

Engenharia de Sistemas

SysML - Modelagem: Atividades e Sequência

Professor: **Jamil Farhat**

UTFPR - Universidade Tecnológica Federal do Paraná - Paraná (Brasil)
DAELN - Departamento Acadêmico de Eletrônica



- Uma **atividade** descreve o comportamento de **transformação de entradas em saídas** através de uma **sequência de ações**;
- A principal representação para modelar este comportamento é o **diagrama de atividades**;

- As **ações** descrevem como as atividades são **executadas**;
- Cada ação pode aceitar **entradas** e produzir **saídas**, chamadas **tokens**;
- Nós de controle, como **join**, **fork**, **decision**, **merge** podem ser usados para controlar o **roteamento de tokens** para uma sequência de ações.

Diagrama de Atividades

- O principal diagrama usado para descrever atividades é chamado de **diagrama de atividades**;
- Cabeçalho da seguinte forma:
 - **act** [tipo de elemento do modelo] nome da ativ. [nome do diagrama]

Diagrama de Atividades

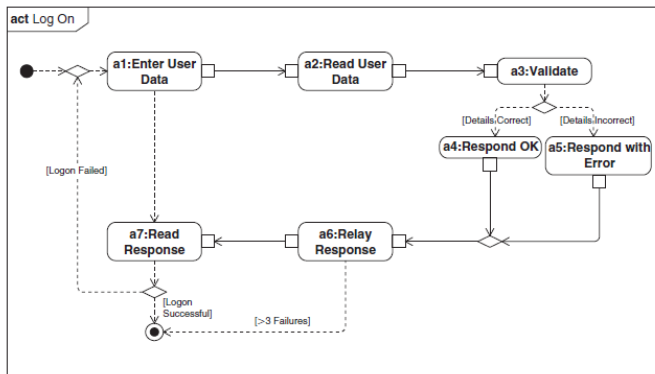


FIGURE 9.1

An example activity diagram.

- Os **tokens** contêm os valores de **entradas**, **saídas** e **controlam o fluxo** de uma ação para outra.
- Uma **ação** **processa tokens** colocados em seus pinos.
- Um **pin** atua como um buffer onde os **tokens de entrada e saída** podem ser **armazenados** antes ou durante a execução.

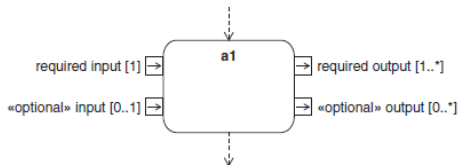
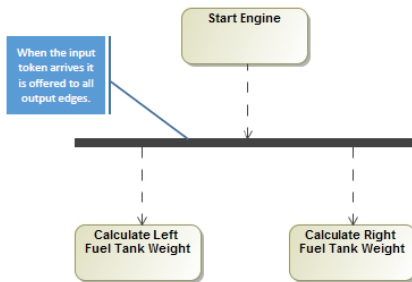


FIGURE 9.3

An action with input and output pins and input and output control flow.

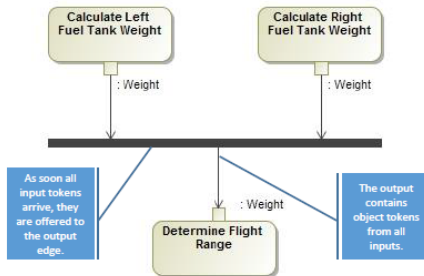
- Porém, há muitas situações em que simplesmente conectar nós não permite uma descrição adequada do fluxo de tokens através da atividade;
- SysML fornece vários mecanismos para expressões mais sofisticadas para fluxos de roteamento;

- **Fork:** um fluxo de entrada e um ou mais fluxos de saída – replica cada token de entrada que ele recebe em cada um de seus fluxos de saída;



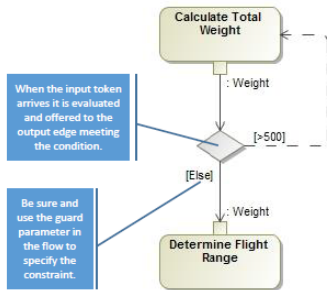
One input, multiple outputs.

- **Join:** um fluxo de saída e um ou mais fluxos de entrada.
Comportamento padrão é produzir tokens de saída somente quando um token de entrada estiver disponível em cada fluxo de entrada (sincroniza o fluxo de tokens de muitas fontes).



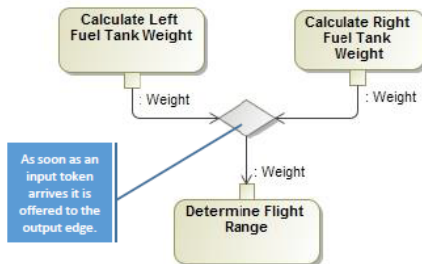
Multiple inputs, one output.

- **Decision:** Um token de entrada só pode atravessar um fluxo de saída. O fluxo de saída é normalmente estabelecido colocando proteções mutuamente exclusivas em todos os fluxos de saída.



One input, multiple outputs, but only one at a time.

- **Merge:** Ao contrário de um nó de junção, um nó de mesclagem não requer tokens em todos os seus fluxos de entrada antes de oferecê-los em seu fluxo de saída. Em vez disso, oferece tokens em seu fluxo de saída à medida que os recebe.



Multiple inputs, one output.

Diagrama de Sequência

- Uma abordagem alternativa para representar a **interação entre elementos do sistema**, como uma **sequência de troca de mensagens**, é o **diagrama de sequência**;
- A interação pode ser entre o **sistema e seu ambiente** ou **entre os componentes de um sistema** em qualquer nível da hierarquia;
- Uma mensagem pode representar a invocação de um serviço de um componente ou o envio de um sinal.

- Os elementos fundamentais deste bloco são as **linhas de vida**;
- O diagrama de sequência descreve a interação entre essas linhas de vida como uma série ordenada de ocorrências.
- Cabeçalho da seguinte forma:
 - **sd** [interaction] nome da interação [nome do diagrama]

- A representação de uma linha de vida é mostrada por um retângulo (cabeçalho) com uma linha tracejada descendo da base até o fim do diagrama;
- O retângulo contém o nome e tipo do proprietário.

Diagrama de Sequência

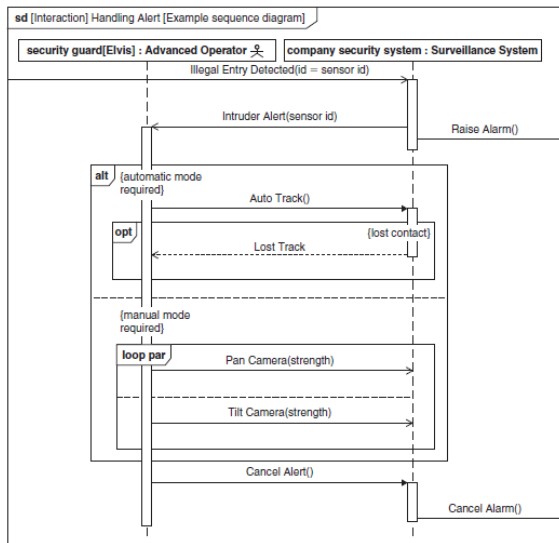


FIGURE 10.1

An example sequence diagram.

- Mensagens podem ser trocadas entre as instâncias representadas por linhas de vidas;
- Uma mensagem representa uma invocação ou solicitação de serviço;
- Embora seja modelado para informações, também podem indicar material ou energia.

Diagrama de Sequência

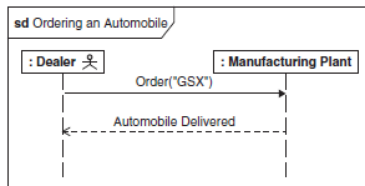


FIGURE 10.4

A simple example of message exchange.

- Os dois tipos básicos de mensagens são **assíncronas** e **síncronas**.
- A **mensagem síncrona aguarda resposta** do receptor informando que seu processamento foi concluído antes de continuar a execução.
- As mensagens são representadas por setas entre as linhas de vida;

- A forma da ponta e o estilo da linha da seta indica o tipo da mensagem:
 - Uma **seta simples** indica uma **mensagem assíncrona**;
 - Uma **seta preenchida** significa uma **mensagem síncrona**;
 - Uma **seta simples** em uma **linha tracejada** representa uma mensagem de resposta.

Diagrama de Sequência

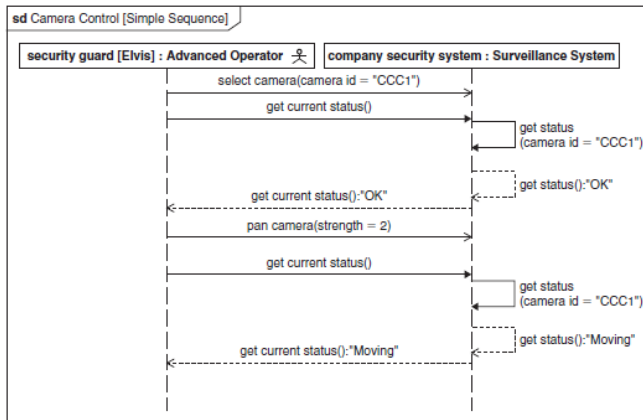


FIGURE 10.5

Synchronous and asynchronous messages exchanged between lifelines.

- Outra informação importante está relacionada aos operadores de interação:
 - **Par**: operador no qual os operandos podem ocorrer em paralelo, cada um seguindo regras de sequenciamento fracas (sequenciamento diferente para diferentes linhas de vida).
 - **Alt/else**: operador no qual exatamente um de seus operandos será selecionado com base no valor de escolha.

- Em acréscimo, outra informação importante está relacionada aos operadores de interação:
 - **Opt**: operador **alt** com um operando. Ele é executado ou ignorado dependendo do valor da escolha.
 - **Loop**: operador no qual execução se repete até o término da restrição.

Diagrama de Sequência

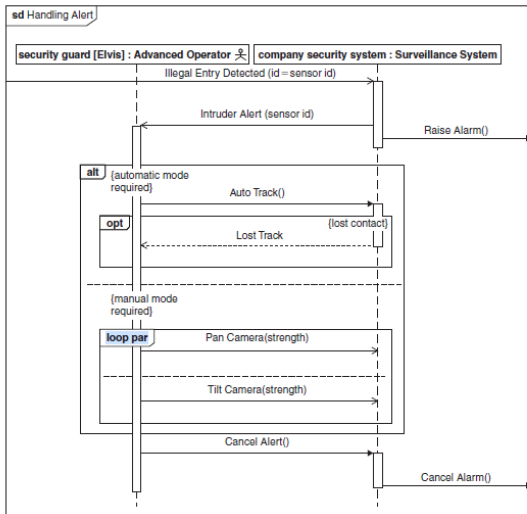


FIGURE 10.12

Complex interaction described using interaction operators.

Atividade: Definição do Projeto

- Para **próxima aula** confeccionar uma apresentação (máximo 5 minutos) do projeto escolhido considerando os conceitos de Engenharia de Sistemas baseado em Modelos:
 - Escolha um subsistema do projeto de vocês e confeccione o diagrama de Atividades e Interações;
 - Escolha outro subsistema e confeccione o diagrama de Máquina de Estados e Casos de Uso (considerando os atores associados);
- Deverá ser realizado upload da apresentação via Moodle até **às 14h40 do dia da apresentação**.

Dúvidas?