AIAD - FEUP

Projeto 1
Negotiation-Based Meeting Scheduling

Descrição do problema

O agendamento de reuniões em grandes empresas pode ser uma tarefa complexa. Cada funcionário pertence a um ou mais grupos e cada um destes tem uma série de reuniões a marcar. Cada empregado tem um grau de preferência em relação ao agendamento de cada reunião, podendo mesmo estar indisponível (preferência de grau 0), sendo que pode ter outros compromissos que poderão ou não ser cancelados. Para cada reunião, alguns funcionários têm presença obrigatória, enquanto que outros poderão ser dispensados. Com este cenário em mente, pretende-se desenvolver um SMA com o objetivo de solucionar o problema de agendamento de reuniões numa empresa, em que os agentes representam os seus funcionários. Este agendamento deve ser dinâmico, através da negociação entre os agentes envolvidos.

Variáveis independentes:

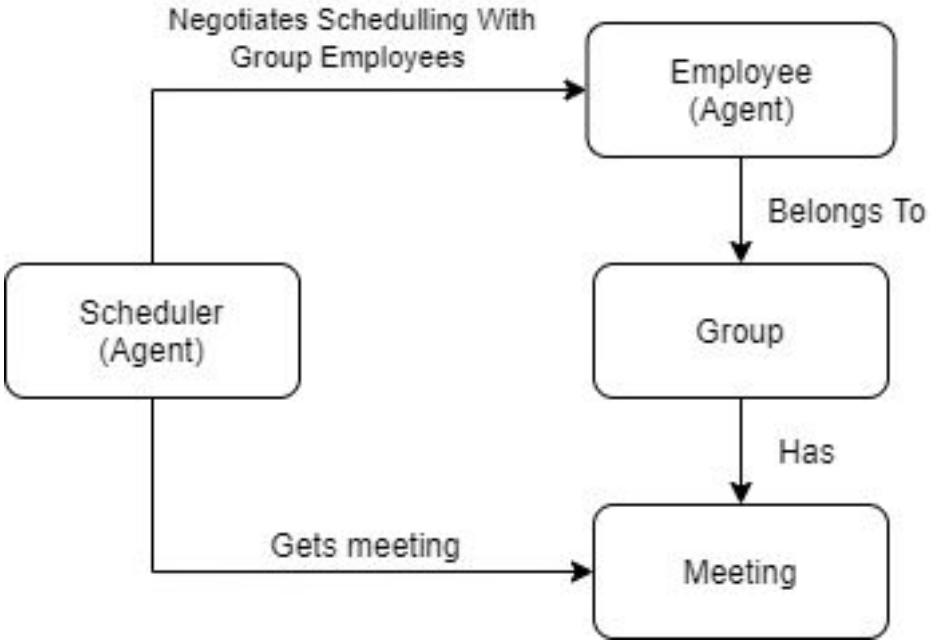
- Preferências de cada funcionário;
- Duração das reuniões;
- Funcionários obrigatórios na reunião;

Variáveis dependentes:

- Horário das reuniões;
- Fator de ocupação da agenda de cada funcionário;

Esquema global

O esquema global à direita pretende mostrar a forma como os agentes envolvidos se encaixam no todo. A empresa tem vários grupos de Employees, cada um com diversas reuniões a marcar. O agente Scheduler, para cada reunião, negoceia com os Employees pertencentes ao grupo da reunião, com o objetivo de encontrar um intervalo de time slots disponíveis entre todos esses Employees que seja igual à duração da reunião.



Interação e protocolos

A solução desenvolvida requer uma negociação em duas partes entre os agentes, a sugestão de *Timeslots* por parte dos *Employees* e a decisão de um *Timeslot* entre o *Scheduler* e os *Employees*. Para este efeito foi necessária a utilização de um protocolo iterativo, sendo o *Contract Net* da FIPA insuficiente para chegar ao resultado pretendido de forma eficiente.

O sistema foi implementado seguindo o protocolo *Iterated Contract Net* da FIPA, estendendo as classes de JADE que seguem o mesmo. O iniciador da interação utiliza uma extensão da classe *ContractNetInitiatorBehaviour*, que implementa o protocolo Contract Net e permite estender para Iterated através do método *newIteration()*. Os respondentes utilizam uma extensão da classe *SSResponderDispatcher*, que em turno gera instâncias de uma extensão da classe *SSIteratedContractNetResponder*, sendo esta a classe que implementa o lado respondedor do protocolo usado.

A comunicação entre agentes é feita através de *ACLMessages* e a maioria dos campos segue o protocolo classes anteriormente referidas. O conteúdo destas mensagens é do tipo *MessageContent* e contém informação relevante ao problema em questão.

Arquiteturas dos agentes, e estratégias utilizadas

Os agentes *Employee* e *Scheduler* extendem a classe *Agent*, dando override ao método *setup* deste no qual é adicionado o *behaviour* designado a cada agente (*EmployeeBehaviour* e *SchedulerBehaviour*, respetivamente, que são *SequentialBehaviour*s com *subbehaviours*).

O primeiro foca-se em receber um pedido de reconhecimento do *Scheduler* e proceder à sua identificação (*EmployeeSendIDBehaviour*), depois, responder ao pedido de sugestão do mesmo para marcação de reuniões e a eventuais propostas, aceitando ou recusando (*EmployeeContractNetResponderBehaviour*).

O último consiste em pedir e receber identificação dos *Employees* (*Request* and *ReceiveEmployeeIDsBehaviour*), criar um *behaviour* para marcar cada uma das *meetings* (*SchedulerContractNetInitiatorBehaviour*) e guardar os resultados das alocações.

Esta arquitetura é baseada no protocolo de negociação entre agentes (*Employees*), tendo um agente central (Scheduler) que coordena a sua comunicação.

Outros mecanismos

DFService:
O DFService do JADE foi usado com o objetivo de encontrar os AIDs dos Employees por parte do Scheduler. No método setup do Employee este regista-se no DFService com o serviço “meeting-scheduling”. Por outro lado, no método setup do Scheduler, este procura no DFService por agentes com o serviço “meeting-scheduling” e guarda os AIDs obtidos numa lista.

Comunicação Inicial:
Para cada reunião, o Scheduler possui informação acerca do grupo à qual esta pertence, nomeadamente uma lista de IDs de Employees que pertencem a esse grupo. Como o objetivo de estabelecer a relação entre este Employee ID e os AIDs guardados durante o setup, é feita uma comunicação inicial entre o Scheduler e cada um dos Employees na qual o Scheduler começa por enviar um REQUEST a todos os Employees (**RequestEmployeeIDsBehaviour**). Os Employees respondem com uma mensagem do tipo INFORM cujo conteúdo é o seu ID único (**EmployeeSendIDBehaviour**). Por outro, o Scheduler espera por esta reposta do Employee, guardando esta informação num HashMap por forma a estabelecer a correspondência entre Employee ID e AID (**ReceiveEmployeeIDsBehaviour**).

Experiências realizadas e análise dos resultados

O sistema foi executado com várias configurações de argumentos (*Groups*, *Meetings* e *Employees*), tendo gerado um ficheiro de resultados aceitável consistentemente.

É de notar que, devido à falta de *backtracking* e incompatibilidade entre horários de *Employees* diferentes, é comum alguns *Meetings* não poderem ser marcados. Também por este motivo, verifica-se que os resultados estão dependentes da ordem em que os meetings são tratados, uma vez que estes são tratados individualmente de acordo com os time slots disponíveis dos Employees e não ocupados por meetings tratados anteriormente.

Conclusões

Tal como referido na análise de resultados, é de notar que estes não são ideais uma vez que cada reunião está a ser tratada de forma independente. Ou seja, não foi implementado um sistema de backtracking que aquando a impossibilidade de marcação de uma reunião, procuraria reagendar uma outra já marcada por forma a libertar time slots. Deste modo, pensamos que esta implementação seria algo a realizar em trabalho futuro.

Foi também notado que o problema escolhido apresentava uma complexidade superior à esperada, nomeadamente devido à necessidade de utilizar uma Iterated Contract Net devido à insuficiência da Contract Net base.

Informação Adicional

Exemplos detalhados de execução

```
Nov 14, 2020 7:29:23 PM logger.MyLogger logInfo
INFO:  FOUND AGENT IN DF SERVICE: AID: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logInfo
INFO:  FOUND AGENT IN DF SERVICE: AID: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logInfo
INFO:  FOUND AGENT IN DF SERVICE: AID: EmployeeAgent3
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENT REQUEST: GET ID
       TO: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENT REQUEST: GET ID
       TO: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENT REQUEST: GET ID
       TO: EmployeeAgent3
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED INFORM: ID:2
       FROM: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED INFORM: ID:1
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED INFORM: ID:3
       FROM: EmployeeAgent3
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  PREPARED CFP: Message Content[State: Request TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: -1 | Meeting Duration: 3 | Acceptance: false];
       TO: EmployeeAgent1; EmployeeAgent2;
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Request TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Request TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: true];
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: true];
       FROM: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  PREPARED ACCEPT_PROPOSAL: Message Content[State: Decide TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       TO: EmployeeAgent1; EmployeeAgent2;
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED RESULT NOTIFICATION: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED RESULT NOTIFICATION: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  PREPARED CFP: Message Content[State: Request TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: -1 | Meeting Duration: 2 | Acceptance: false];
       TO: EmployeeAgent1; EmployeeAgent2;
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Request TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Request TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: true];
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: true];
       FROM: EmployeeAgent2
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  PREPARED ACCEPT_PROPOSAL: Message Content[State: Decide TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       TO: EmployeeAgent1; EmployeeAgent2;
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED RESULT NOTIFICATION: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent1
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED RESULT NOTIFICATION: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       FROM: EmployeeAgent2
```

```
Nov 14, 2020 7:29:23 PM logger.MyLogger logInfo
INFO:  REGISTERED IN DF SERVICE
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED REQUEST: GET ID
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENT INFORM: ID:1
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Request TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: -1 | Meeting Duration: 3 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Request TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Decide TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: true];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED ACCEPT PROPOSAL: Message Content[State: Decide TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RESPONSE TO ACCEPT PROPOSAL: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Request TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: -1 | Meeting Duration: 2 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Request TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Decide TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: true];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED ACCEPT PROPOSAL: Message Content[State: Decide TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RESPONSE TO ACCEPT PROPOSAL: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 1 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
```

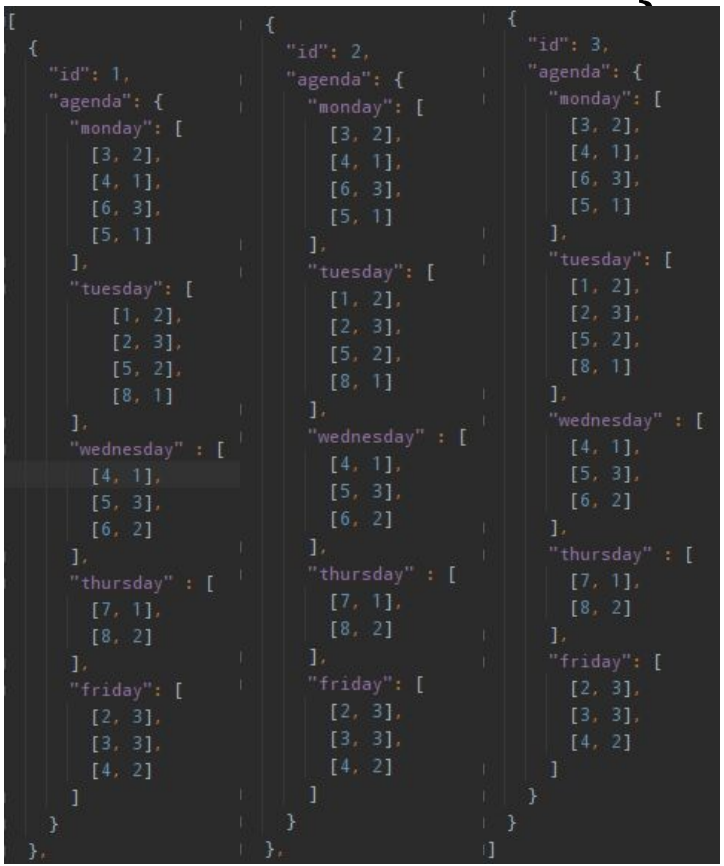
Scheduler (left), Employee 1 (above) and 2 (below) logs

```
Nov 14, 2020 7:29:23 PM logger.MyLogger logInfo
INFO:  REGISTERED IN DF SERVICE
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED REQUEST: GET ID
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENT INFORM: ID:2
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Request TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: -1 | Meeting Duration: 3 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Request TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Decide TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: true];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED ACCEPT PROPOSAL: Message Content[State: Decide TimeSlots | Day: friday | TimeSlot: 2 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RESPONSE TO ACCEPT PROPOSAL: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Request TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: -1 | Meeting Duration: 2 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Request TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED CFP: Message Content[State: Decide TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  SENDING PROPOSAL: Message Content[State: Decide TimeSlots | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: true];
       TO: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RECEIVED ACCEPT PROPOSAL: Message Content[State: Decide TimeSlots | Day: wednesday | TimeSlot: 5 | EmployeeID: -1 | Meeting Duration: -1 | Acceptance: false];
       FROM: SCHEDULER
Nov 14, 2020 7:29:23 PM logger.MyLogger logMessageContent
INFO:  RESPONSE TO ACCEPT PROPOSAL: Message Content[State: NULL | Day:  | TimeSlot: -1 | EmployeeID: 2 | Meeting Duration: -1 | Acceptance: false];
       TO: SCHEDULER
```

Exemplos detalhados de execução (cont.)

```
[
  {
    "id": 1,
    "duration": 3,
    "group": 1,
    "obligatory-employees": [
      1, 2
    ]
  },
  {
    "id": 2,
    "duration": 2,
    "group": 2,
    "obligatory-employees": [
      1, 2
    ]
  }
]
```


```
[
  {
    "id": 1,
    "name": "treasury",
    "employees": [
      1, 2
    ]
  },
  {
    "id": 2,
    "name": "state",
    "employees": [
      1, 2
    ]
  }
]
```

```
[
  {
    "id": 1,
    "agenda": {
      "monday": [
        [3, 2],
        [4, 1],
        [6, 3],
        [5, 1]
      ],
      "tuesday": [
        [1, 2],
        [2, 3],
        [5, 2],
        [8, 1]
      ],
      "wednesday" : [
        [4, 1],
        [5, 3],
        [6, 2]
      ],
      "thursday" : [
        [7, 1],
        [8, 2]
      ],
      "friday": [
        [2, 3],
        [3, 3],
        [4, 2]
      ]
    }
  },
  {
    "id": 2,
    "agenda": {
      "monday": [
        [3, 2],
        [4, 1],
        [6, 3],
        [5, 1]
      ],
      "tuesday": [
        [1, 2],
        [2, 3],
        [5, 2],
        [8, 1]
      ],
      "wednesday" : [
        [4, 1],
        [5, 3],
        [6, 2]
      ],
      "thursday" : [
        [7, 1],
        [8, 2]
      ],
      "friday": [
        [2, 3],
        [3, 3],
        [4, 2]
      ]
    }
  },
  {
    "id": 3,
    "agenda": {
      "monday": [
        [3, 2],
        [4, 1],
        [6, 3],
        [5, 1]
      ],
      "tuesday": [
        [1, 2],
        [2, 3],
        [5, 2],
        [8, 1]
      ],
      "wednesday" : [
        [4, 1],
        [5, 3],
        [6, 2]
      ],
      "thursday" : [
        [7, 1],
        [8, 2]
      ],
      "friday": [
        [2, 3],
        [3, 3],
        [4, 2]
      ]
    }
  }
]
```


Meeting json (top left),
Group json (bottom left),
Employee json (center left),
Results json (right).

The log for Employee 3 was omitted on the previous slide, as it did not take part in the meeting schedule decision.

```
[
  {
    "duration": 3,
    "start_slot": 2,
    "attending_employees": [
      2,
      1
    ],
    "id": 1,
    "end_slot": 4,
    "obligatory_employees": [
      1,
      2
    ],
    "day": "friday",
    "group": {
      "nr_meetings": 1,
      "id": 1,
      "employees": [
        1,
        2
      ]
    }
  },
  {
    "duration": 2,
    "start_slot": 5,
    "attending_employees": [
      1,
      2
    ],
    "id": 2,
    "end_slot": 6,
    "obligatory_employees": [
      1,
      2
    ],
    "day": "wednesday",
    "group": {
      "nr_meetings": 1,
      "id": 2,
      "employees": [
        1,
        2
      ]
    }
  }
]
```

Classes Implementadas

Agentes:

- Scheduler;
- Employee.

Behaviours:

- Scheduler:
  - SchedulerBehaviour;
  - ReceiveEmployeeIDsBehaviour;
  - RequestEmployeeIDsBehaviour;
  - SaveResultsBehaviour;
  - SchedulerContractNetInitiatorBehaviour.
- Employee:
  - EmployeeBehaviour;
  - EmployeeContractNetResponderBehaviour;
  - EmployeeResponderDispatcherBehaviour;
  - EmployeeSendIDBehaviour.

Data:

- Group;
- Macros;
- Meeting;
- MessageContent;
- Timeslot;
- TSPair.

Logger: MyLogger.

Parsers:

- EmployeeParser;
- GroupParser;
- MeetingParser.

Outras Observações

De modo a aprofundar os testes com diversos exemplos, construímos um gerador de problemas (módulo "*problem_generator*" no *working directory* do projeto).

No projeto utilizamos a biblioteca JSONsimple, para além de Jade com o objetivo de ler e escrever JSON files para o. (Download json-simple-1.1.1.jar)

Como executar os programas:
1 - Para fácil utilização deve ser usado o IntelliJ, abrindo o projeto inteiro (com ambos os módulos *meeting_scheduling* e *problem_generator*);
2 - Adicionar as bibliotecas JSONsimple e Jade a ambos os módulos (não à root do projeto);
3 - Adicionar duas *Run Configurations*, uma para o Main de meeting_scheduling com os seguintes *Program arguments*: *<groups_filename> <meetings_filename> <employees_filename>,* e *"Use classpath of module"* com valor *meeting_scheduling*; a segunda para o Main de *problem_generator* com os seguintes *Program arguments*: *<file name> <number of employees> [<number of groups> <number of meetings>]* e *"Use classpath of module"* com valor *problem_generator*. Certifique-se que *"Working directory"* é o diretório pai dos dois módulos.

Gaspar Pinheiro - up201704700
Sofia Lajes - up201704066
Vítor Ventuzelos - up201706403

AIAD - FEUP