

# Comunicação entre agentes no *framework* Embedded-BDI

WESAAC 2024  
Agosto/2024  
Brasília - Brasil

Vítor Luís Babireski Fúrio  
Maiquel de Brito  
Carlos Roberto Moratelli

Universidade Federal de Santa Catarina (UFSC) - Blumenau - Brasil

# Contexto

Sistemas ciberfísicos (CPS): programas de computador, embarcados em diferentes dispositivos, com sensores e atuadores, para,interagir com o ambiente físico.

Requisitos em cenários complexos: autonomia, proatividade, reatividade, habilidades sociais

Ferramentas de *Agent Oriented Programming* (AOP) oferecem estas características, porém com custo computacional incompatível com CPS

Framework Embedded-BDI: AOP para CPS [Santos, 2022]



# Problema e Objetivos

## **Problema:**

Embedded-BDI não fornece suporte à comunicação entre agentes

## **Objetivos:**

Estender o framework para fornecer suporte à comunicação



# Fundamentos

*AgentSpeak*: linguagem para especificação de agentes BDI através de crenças, objetivos e planos

*Embedded-BDI*: versão simplificada de AgentSpeak

- Crenças baseadas em proposições em vez de predicados
- Interpretador não faz alocação dinâmica de memória
- Crenças e objetivos são instanciados quando o agente inicia

```
!start. // desejo (ou objetivo)
+!start <- +happy. //plano para satisfazer o objetivo ``start''
+happy <- !!hello. //plano para tratar a nova crenca ``happy''
+!hello <- say_hello. //plano para satisfazer o objetivo ``hello''
```

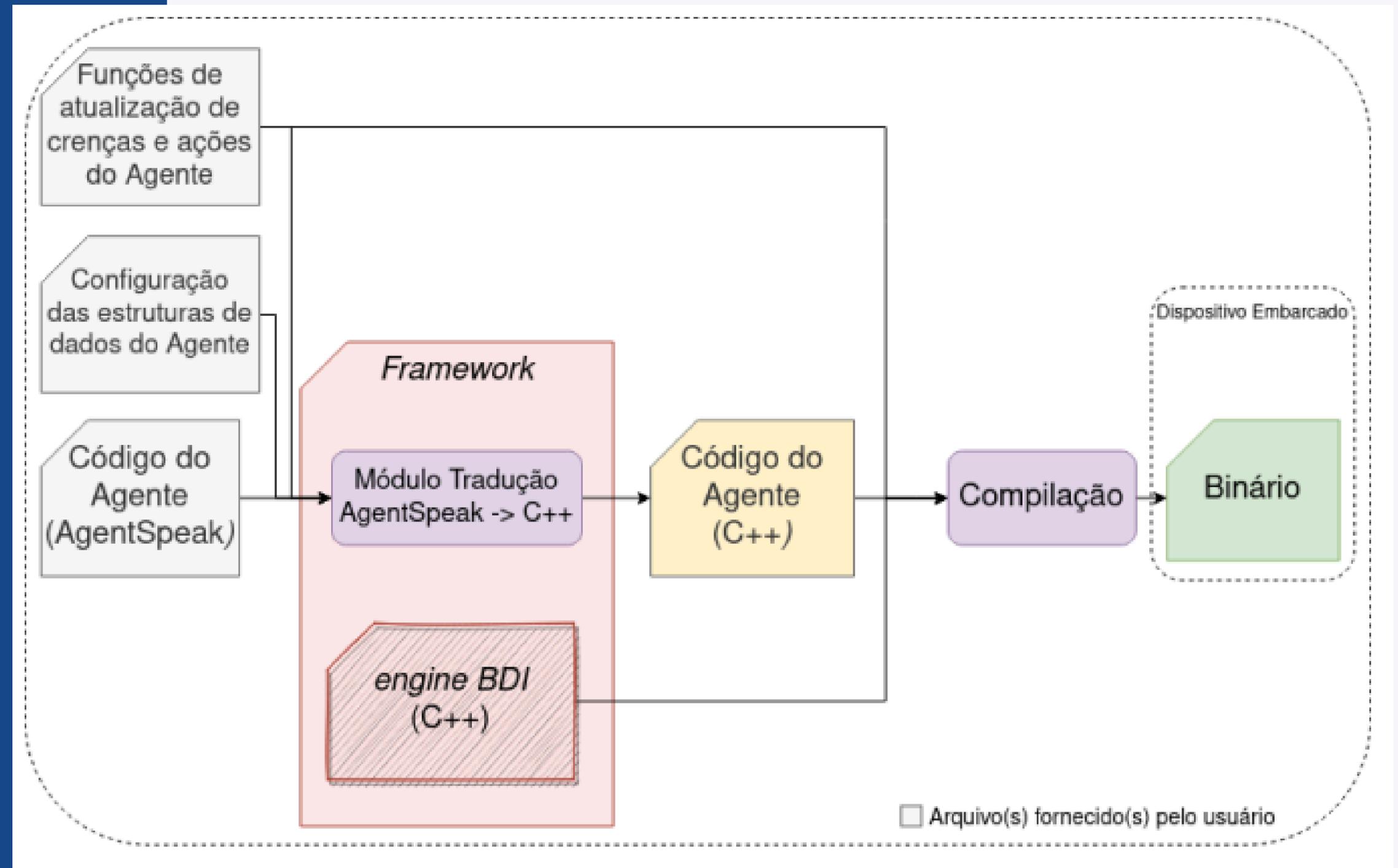
Identificadores de crenças e objetivos são substituídos por números em tempo de execução

```
start - 0
happy - 1
hello - 2
```



# Compilação de um agente em *Embedded-BDI*

# Fundamen tos



# Desenvolvimento

## Integração do ciclo de raciocínio à infraestrutura de comunicação

Interfaces e métodos de acesso à rede dependem do hardware

Solução: comunicação baseada em protocolo conhecido, com amplo suporte de hardware - MQTT

Engine-BDI subscreve um tópico com o mesmo nome do agente e um *broadcast*.

A cada ciclo, a Engine-BDI lê estes tópicos e atualiza o estado interno do agente



# Desenvolvimento

## Conciliação das mensagens MQTT com a semântica do modelo BDI.

Transformação de mensagens MQTT em crenças e objetivos

### Comunicação em KQML

- Performativa: intenção do emissor
- Conteúdo: mensagem transmitida

Inclusão de *internal actions*  
send e broadcast

```
.broadcast(tell,happy)  
.send(bob,achieve,do_something)
```

### A Engine-BDI

- transforma *internal actions* em escrita em tópicos MQTT
- transforma leituras MQTT em crenças e objetivos (cf. performativa)



# Desenvolvimento

## Adaptação às representações internas do *Embedded-BDI*

Crenças e objetivos podem ser mapeados para números diferentes em diferentes agentes

Agente 1

```
start - 0  
happy - 1  
hello - 2
```

Agente 2

```
happy - 0  
start - 1  
hello - 2
```

Solução: armazenar este mapeamento na Engine BDI

- O emissor envia o identificador da crença/objetivo
- O receptor converte o identificador em para seu número correspondente

Outras soluções são possíveis (ex. hash)



# Resultados

Agentes diferentes são capazes de se comunicar

Construtores BDI sobre MQTT

bob.asl

```
!start.  
its_night.  
+!start  
  <- say_hello;  
    .broadcast(achieve,hello).  
+!is_day  
  <- say_night;  
    .send(alice,tell,its_night).
```

alice.asl

```
+!hello  
  <- say_hello;  
    .send(bob,achieve,is_day).  
+its_night <- say_its_night.
```

Console bob

```
Hello, I'm Bob!!  
Sending message to topic broadcast: ACHIEVE/hello  
Message arrived on topic bob: ACHIEVE/is_day  
It's night now, Alice.  
Sending message to topic alice: TELL/its_night
```

Console alice

```
Message arrived on topic broadcast: ACHIEVE/hello  
Hello Everyone, I'm Alice!  
Sending message to topic bob: ACHIEVE/is_day  
Message arrived on topic alice: TELL/its_night  
Meh, it's night, I'm going to sleep...
```



# Conclusões

Capacidade de comunicação adicionada ao *Embedded-BDI*

Execução em ESP32, ESP8266 ou superiores

Abordagens existentes são acopladas a hardwares específicos [William, 2022]

MQTT usado para viabilizar o “transporte” de mensagens (pode não ser a melhor escolha)

Trabalhos futuros: testes em outros dispositivos, medição de recursos de hardware requeridos



# Referências

Bordini, R. H., Hübner, J. F., and Wooldridge, M. J. (2007). Programming Multi-Agent Systems in AgentSpeak Using Jason. J. Wiley

Santos, M. (2022). Programação orientada a agentes bdi em sistemas embarcados. Dissertação (mestrado), Universidade Federal de Santa Catarina, Florianópolis, Brasil.

William, J. (2022). Multi-agent systems on ultra low power platforms. Dissertação (mestrado), ETH Zurich, Suíça.

