havendo disponibilidade, ser afectado um carro desse modelo. Não havendo disponibilidade, deverá ser afectado um carro da mesma classe.

- Se o pedido de reserva não referir nenhum modelo, deverá ser afectado um qualquer carro da classe solicitada.

- (*)[2] Caso existam vários carros possíveis para afectação, deverá ser atribuído o de menos quilometragem.

[2]Os itens assinalados com (*) correspondem a aspectos de implementação mais avançados, requeridos apenas para obter a nota máxima. Mais informação na secção 4.

- (*) Quando não existem carros disponíveis da classe subjacente a um pedido de reserva, deverá ser feito um *upgrade* afectando um carro da classe imediatamente superior. Este *upgrade* poderá ter que ser propagado para outras reservas, de modo a que esta alteração nunca corresponda à afectação de um carro que não pertença à classe imediatamente superior[3].

2.3 Concretização do aluguer

No início de cada dia, deverá ser registado o aluguer de todos os carros que têm reservas a iniciar nesse dia.

Por simplificação, vamos assumir que nenhum cliente se esquece de ir levantar o carro que tem reservado.

2.4 Conclusão do aluguer

Quando o cliente vem devolver o carro à agência devem ser seguidas as seguintes regras:

- Na entrega do carro o cliente tem que efectuar o pagamento.
- Tem que ser verificado se o carro está em condições (não foi danificado).
- Caso existam danos no carro, o cliente é colocado na *lista negra*.

2.5 Devoluções em falta

Ao final de cada dia, a empresa deverá verificar se todas as devoluções programadas para esse dia foram concretizadas pelos clientes:

- Se um carro que deveria ter sido devolvido no dia em causa, não o foi, a empresa deverá contactar o cliente.
- Se um carro já deveria ter sido devolvido na véspera, então, para além de nova tentativa de contacto, o cliente é adicionado à *lista negra*.

3 Funcionamento do Sistema

Cada uma das tarefas referidas na secção anterior deverá corresponder a uma execução independente do sistema. O que há de comum é a informação de suporte (informação sobre clientes, carros, reservas).

3.1 Funcionamento genérico

Não se pretende o desenvolvimento de nenhum interface particular, podendo existir dois modos alternativos de funcionamento:

1. Sem qualquer leitura de dados do utilizador. Os dados relativos ao caso a testar são previamente carregados, sendo seguidamente executado o programa (função) correspondente. Como consequência, a base de factos poderá ser alterada e deverão ser apresentadas mensagens ao utilizador sintetizando o resultado da operação.

[3] Por exemplo, se a reserva solicita um carro da classe *mini*, mas não há disponibilidade nessa classe nem na imediatamente superior (*economico*), então uma das reservas de *economico* passa a *compacto* e a de *mini* passa a *economico*.

2. (*) Os dados existem parcialmente no sistema, sendo o utilizador interrogado sobre eventual informação em falta que seja necessária para executar a tarefa em causa.

Algumas observações:

- Em qualquer momento, deverá ser possível guardar num ficheiro o estado do programa (base de factos). Este estado poderá ser recuperado sempre que desejado.
- Na execução de cada tarefa, deverá sempre ser garantida a coerência da informação registada (na base de factos).
- Poderá sempre assumir a consistência dos dados, nomeadamente quando for utilizada a opção de funcionamento 1. Ou seja, não é preciso, por exemplo, verificar se uma data é válida (no sentido de garantir que os meses estão entre 1 e 12 e o dia é compatível com o mês em causa), ou que a classe veículo é uma das que foi identificada. Poderá sempre assumir que os valores carregados estão correctos.

3.2 Tarefas específicas

Os exemplos seguintes assumem em todos os casos a opção de funcionamento 1 indicada acima.

3.2.1 Pedido de reserva

Os pedidos de reserva podem ser feitos por clientes já registados (para os quais já existe o nome, a data de nascimento, a data da carta de condução e a respectiva data de validade) ou por novos clientes, caso em que os dados referidos têm que ser fornecidos.

Clientes Existentes

Por exemplo, assumindo que temos o cliente:

Tabela 1: Registo de Clientes

IdCliente	Nome	DataNascimento	DataCarta	DataValidadeCarta
2222222	Daniel Silva	3/4/1995	12/3/2016	12/3/2045

E dois pedidos de reserva em seu nome:

Tabela 2: Pedidos de Reserva de Clientes

IdPedido	IdCliente	Data Lev	Data Dev	Classe	Modelo
22	2222222	15/3/2018	18/3/2018	mini	mazda-2
23	2222222	17/3/2018	19/3/2018	-	-

O funcionamento do sistema pode resumir-se a:

```
CLIPS> (clear)
CLIPS> (load ...)
CLIPS> (reset)
CLIPS> (pedido-reserva 22)
Decisão: O pedido de reserva 22 foi aceite
CLIPS> CLIPS> (pedido-reserva 23)
Decisão: O pedido foi rejeitado
Justificação: já existe uma reserva para o mesmo cliente que se sobrepõe ao período indicado.
```

Note que, no caso de um pedido de reserva ser rejeitado, há dois comportamentos possíveis:

1. A tarefa pára assim que seja encontrada a primeira violação de uma regra, sendo apresentada a respectiva justificação.

2. (*) A tarefa verifica todos os critérios, sendo apresentandas no final todas justificações encontradas para a rejeição do pedido.

Novos clientes

No caso de pedidos de reserva feitos por pessoas não registadas como clientes, tem que ser fornecida toda a informação da pessoa:

Tabela 3: Pedidos de Reserva (não clientes)

IdPedido	DataLev	DataDev	Classe	Modelo
33	15/3/2018	18/3/2018	mini	–

Nome	DataNasc	DataCarta	Validade
Filomena Guedes	11/3/1990	17/12/2015	11/3/2040

Neste caso, o sistema deverá processar o pedido e, se o pedido vier a ser aceite, registar um novo cliente com os dados fornecidos:

```
CLIPS> (pedido-reserva 33)
Decisão: O pedido de reserva 33 foi aceite
O novo cliente foi registado com o número 3333333
CLIPS>
```

3.2.2 Afectação de automóveis

Assim que um pedido de reserva é aceite, deverá ser executado processo de afectação de um automóvel a esse pedido, criando-se assim uma reserva.

Imagine que existem os seguintes veículos registados:

Tabela 4: Alguns automóveis disponíveis

Classe	MarcaModelo	KM	Matrícula
mini	mazda-2	9000	00-ZZ-00
mini	smart-for-two	11500	99-XX-99

A execução do sistema poderá resumir-se a:

```
CLIPS> (afecta-carro 33)
Foi afectado o carro 99-XX-99 ao pedido de reserva 33
```

Poderá criar uma função que agregue estes dois passos (pedido de reserva + afectação de automóvel).

3.2.3 Aluguer / início do dia

Deverá definir uma função que processe o início de um dia.

O objectivo desta tarefa é a concretização do aluguer relativo a todas as reservas que se iniciam nesse dia, simulando assim o acto de levantamento do carro por parte do cliente (vamos assumir que todos os clientes levantam os seus carros na data de início da reserva).

O sistema deverá ter sempre presente uma data correspondente ao dia a processar (o *dia de hoje*). Por exemplo:

```
CLIPS> (aluga-carros)
Dia de hoje: 15/3/2018
Foram alugados os carros relativos às reservas:
Reserva 22: 00-ZZ-00
Reserva 33: 99-XX-99
```

3.2.4 Conclusão do aluguer / devolução do carro

A informação necessária relativa à entrega do carro por parte do cliente é a seguinte:

Tabela 5: Informação sobre recepção dos automóveis

IdReserva	**ClientePagou**	**HaDanos**
22	Sim	Não
33	Sim	Sim

Com base nesta informação, poderão ser executados os comandos:

```
CLIPS> (recebe-carro 22)
O carro 00-ZZ-00 foi recebido com sucesso
CLIPS> (recebe-carro 33)
O carro 99-XX-99 foi recebido com danos
O cliente foi adicionado à lista negra.
```

3.2.5 Fecho do dia / verificar faltas

Como última tarefa do dia a empresa deverá verificar se algum cliente não entregou um carro que deveria ter sido entregue, de acordo com as regras definidas acima.

Por exemplo:

```
CLIPS> (verifica-faltas)
Dia de hoje: 18/3/2018
O carro da reserva 44 deveria ter sido entregue hoje: contactar cliente
```

4 Trabalho a Realizar

Para concretizar o funcionamento pretendido é necessário definir em CLIPS o seguinte:

- Um conjunto de *templates* (**deftemplate**) que definam a estrutura dos factos que possibilitem a representação dos dados relevantes para o problema (clientes, carros, reservas, ...). Note que os exemplos acima são meramente ilustrativos. Os templates que definir poderão incluir outra informação para além daquela mencionada nas tabelas anteriores.

- Um conjunto de regras CLIPS que reflitam as regras de negócio enunciadas acima.

- Um conjunto de regras e funções CLIPS que permitam a execução adequada de cada uma das tarefas, garantindo actualização da base de factos de modo a manter a coerência da informação.

Tal como referido acima, a concretização dos itens assinalados com (*) permite a obtenção da nota máxima. A sua não concretização, limita a nota a um máximo de 18 valores.

4.1 Recursos disponibilizados

Durante os próximos dias serão disponibilizados no moodle alguns recursos adicionais, nomeadamente:

- Um conjunto de tabelas (ficheiros CSV) com dados que poderão ter como referência para testar os vossos programas;
- Mais alguns exemplos de execução, ilustrando o tipo de funcionamento pretendido.
- Alguma documentação sobre CLIPS dirigida a aspectos que poderão ser úteis para a implementação do trabalho.

5 Entrega

Os trabalhos deverão ser realizados em grupos de 2 ou 3 elementos. O registo dos grupos deverá ser feito no *moodle*, no link que vier a ser disponibilizado para o efeito.

Será também disponibilizado um link para a submissão do trabalho. A submissão do trabalho deverá consistir de um único ficheiro, compactado em formato `.zip`, com o nome `ec_1718_NN_proj1.zip`, e deverá conter o seguinte:

- o ficheiro `rentacar_NN.clp` com todo o código CLIPS desenvolvido;
- outros ficheiros `.clp` (ou `.bat`), se for o caso;
- um pequeno relatório `relatorio_NN.pdf`, com 5-10 páginas explicando:
  - Como optaram por representar os factos do problema.
  - Que tarefas implementaram.
  - Ilustração de alguns exemplos de execução.

6 Avaliação

Para além do que resultar da avaliação do trabalho de grupo entregue, a nota de projeto individual será ponderada pela realização de mini-teste de aferição, de acordo com as seguintes regras.

A nota final no projecto (NFP) é definida do seguinte modo:

- Sendo NP a nota do projecto, considerando o que foi entregue pelo grupo (é uma nota igual para todos os elementos do grupo);
- Sendo NAf a nota obtida no mini-teste de aferição;
- NFP é igual a:
  - NP, se $75\% <= NAf <= 100\%$
  - $0,75 * NP$, se $50\% <= NAf < 75\%$
  - $0,5 * NP$, se $25\% <= NAf < 50\%$
  - $0,25 * NP$, se $0\% <= NAf < 25\%$
- Não existe nota mínima como condição de acesso ao exame.

7 Datas

- Publicação do projecto: 15/3/2018.

- Constituição de grupos no moodle: até 23/3/2018.
- Submissão do projecto: até 15/4/2018, 23:55, hora de Lisboa.
- Teste de aferição: 18/4/2018, 18:00, em sala a anunciar.

8 Versões

v1.1 (23/3/2018) Correcção de gralhas e da regra de nota mínima na secção 6 (Avaliação) e de um link na Bibliografia.

v1.0 Publicada em 15/3/2018

Bibliografia

Jérôme Boyer and Hafedh Mili. *Agile Business Rule Development*. Springer, 2011.

L. Frias, A. Queralt, and A. Olivé. EU-Rent Car Rentals Specification. Technical report LSI-03-59-R, Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàtics, 2003. DisponÃvel em http://www.cs.upc.edu/~lfrias/research/eurent/R03-59.zip.

David Hay and Kerin Anderson Healy. Defining business rules – what are they really? Technical report, The Business Rules Group, 2000. DisponÃvel em http://businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf.