Receivers

Prof. Göhner

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thesis No: | Type: | Class of Process Model: | | Student: | | Supervisor: | Begin: | End: |
|  |  |  | |  | | / |  |  |
| Document: | | | Version: | Author: | | Date: | Status: | |
|  | | |  |  | | 05.01.17 |  | |
| Filename: | | | | | Pages: | Print Date: | Template: | |
|  | | | | |  | 12/22/1999 2:53 PM |  | |

**Template Version Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Author | QA | Date | Status | Changes |
| 1.0 | Ringler | Fl | 15.11.00 | Accepted | Creation |
| 1.1 | Serrano | We | 04.10.04 | Accepted | Layout, added example for use cases |

**Document Version Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Author | QA | Date | Status | Changes |
| 0.1 |  |  |  | in progress | Creation |
|  |  |  |  |  |  |

# Tabela de Conteúdo

0 Tabela de Conteúdo 3

1 Modelo de Software do Sistema 5

1.1 Visão Geral dos Casos de Uso 5

1.2 Diagrama de Casos de Uso 5

1.3 Descrição dos Diagramas de Caso de Uso 6

1.4 Diagrama de Sequência do Caso de Uso 9

2 Hardware System Model 9

2.1 Block Diagram 9

2.2 Description of functional blocks 9

2.2.1 Functional block <xxx>\* 9

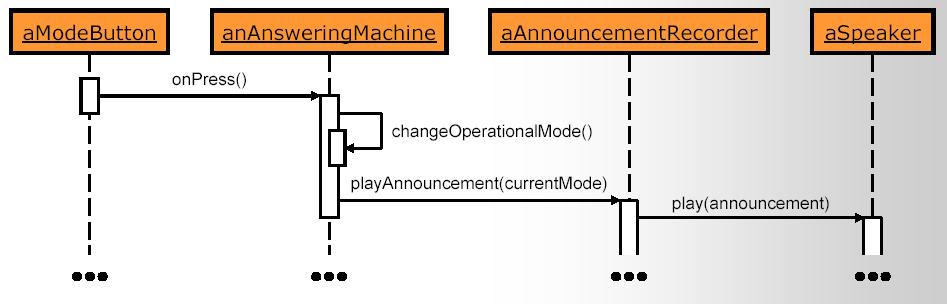
2.2.2 Functional block <xxx>\* 9

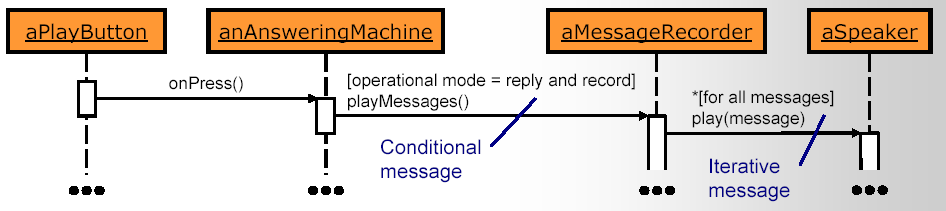
***Hints***

***3. Sequence Diagram***

*The sequence diagram describes interactions between objects from the analysis view. It provides an overview, which objects exchange which messages for a particular scenario.*

*You can use sequence diagrams in addition to the use case descriptions.Sequence diagrams can clarify the course of events and actions.*

**

**

*Figure 4: Sequence Diagrams “Change Operational Mode“  
and “Query Recorded Messages”*

# Modelo de Software do Sistema

## Visão Geral dos Casos de Uso

Os atores definidos foram:

1. Estudante;
2. Robotino.

Os casos de uso identificados foram:

1. Iniciar a execução do programa;
2. Sair do ponto de partida;
3. Percorrer o caminho;
4. Detectar obstáculos;
5. Desviar de obstáculos;
6. Detectar ponto de chegada;
7. Mostrar mensagens de erro;
8. Reiniciar a execução do programa.

## Diagrama de Casos de Uso

O diagrama de casos de uso do sistema está ilustrado na Figura 1.

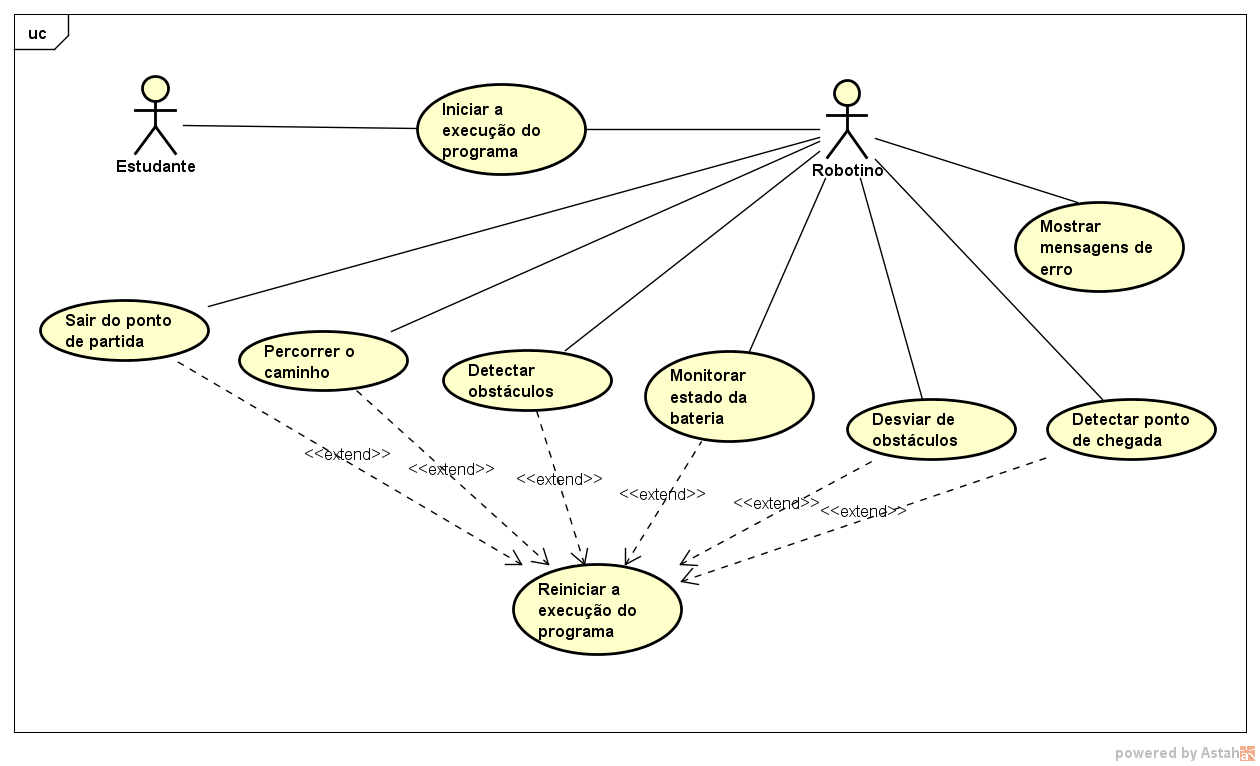


Figure 1. Diagrama de casos de uso do sistema

## Descrição dos Diagramas de Caso de Uso

|  |  |
| --- | --- |
| **CASO DE USO** | **Iniciar execução do programa** |
| **Objetivo** | Iniciar a execução do programa que fará as verificações iniciais, definirá a direção e sentido do movimento rumo ao ponto de chegada. |
| **Categoria** | Primário |
| **Atores externos** | Estudante |
| **Condição prévia** | 1. O Robotino precisa estar com a bateria carregada. 2. O Robotino precisa estar com o sistema operacional inicializado. |
| **Evento de disparo** | O estudante acionou o evento responsável por iniciar a execução do programa. |
| **Sucesso posterior na condição especificada** | O Robotino sai do ponto de partida e começa a percorrer o ambiente (sala). |
| **Falha posterior na condição especificada** | Uma mensagem de erro deve ser mostrada especificando a causa da falha: nível de bateria baixo ou erro no programa (exceção), caso ocorra. |
| **Descrição** | Inicia a execução do programa devendo fazer com que ele saia do ponto de partida.  a duração de todo este processo. |
| **Extensões** | Avaliar Parar Robotino ou Reset no programa do Robotino? |
| **Alternativas** | Executar o programa em outro Robotino. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Sair do ponto de partida** |
| **Objetivo** | O Robotino sai do ponto de partida especificado e inicia seu percurso. |
| **Categoria** | Primário |
| **Atores externos** | Nenhum |
| **Condição prévia** | O programa já deve ter iniciado sua execução no Robotino (caso de uso “executar o programa no Robotino”). |
| **Evento de disparo** | O programa inicia sua execução. |
| **Sucesso posterior na condição especificada** | O Robotino passa a percorrer o caminho no ambiente (caso de uso “percorrer o caminho”). |
| **Falha posterior na condição especificada** | O Robotino não sai do ponto de partida. |
| **Descrição** | O Robotino sai do ponto de partida e começa a percorrer a sala. A partir deste momento ele já pode detectar obstáculos e desviar deles com o objetivo de atingir o ponto de chegada. |
| **Extensões** | Nenhuma |
| **Alternativas** | Executar o programa novamente. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Percorrer o caminho** |
| **Objetivo** | Percorrer o ambiente. |
| **Categoria** | Primário |
| **Atores externos** | Nenhum |
| **Condição prévia** | O Robotino deve ter saído do ponto de partida (caso de uso “sair do ponto de partida”). |
| **Evento de disparo** | O Robotino saiu do ponto de partida. |
| **Sucesso posterior à condição especificada** | O Robotino passa a percorrer o ambiente. |
| **Falha posterior à condição especificada** | O Robotino não sai do ponto de partida. |
| **Descrição** | O Robotino inicia seu percurso ao longo do ambiente. |
| **Extensões** | Nenhuma |
| **Alternativas** | Reiniciar o programa. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Detectar obstáculos** |
| **Objetivo** | Detectar obstáculos que surgirem ao longo do percurso. |
| **Categoria** | Primário |
| **Atores externos** | Nenhum |
| **Condição prévia** | O Robotino precisa estar percorrendo o ambiente (caso de uso “percorrer o ambiente”). |
| **Evento de disparo** | Um obstáculo está situado próximo (distância específica ou um intervalo de distância) ao Robotino. |
| **Sucesso posterior à condição especificada** | O Robotino irá desviar do obstáculo (caso de uso “desviar de obstáculos”). |
| **Falha posterior à condição especificada** | O Robotino irá colidir com o obstáculo. |
| **Descrição** | O Robotino irá identificar que há um obstáculo à frente no seu percurso. |
| **Extensões** | Nenhuma |
| **Alternativas** | Parar movimentação do Robotino. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Desviar de obstáculos** |
| **Objetivo** | Desviar dos obstáculos presentes ao longo do caminho. |
| **Categoria** | Primário |
| **Atores externos** | Nenhum |
| **Condição prévia** | O Robotino deve ter detectado o obstáculo (caso de uso “detectar obstáculos”). |
| **Evento de disparo** | Um obstáculo foi detectado. |
| **Sucesso posterior à condição especificada** | O Robotino desvia do obstáculo, continua seu percurso e retoma o “caminho ideal”. |
| **Falha posterior à condição especificada** | O Robotino irá parar ao colidir com o obstáculo. |
| **Descrição** | O Robotino irá contornar o obstáculo detectado. |
| **Extensões** | Nenhuma |
| **Alternativas** | Reiniciaro programa. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Detectar o ponto de chegada** |
| **Objetivo** | Detectar os obstáculos que surgirem ao longo do percurso. |
| **Categoria** | Primária |
| **Atores externos** | Nenhum |
| **Condição prévia** | O Robotino precisa estar percorrendo o ambiente (caso de uso “percorrer o ambiente”). |
| **Evento de disparo** | O Robotino atingiu o ponto de chegada. |
| **Sucesso posterior à condição especificada** | O tempo de chegada será registrado e o Robotino irá parar. |
| **Falha posterior à condição especificada** | O Robotino continuará percorrendo o caminho, detectando obstáculos e desviando dos mesmos até a bateria acabar. |
| **Descrição** | O Robotino irá identificar as coordenadas especificadas como o ponto de chegada e irá parar. |
| **Extensões** | Nenhuma |
| **Alternativas** | Fazer o Robotino parar. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Mostrar mensagem de erro** |
| **Objetivo** | Mostrar uma mensagem caso haja algum problema ao realizar alguma das operações. |
| **Categoria** | Secundário |
| **Atores externos** | Nenhum |
| **Condição prévia** | O programa deve estar em execução. |
| **Evento de disparo** | Um problema foi detectado em alguma operação do Robotino (exceção). |
| **Sucesso posterior à condição especificada** | A mensagem será mostrada indicando o motivo do erro. |
| **Falha posterior à condição especificada** | Nenhuma mensagem será mostrada. |
| **Descrição** | Uma mensagem de erro ou alerta será mostrada, caso algum erro tenha ocorrido durante a execução de alguma das operações. |
| **Extensões** | Nenhuma |
| **Alternativas** | Nenhuma |

|  |  |
| --- | --- |
| **CASO DE USO** | **Reiniciar a execução do programa** |
| **Objetivo** | Reiniciar o programa no Robotino. |
| **Categoria** | Secundário |
| **Atores externos** | Estudante |
| **Condição prévia** | O Robotino não está apresentando um comportamento de acordo com os requisitos. |
| **Evento de disparo** | O Robotino não realiza uma operação devidamente. |
| **Sucesso posterior à condição especificada** | O Robotino irá parar, reiniciar e, ao iniciar a execução do programa, identificar sua localização em relação ao ponto de chegada para continuar o percurso. |
| **Falha posterior à condição especificada** | Forçar reinicialização com uso de rotinas de Watchdog. |
| **Descrição** | O programa do Robotino será reinicializado. |
| **Extensões** | Nenhuma |
| **Alternativas** | Nenhuma. |

|  |  |
| --- | --- |
| **CASO DE USO** | **Monitorar estado da bateria** |
| **Objetivo** | Monitorar o nível de carga da bateria do Robotino |
| **Categoria** | Primária |
| **Atores externos** | Nenhum |
| **Condição prévia** | O programa já deve ter iniciado sua execução no Robotino (caso de uso “executar o programa no Robotino”). |
| **Evento de disparo** | Ao iniciar o Robotino deve-se ler de nível de carga da bateria |
| **Sucesso posterior à condição especificada** | Nível de carga da bateria é visualizado via mensagem de sucesso. |
| **Falha posterior à condição especificada** | Mensagem de erro na leitura do nível de bateria. |
| **Descrição** | O programa deve monitorar periodicamente a partir do seu início. |
| **Extensões** | Nenhuma |
| **Alternativas** | Nenhuma |

## Diagrama de Seqüência do Caso de Uso

# Hardware System Model

## Block Diagram

## Description of functional blocks

### Functional block <xxx>\*

### Functional block <xxx>\*