

Modelação e Simulação Inteligente

Simulação Discreta por Eventos

Caso de Estudo: Sistema de Fabrico

Paulo Matos

P. PORTO

isep Instituto Superior de
Engenharia da Porto

DEPARTAMENTO DE ENGENHARIA
INFORMÁTICA

© DE/ISEP

1

Disclaimer

Materiais e Slides

- Materiais/slides são adaptados dos slides criados pela prof. Isabel Praça

2023/24

M005

2

© DE/ISEP

2

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico

- Considere-se uma pequena secção fabril responsável pela produção de três tipos de produtos diferentes: canecas, pratos e talheres em alumínio.
- À secção chegam placas retangulares de alumínio, à cadência média de 2 por minuto e com um desvio padrão de 0,5 minutos. As placas entram numa máquina de corte para separação em três novas peças: um prato, uma caneca e um talher.
- Depois da separação das peças, na operação de corte, todas elas passam por uma nova máquina, uma prensa, onde cada uma é moldada conforme o produto a que se destina. A prensa não necessita de mudar de ferramentas entre o fabrico das diferentes peças, mas tem tempos de operação diferentes para cada uma.

© DEI/ISEP

3

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico

- Depois desta primeira operação, as peças passam a seguir percursos diferentes:
 - O prato vai diretamente para o tratamento final.
 - A caneca passa pela operação de colocação da asa, e só depois para o tratamento final.
 - O talher passa para uma operação de separação dos seus elementos, e só depois segue para o tratamento final.

	Maq. Corte	Prensa	MaqAsa	MaqSeparar	TratFinal
Placa Alumínio	N(1.0; 0.3)				
Prato		0.5			N(1.0; 0.3)
Caneca		1.0	N(2.0; 0.5)		N(2.0; 0.5)
Talher		0.5		N(2.5; 0.8)	N(1.5; 0.5)

© DEI/ISEP

4

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico

- O responsável pela secção tem vindo a detectar que, devido a recentes aumentos na cadência de chegada de placas de alumínio, grande parte do tempo passou a ser consumido na espera pelo tratamento final, pelo que pretende um estudo de simulação para fundamentar a possível aquisição de uma segunda dessas máquinas.

© DE/ISEP

5

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico

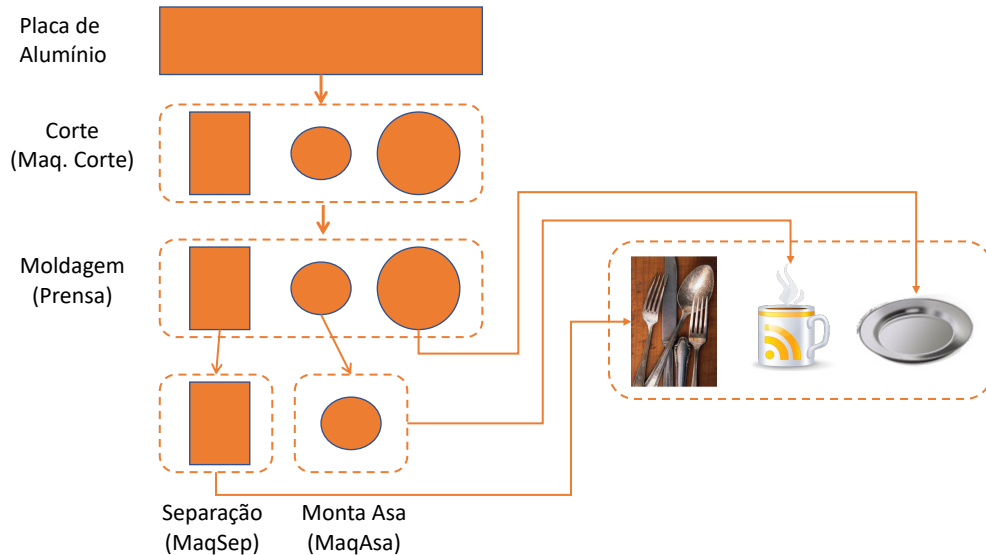
- Identificar as entidades relevantes do sistema
- Identificar e classificar as suas atividades
- Representar a dinâmica do sistema
- Identificar os eventos
- Definir as rotinas associadas aos eventos

© DE/ISEP

6

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico



7

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico

- Entidades criadas dentro do próprio sistema
- Entidades diferentes usam a mesma máquina com tempos de operação diferentes
- Entidades Temporárias
 - Placa de Alumínio
 - Caneca
 - Prato
 - Talher
- Identificar as suas atividades
 - Ciclos de atividades

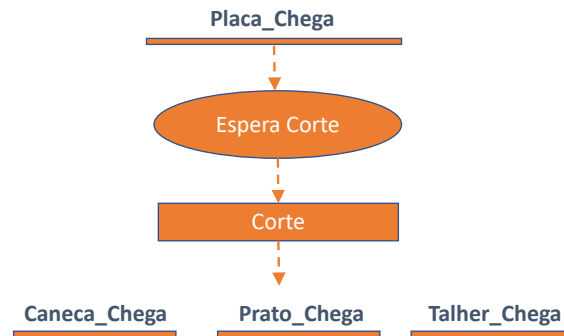
© DEI/ISEP

8

Simulação Discreta por Eventos

Caso de Estudo – Sistema de Fabrico

- Entidades criadas dentro do próprio sistema



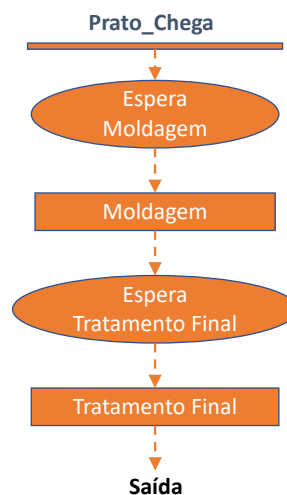
© DE/ISEP

9

Simulação Discreta por Eventos

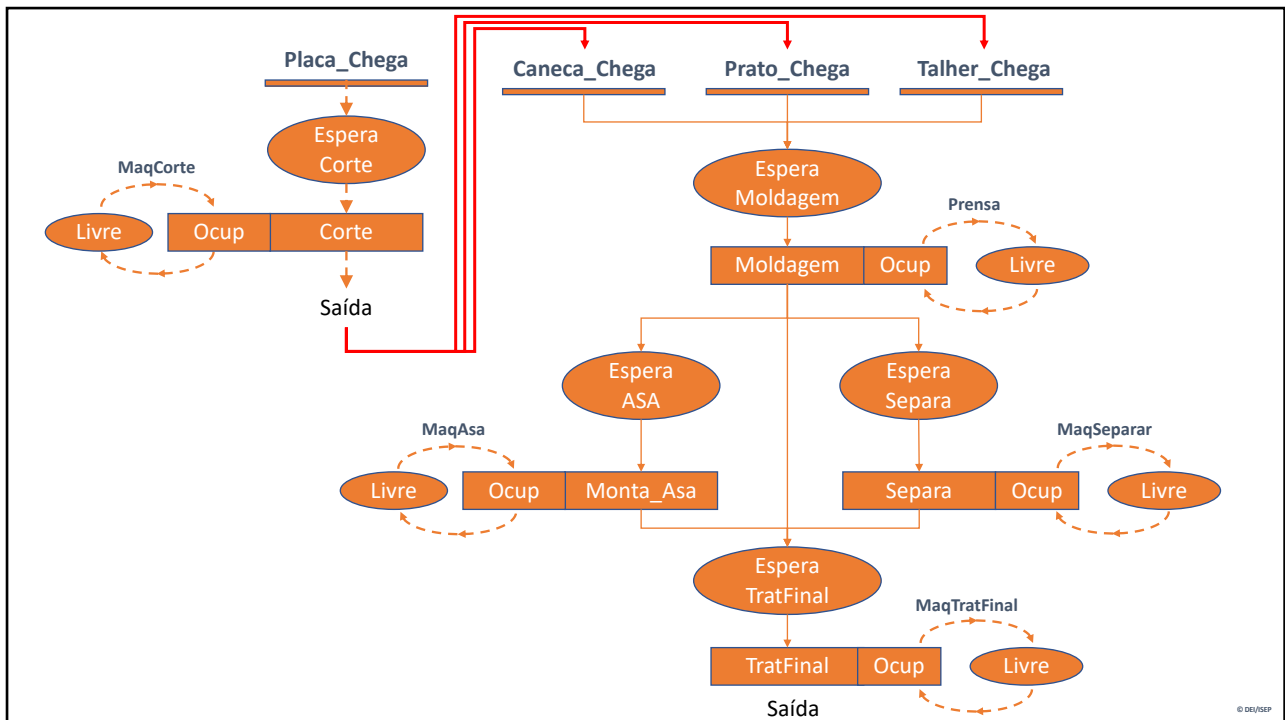
Caso de Estudo – Sistema de Fabrico

- Ciclo de Actividades do Prato



© DE/ISEP

10



11

Simulação Discreta por Eventos

- Identificar entidades
 - Temporárias
 - Placa de Alumínio
 - Caneca
 - Prato
 - Talher
 - Permanentes
 - Máquina de Corte
 - Prensa
 - Máquina de Colocação da Asa
 - Máquina de Separar peças do talher
 - Máquina de Tratamento Final

12

Simulação Discreta

- Abordagem por Eventos
 - Focada nos instantes da simulação em que se preveem transições de estado no sistema
 - A cada transição é associado um evento
 - Criar o modelo é descrever para cada evento relevante do sistema, o conjunto das ações a ele associadas
 - As ações deverão ser executadas sempre que esse evento surja na simulação

© DE/ISEP

13

Simulação Discreta por Eventos

- Eventos e Actividades



© DE/ISEP

14

Simulação Discreta por Eventos

- Eventos
 - Atividades Mortas
 - Evento de Início
 - Mudança de estado
 - Atividades Vivas
 - Evento de Início
 - Mudança de estado
 - Marcação do evento de fim (atividades com duração bem definida)
 - Evento de Fim

© DE/ISEP

15

Simulação Discreta por Eventos

- Eventos
 - PlacaChega
 - InícioCorte
 - FimCorte

 - CanecaChega
 - PratoChega
 - TalherChega

© DE/ISEP

16

Simulação Discreta por Eventos

- Eventos
 - InícioMoldagem
 - FimMoldagem
 - InícioAsa
 - FimAsa
 - InícioSepara
 - FimSepara
 - InícioTrataFinal
 - FimTrataFinal

© DE/ISEP

17

Simulação Discreta por Eventos

```
PlacaChega() {  
    1. retira placa atual do exterior  
    2. coloca placa na fila EsperaCorte  
    // cria próximo placa  
    3. calcula o instante de chegada dessa próxima placa (tnext)  
    4. Escalonar(PlacaChega, tnext) // marca próxima chegada  
  
    // Se só existir a placa atual em espera para o corte  
    // pode marcar o evento de início de corte para o instante atual  
    5. if (PlacasEmEspera.tamanho==1 && MaqCorte.estado==LIVRE)  
        Escalonar(InícioCorte, tsim) // marca início do corte  
}
```

© DE/ISEP

18

Simulação Discreta por Eventos

```
InicioCorte() {  
    1. retira placa da fila EsperaCorte  
    2. coloca placa na fila Corte  
    3. coloca MaqCorte.estado = OCUPADO  
    4. calcula o tempo que durará o Corte (dT)  
  
    // marca fim do corte  
    5. Escalonar(FimCorte, tsim+dT)  
}
```

© DE/ISEP

19

Simulação Discreta por Eventos

```
FimCorte() {  
    1. retira placa da fila Corte  
    2. coloca-a no exterior (...)  
    3. coloca MaqCorte.estado = LIVRE  
  
    // Cria e injecta no sistema as três novas entidades  
    4. Cria uma caneca e coloca-a na fila EsperaMoldagem  
    5. Cria um prato e coloca-o na fila EsperaMoldagem  
    6. Cria um talher e coloca-o na fila EsperaMoldagem  
  
    // Verifica se pode reiniciar Corte (há placas na fila de espera)  
    7. if (PlacasEmEspera.tamanho>0)  
        Escalonar(InicioCorte, tsim) // marca início do corte  
  
    // Verifica se pode iniciar Moldagem (i.e. só existem estas 3 na fila de moldagem)  
    8. if (EsperaMoldagem.tamanho==3 && Prensa.estado==LIVRE)  
        Escalonar (InicioMoldagem, tsim) // marca início moldagem  
}
```

© DE/ISEP

20

Simulação Discreta por Eventos

```
InicioMoldagem(){  
    1. retira uma peça da fila EsperaMoldagem  
    2. coloca a peça na fila Moldagem  
    3. coloca Prensa.estado = OCUPADO  
    4. calcula o tempo que durará a Moldagem de acordo com  
       o tipo de peça (dTi)  
  
    // marca fim da moldagem  
    5. Escalonar(FimMoldagem, tsim+dTi)  
}
```

© DE/SEP

21

```
FimMoldagem() {  
    1. retira peça da fila Moldagem  
    // De acordo com o tipo de peça, coloca-a na próxima espera  
    2. if (peça.tipo == "caneca") coloca peça na fila EsperaAsa  
    3.   else if (peça.tipo == "prato") coloca peça na fila EsperaTratFinal  
    4.     else if (peça.tipo == "talher") coloca peça na fila EsperaSepara  
    5. coloca Prensa.estado = LIVRE  
    // Verifica se pode reiniciar Moldagem  
    6. if (EsperaMoldagem.tamanho>0) Escalonar(InicioMoldagem, tsim)  
    // De acordo com o tipo de peça tenta iniciar próxima actividade  
    // Só inicia se for a única peça na fila de espera (por causa de sincronismo, slide 29)  
    7. if (peça.tipo=="caneca")  
        if (EsperaAsa.tamanho==1 && MaqAsa.estado==LIVRE)  
            Escalonar (InicioAsa, tsim) // Início da Colocação de Asa  
    8. else if (peça.tipo=="prato" )  
        if (EsperaTratFinal.tamanho==1 && MaqTratFinal.estado==LIVRE))  
            Escalonar InicioTratFinal, tsim) // Início Tratamento Final  
    9. else if (peça.tipo=="talher" )  
        if (EsperaSepara.tamanho==1 && MaqSepara.estado==LIVRE))  
            Escalonar(InicioSepara, tsim) // Início da Separação de Talher  
}
```

© DE/SEP

22

Simulação Discreta por Eventos

```
InicioAsa(){  
    1. retira uma caneca da fila EsperaAsa  
    2. coloca a peça na fila MontaAsa  
    3. coloca MaqAsa.estado = OCUPADO  
    4. calcula o tempo que durará a montagem da asa (dT)  
  
    // Marca fim da montagem da asa  
    5. Escalonar(FimAsa, tsim+dT)  
}
```

© DE/ISEP

23

Simulação Discreta por Eventos

```
FimAsa(){  
    1. retira caneca da fila MontaAsa  
    2. coloca-a na fila EsperaTratFinal  
    3. coloca MaqAsa.estado = LIVRE  
  
    // Verifica se pode reiniciar montagem de Asa  
    4. if (EsperaAsa.tamanho>0)  
        Escalonar(InicioAsa, tsim)  
  
    // Verifica se pode iniciar Tratamento Final (só inicia se fila  
    // tiver tamanho 1, pois se tamanho for >1 pode já estar outro  
    // InícioTratFinal escalonado (problema de sincronismo)  
    5. if (EsperaTratFinal.tamanho==1 && MaqTratFinal.estado==LIVRE)  
        Escalonar(InicioTratFinal, tsim)  
}
```

© DE/ISEP

24

Simulação Discreta por Eventos

```
InicioSepara() {  
    1. retira um talher da fila EsperaSepara  
    2. coloca a peça na fila Separa  
    3. coloca MaqSepara.estado = OCUPADO  
    4. calcula o tempo que durará a separação do talher (dT)  
  
    // marca fim da separação do talher  
    5. Escalonar(FimSepara, tsim+dT)  
}
```

© DE/SEP

25

Simulação Discreta por Eventos

```
FimSepara() {  
    1. retira talher da fila Separa  
    2. coloca-o na fila EsperaTratFinal  
    3. coloca MaqSepara.estado = LIVRE  
  
    // verifica se pode reiniciar separação de talher  
    4. if (EsperaSepara.tamanho>0)  
        Escalonar(InicioSepara, tsim)  
  
    // verifica se pode iniciar Tratamento Final (i.e., só há este talher  
    // à espera de Tratamento Final)  
    5. if (EsperaTratFinal.tamanho==1 && MaqTratFinal.estado==LIVRE)  
        Escalonar(InicioTratFinal, tsim)  
}
```

© DE/SEP

26

Simulação Discreta por Eventos

```
InicioTratFinal(){  
    1. retira uma peça da fila EsperaTratFinal  
    2. coloca a peça na fila TratFinal  
    3. coloca MaqTratFinal.estado = OCUPADO  
    4. calcula o tempo que durará o tratamento final,  
       de acordo com o tipo de peça (dTi)  
  
    // marca fim do tratamento final  
    5. Escalonar(FimTratFinal, tsim+dTi)  
}
```

© DEI/SEP

27

Simulação Discreta por Eventos

```
FimTratFinal() {  
    1. retira peça da fila TratFinal  
    2. coloca-a no exterior (elimina ...)  
    3. coloca MaqTratFinal.estado = LIVRE  
  
    // verifica se pode reiniciar o tratamento noutra peça  
    // i.e., existem peças na fila de espera de tratamento final  
    4. if (EsperaTratFinal.tamanho > 0 )  
        Escalonar(Início TratFinal, tsim)  
}
```

© DEI/SEP

28

Simulação Discreta por Eventos

- Problemas de Sincronismo
 - Dois eventos escalonados com o mesmo instante de ocorrência geram conflito
 - Qual deles executar primeiro?

© DE/ISEP

29

Simulação Discreta por Eventos

Problemas de Sincronismo - Exemplo

```
InicioAsa() {  
    1. retira uma caneca da fila EsperAsa  
    2. coloca a peça na fila MontaAsa  
    3. coloca MaqAsa.estado = OCUPADO  
    4. calcula o tempo que durará a montagem da asa (dT)  
  
    // marca fim da montagem da asa  
    5. Escalonar(FimAsa, sim+dT)  
}
```

© DE/ISEP

30

Simulação Discreta por Eventos

- Problemas de Sincronismo - Exemplo

```
FimAsa() {  
  1. retira caneca da fila MontaAsa  
  2. coloca-a na fila EsperaTratFinal  
  3. coloca MaqAsa.estado = LIVRE  
  
  // Verifica se pode reiniciar montagem de Asa  
  4. if (EsperaAsa.tamanho>0)  
    a. retira uma caneca da fila EsperaAsa  
    b. coloca a peça na fila MontaAsa  
    c. coloca MaqAsa.estado = OCUPADO  
    d. calcula o tempo que durará a montagem da asa (dT)  
    e. Escalonar(FimAsa, tsim+dT) // marca fim da montagem da asa  
  
  // Verifica se pode iniciar Tratamento Final  
  5. if (EsperaTratFinal.tamanho==1 && MaqTratFinal.estado==LIVRE)  
    a. retira uma caneca da fila EsperaTratFinal  
    b. coloca a peça na fila TratFinal  
    c. coloca MaqTratFinal.estado = OCUPADO  
    d. calcula o tempo que durará o tratamento final à caneca (dT)  
    e. Escalonar(FimTratFinal, tsim+dT) // marca fim do tratamento final  
}
```

© DEI/ISEP

31

Simulação Discreta por Eventos

- Problemas de Sincronismo
 - Incluir código das rotinas associadas a eventos de início de actividade em rotinas de eventos de fim
 - Usar rotinas separadas, para eventos de início e de fim, mas “ignorar” o escalonamento de eventos de início de actividades vivas (fazer somente a chamada à rotina)
 - Atribuir prioridade a cada evento, que será usada na função de escalonamento para decidir a ordem de execução de eventos simultâneos
 - Atribuir maior prioridade a eventos de fim de actividades

© DEI/ISEP

32

Modelação e Simulação Inteligente

Simulação Discreta por Eventos

Caso de Estudo: Sistema de Fabrico

Paulo Matos

P. PORTO

isep Instituto Superior de
Engenharia da Porto

**DEPARTAMENTO DE ENGENHARIA
INFORMÁTICA**

© DE/ISEP