

**Modulação de uma Arquitetura de um Sistema Multiagente**

**Sistemas Robóticos e CIM**

Outubro 2016

Duarte Gonçalves nº 40581

André Lourenço nº 41038

Índice

[1. Introdução 3](#_Toc464166136)

[2. Arquitetura 4](#_Toc464166137)

[3. Comunicação 5](#_Toc464166138)

[4. Diagramas de Sequência 6](#_Toc464166139)

[5. Diagramas de Classe 8](#_Toc464166140)

[*5.1.* *Product Holon (PH)* 8](#_Toc464166141)

[*5.2.* *Task Holon (TH)* 8](#_Toc464166142)

[*5.3.* *Supervisor Holon (SH)* 9](#_Toc464166143)

[*5.4.* *Operational Holon (OH)* 9](#_Toc464166144)

[6. Diagramas de Atividade 10](#_Toc464166145)

[*6.1.* *Product Holon (PH)* 10](#_Toc464166146)

[*6.2.* *Task Holon (TH)* 10](#_Toc464166147)

[*6.3.* *Supervisor Holon (SH)* 11](#_Toc464166148)

[*6.4.* *Operational Holon (OH)* 12](#_Toc464166149)

[7. Conclusão 13](#_Toc464166150)

[8. Referências 13](#_Toc464166151)

Índice de Figuras

[Figura 1: Arquitetura [1] 4](#_Toc464166152)

[Figura 2: FIPA Request Protocol [2] 5](file:///G:\Documentos\FCT\5o%20Ano\1o%20Semestre\SRCIM\TP_1\SRCIM_Relatorio_Trab1_40581_41038.docx#_Toc464166153)

[Figura 3: FIPA ContractNet Protocol [3] 5](file:///G:\Documentos\FCT\5o%20Ano\1o%20Semestre\SRCIM\TP_1\SRCIM_Relatorio_Trab1_40581_41038.docx#_Toc464166154)

[Figura 4: Interação entre PH e TH [1] 6](#_Toc464166155)

[Figura 5: Interação entre TH e SH [1] 6](#_Toc464166156)

[Figura 6: Interação (FIPA ContractNet)) entre SH e OH 7](#_Toc464166157)

[Figura 7: Interação (FIPA Request) entre SH e OH 7](#_Toc464166158)

[Figura 8: Diagrama de Classe do ProductHolon 8](#_Toc464166159)

[Figura 9: Diagrama de Classe do Task Holon 8](#_Toc464166160)

[Figura 10: Diagrama de Classe do Supervisor Holon 9](#_Toc464166161)

[Figura 11: Diagrama de Classe do Operacional Holon 9](#_Toc464166162)

[Figura 12: Diagrama de Atividades do Product Holon 10](#_Toc464166163)

[Figura 13: Diagrama de Atividades do Task Holon 11](#_Toc464166164)

[Figura 14: Diagrama de Atividades do Supervisor Holon 12](#_Toc464166165)

[Figura 15: Diagrama de Atividades do Operacional Holon 13](#_Toc464166166)

# Introdução

Nas últimas décadas deparámo-nos com uma mudança significativa nos mercados e na maneira como os consumidores os abordam. Estes necessitam cada vez mais de produtos customizados o que leva a que quem não esteja preparado para tal eventualidade se arrisque a perder competitividade e inclusivamente a desaparecer do mercado. Como tal, e com a necessidade das empresas em conseguir responder aos diversos pedidos, é necessária uma mudança de paradigma para um que nos permita ser flexíveis e ágeis enquanto mantemos a qualidade do nosso produto e a produtividade da nossa fábrica.

Tendo isto em mente o objetivo deste projeto é desenhar e implementar um *Holonic Manufacturing System (HMS)* que permitirá fazer frente a estas novas necessidades. O seu objetivo principal é dar a capacidade de mutação, flexibilidade, dinamismo e autonomia necessários para responder aos diversos pedidos e alterações.

Este projeto está dividido em três fases. A primeira consiste no desenho e na modulação da arquitetura, a segunda a implementação da referida arquitetura e o terceiro a integração do trabalho anterior em hardware. Este relatório cobre a primeira fase do projeto.

Foi escolhido o HMS devido ao facto de não só ser possível escolher entre uma abordagem hierárquica e uma não hierárquica, escalável, modular e reconfigurável é também tolerante a erros. Queremos com isto dizer que enquanto que num *Evolvable Assembly Systems* ou *Evolvable Production Systems* se há algum erro se desperdiça a alocação realizada num HMS devido à presença do *supervisor* e da propagação de falhas toda a arquitetura se adapta às novas condições.

# Arquitetura

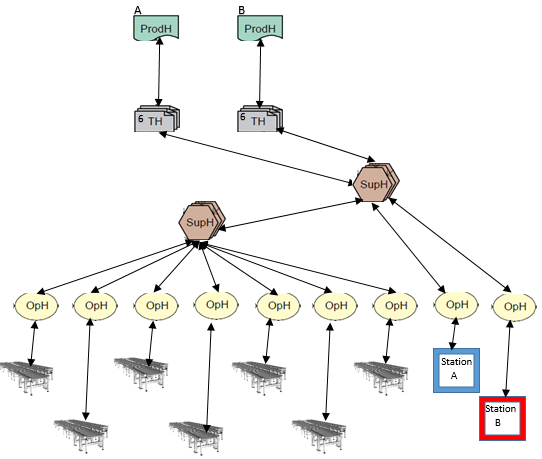


Figura : Arquitetura [1]

Na Figura 1 pode-se observar a arquitetura ADACOR a ser implementada no decorrer deste projeto.

Os *Product (PH), Task (TH), Supervisor (SH)* e *Operacional (OH) Holons* são os módulos genéricos que constituem o sistema com as propriedades desejadas. O PH representa um produto disponível para ser produzido na fábrica. É também a ponte entre o nível de planeamento e o nível físico contribuindo para a integração de funções de controlo como planeamento, calendarização e execução de planos. O TH é a representação de cada work order entregue. Contém informação relacionada com a ordem e é responsável por gerir a sua execução. O OH representa os recursos físicos disponíveis no nível físico tais como robots gerindo os seus comportamentos de acordo com os objetivos do recurso e as suas *skills*. Gere também a agenda dos recursos, i.e, a lista de ordens que os recursos têm de efetuar ao longo do tempo. Por fim com a existência de diferentes níveis de hierarquia aparece a necessidade de ter uma entidade que coordene as capacidades de cada recurso e ofereça serviços combinados e complexos. O SH é então o responsável por tal coordenação.

O nosso projeto irá então conter dois *product holons* de modo a representar os produtos A e B, 12 taks holons que serão movimentar os tapetes para os lados, para cima e para baixo juntamente com uma operação específica de cada estação, 2 *supervisor holons sendo 1 dedicado aos tapetes e outro às estações* e 10 *operational holons* representando 8 *convoyers* e 2 estações.

Ao longo deste relatório estes agentes serão descritos em detalhe.

# Comunicação

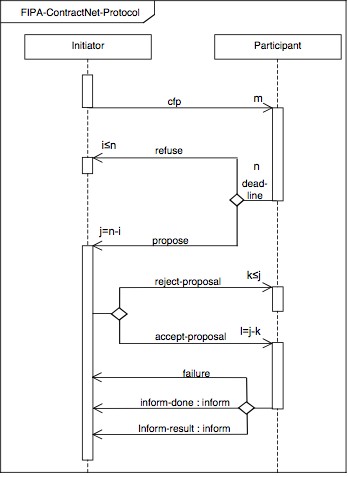
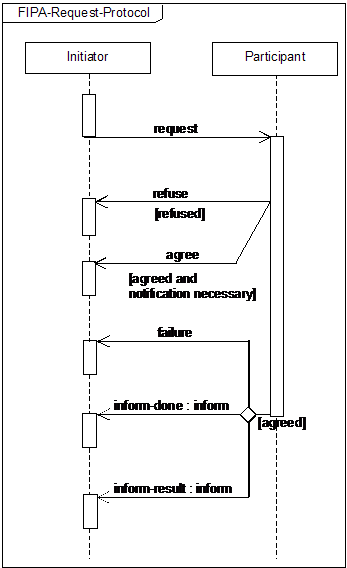


Figura : FIPA Request Protocol [2]

Figura : FIPA ContractNet Protocol [3]

Os protocolos implementados para a comunicação entre agentes são o *FIPA-Request* e o *FIPA-ContractNet*. O último é usado para a negociação entre agentes. O seu objetivo é comparar todos as ofertas de agentes que possuam uma *skill* específica pedida pelo agente e permitir ao mesmo decidir qual das propostas recebidas é a melhor. O protocolo *FIPA-Request* é usado de seguida para dar inicio a todo o processo inerente a uma *work order.*

# Diagramas de Sequência

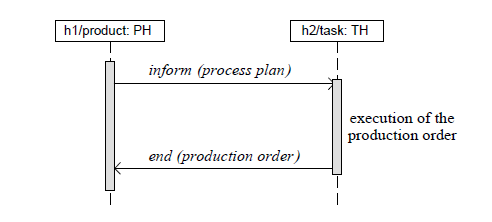


Figura : Interação entre PH e TH [1]

Na Figura 4 podemos observar a interação entre o *Product Holon* e o *Task Holon* em que é usado *FIPA Request* para o *Product Holon* requisitar um produto X e enviar o plano a seguir para o *Task Holon* deixando o mesmo responsável pela execução do plano.

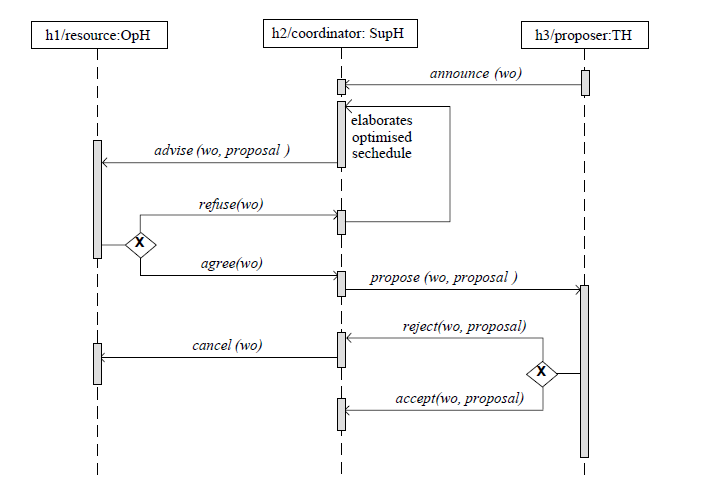


Figura : Interação entre TH e SH [1]

A Figura 5 diz-nos como é feita a coordenação entre os *Task*, *Supervisor* e *Operacional* *Holons*. O TH anuncia então ao(s) *supervisor*(s) a *work order* construída partir do plano de processamento dado pelo PH. De seguida o *supervisor* envia para o(s) OH uma espécie de conselho com uma proposta de resolução sendo que o(s) *operational*(s) pode(m) recusar ou aceitar. Caso recuse o SH elabora novo plano e volta a aconselhar o(s) OH. Caso contrário o *supervisor* envia a proposta ao TH e este pode então aceitar e continuar com esse plano ou recusar sendo que neste caso o *supervisor* cancela a *work order* com o OH.

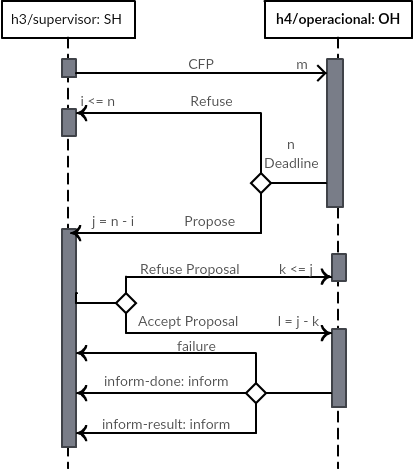


Figura : Interação (FIPA ContractNet)) entre SH e OH

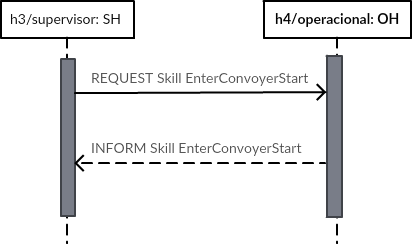


Figura : Interação (FIPA Request) entre SH e OH

Nas Figura 6 e Figura 7 podemos observar que em primeira instância o *Supervisor Holon* procura no seu grupo quem oferece a *skill* desejada fazendo uso do protocolo *FIPA ContractNet* e só depois é lançado o protocolo *FIPA Request* de modo a dar a instrução a quem ganhou o “concurso”. Note-se que na mensagem *cfp(Call For Proposals)* vai descrita a *skill* desejada.

# Diagramas de Classe

Nesta secção é apresentado o modelo para a futura implementação da arquitetura. A ideia é termos um modelo concreto que nos permita simplificar a implementação na próxima fase deste projeto.

Abordamos assim como a implementação de todos os agentes atrás mencionados pode ser realizado.

## *Product Holon (PH)*

A classe do *ProductHolon* tem de ser capaz de representar um produto a ser produzido sendo o mais importante a especificação do conjunto de tarefas necessárias para o produzir. Assim mostra-se necessário ter uma lista de *TaskHolon*’s. Para construir essa lista é necessário que a classe tenha acesso à classe *TaskHolon*.



Figura : Diagrama de Classe do ProductHolon [4]

## *Task Holon (TH)*

Esta classe representa cada tarefa que no seu conjunto constituem todo o processo de fabrico. Como tal é necessário saber que tipo de *skillI* necessita, saber quem é (são) o(s) *supervisor holon(s)* a contactar, conseguir aceitar ou rejeitar propostas enviadas pelo *supervisor* e qual a *work order* a executar. Temos também a possibilidade de adicionar e remover *supervisor’*s para futuras configurações. Para tal temos de ter acesso à classe *SupervisorHolon*.

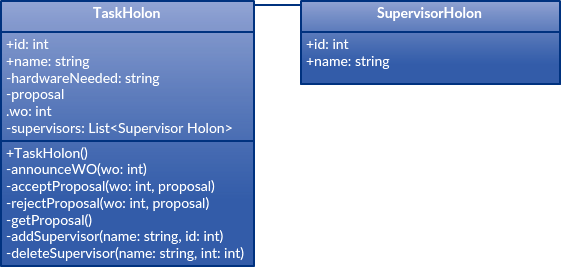


Figura : Diagrama de Classe do Task Holon [4]

## *Supervisor Holon (SH)*

Sendo que é o módulo que negoceia com quem oferece as *skills* necessárias esta classe tem de ter acesso à classe das páginas amarelas (YellowPages) de modo a aceder ao DF que será o nosso mecanismo de comunicação. Tem de saber também que *OperacionalHolon*’s pertencem ao seu grupo antes de poder negociar. Tem também uma lista de *skills* necessárias de modo a não ficar bloqueado à espera que uma tarefa acabe até aceitar outra sendo que assim é possível negociar diversas atividades.

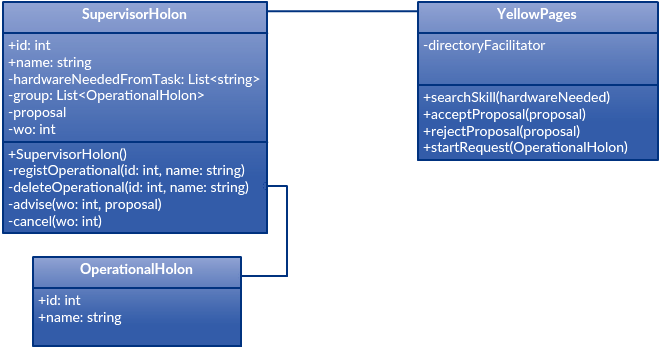


Figura : Diagrama de Classe do Supervisor Holon [4]

## *Operational Holon (OH)*

Por fim esta classe representa o recurso em si e, como tal, é ela que é responsável pelo anúncio de *skills* e consequentemente tem de saber que tipo de *skill* contém. Pode também recusar e aceitar propostas pelo *supervisor holon* e tem de ter um *listener* para saber se alguma das ordens foi cancelada de modo a conseguir desligar o hardware físico.

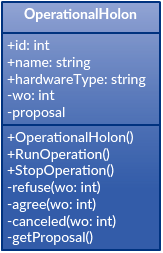


Figura : Diagrama de Classe do Operacional Holon [4]

# Diagramas de Atividade

## *Product Holon (PH)*

O PH representa cada produto disponível para ser produzido pela fábrica e inicia o seu ciclo esperando por ordens de novos produtos. Cada ordem vai gerar uma nova *thread* que se encarregará da sua execução. O PH continuará à espera de novas ordens, podendo processar várias ordens de produtos em simultâneo, limitado pela sua capacidade de produção.

O PH é então responsável pelo planeamento de processos de curto prazo, pelo controlo das subpartes e pelo controlo da produção de ordens de execução.

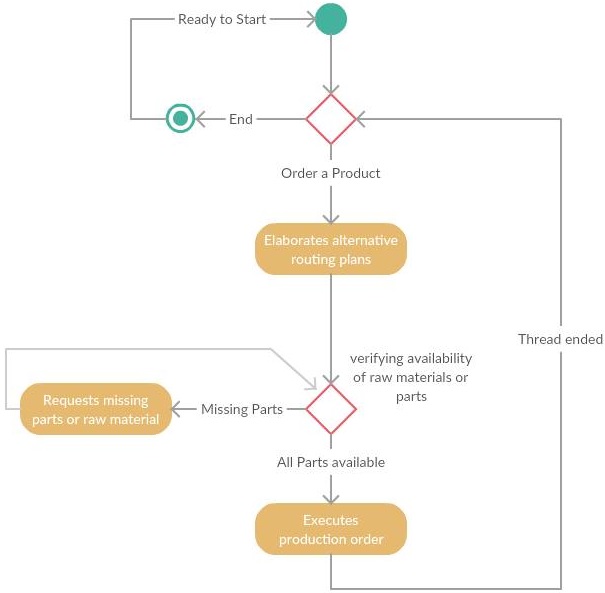


Figura : Diagrama de Atividades do Product Holon

## *Task Holon (TH)*

Um TH representa cada *work order* disponível e é responsável por realizar a decomposição da ordem, por planear a alocação de recursos e por planear atividades de execução.

Inicialmente o TH pede um *pallet* e material para o sistema de armazenamento, seguindo-se a decomposição da *production order* num conjunto de ordens individuais e a análise do plano do processo fornecido pelo PH.

O TH recebe então todas as propostas dos *holons* interessados em realizar essa *work order* e com a *skill* correspondente. Quando receber todas as propostas ou o tempo para esse efeito terminar inicia um procedimento para associar cada ordem à melhor proposta e após a alocação de todas as ordens inicia a execução do plano.

Uma vez começada a ordem, o controlo é dado ao OH e o TH espera pelo fim da execução da *work order*. O processo é repetido para todas as ordens incluídas na *production order*.

Após a execução de todas as ordens de trabalho o TH executa uma série de ações para libertar a palete e finaliza o seu ciclo de vida, transferindo para o PH a informação relevante sobre a execução da produção da ordem.

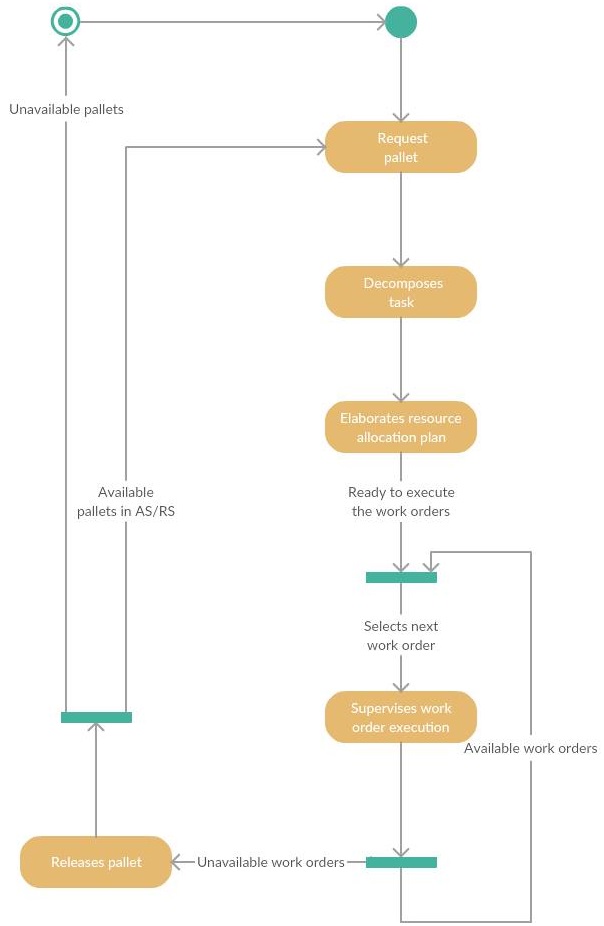


Figura : Diagrama de Atividades do Task Holon

## *Supervisor Holon (SH)*

O SH introduz coordenação e otimização em abordagens de controlo descentralizadas, coordenando vários OH e SH.

O SH demonstra simultaneamente um comportamento reativo – como um servidor espera por pedidos de outros *holons* – e proactivo – gerando planos otimizados que são enviados aos *holons* de baixo nível.

Um SH é então responsável pela elaboração de planos otimizados, pela execução de decisões coordenadas e pela agregação de competências.

É realizada uma elaboração que executa um plano otimizado de todas as ordens de trabalho para todos os *holons* dentro do seu domínio, sendo que esta elaboração ou é executada periodicamente ou é executada para otimizar um horário alcançado depois de um distúrbio.

Podemos então dizer que o SH coordenada um conjunto de *holons* dentro do seu domínio o que faz com que agrupe as habilidades e capacidades dos OH quando estes se juntam ao grupo.

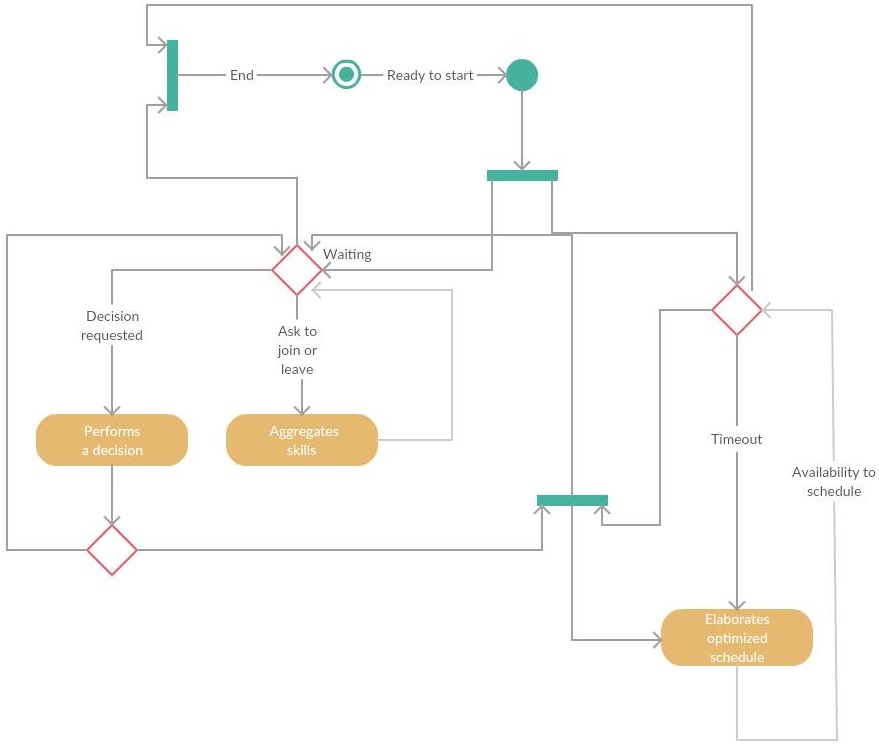


Figura : Diagrama de Atividades do Supervisor Holon

## *Operational Holon (OH)*

Cada OH representa um recurso físico que gere os seus comportamentos de acordo com os seus objetivos, restrições e competências, otimizando os seus objetivos.

Inicialmente o OH inicializa os seus componentes e regista as suas competências e capacidades no SH apropriado.

A análise da disponibilidade para executar uma *work order* requer que o *holon* tenha as habilidades e as capacidades necessárias para executar essa mesma ordem.

Quando o OH recebe uma proposta para a execução da *work order* decide se aceita ou não, baseando-se no seu grau de autonomia e na sua capacidade atual.

A proposta para a execução da *work order* pode ser de 3 tipos diferentes: um pedido para uma operação auxiliar que requer que as habilidades dos recursos sejam compatíveis com os requerimentos da ordem e da sua capacidade, uma decisão de um premio relativo a uma work order anterior enviado pelo SH que requer a confirmação do respetivo plano e por fim uma proposta de *work order* que requer a análise do seu grau de autonomia e da atual capacidade do *holon*.

A próxima *work order* a ser executada é selecionada com base no plano local, na disponibilidade do buffer e no estado da máquina.

Por fim concluímos com base no diagrama de atividade do OH que as atividades de monitorização e adaptação são usadas para regular o comportamento do *holon*.

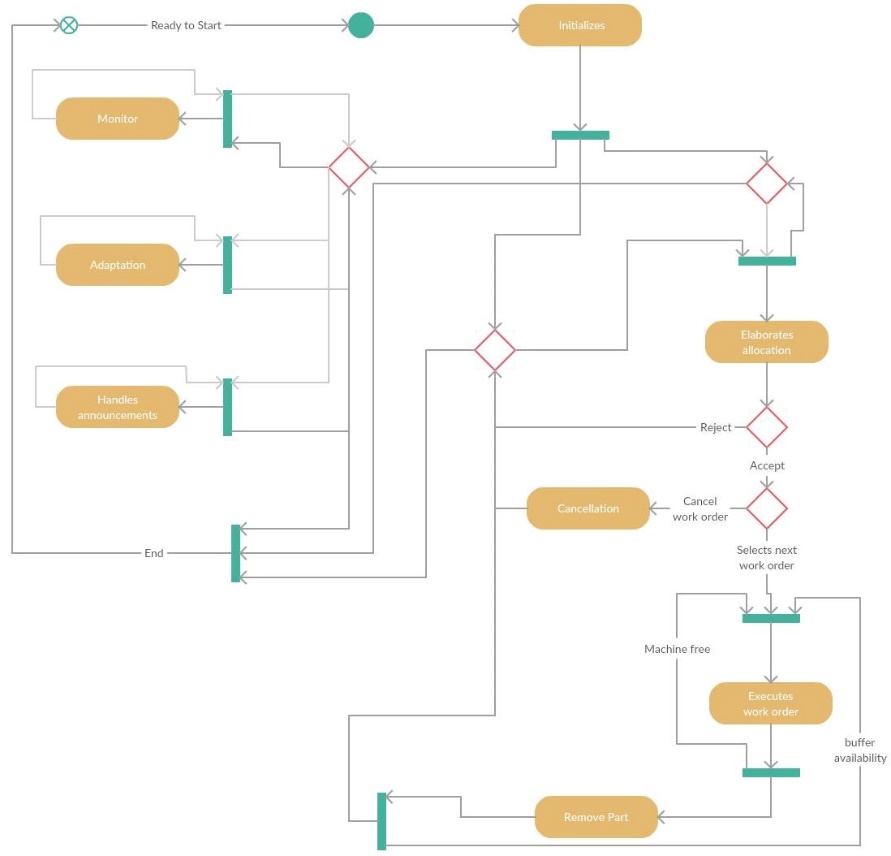


Figura : Diagrama de Atividades do Operacional Holon

# Conclusão

Chegámos então a uma arquitetura multiagente robusta capaz de responder às necessidades impostas pelos mercados e assim alcançar os objetivos propostos. No entanto não é perfeita sendo que poderão aparecer restrições no futuro que nos levem a fazer alterações antes do final do projeto. Tais restrições poderão aparecer principalmente nos diagramas de classe pois só numa fase de implementação teremos a certeza de que atributos e operações necessitamos.

Depois de implementar podemos encontrar dificuldades na aplicação na vida real como a inércia do mundo da indústria visto que as pessoas não estão habituadas ao paradigma e ao facto de que é de difícil implementação fazendo com que fique sempre para trás.

# Referências

[1] Paulo Jorge Pinto Leitão, An Agile and Adaptative Holonic Architecture for Manufacturing Control, 2004

[2] <http://www.fipa.org/specs/fipa00026/SC00026H.html>, visitado a 05/10/2016

[3] <http://www.fipa.org/specs/fipa00029/SC00029H.html>, visitado a 05/10/2016

[4] Dimo Naydenov, António Mendes, Relatório sobre a Arquitetura a Implementar num Sistema de Manufatura Multiagente, 2016