



Nuno Miguel Nave Alcaçarenho

Licenciado em Ciências da Engenharia Eletrotécnica
e de Computadores

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Plataforma de Simulação de Sistemas Distribuídos de Manufatura

Orientador: Doutor João Rosas, Professor Auxiliar, Faculdade de Ciências
e Tecnologia da Universidade Nova de Lisboa



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2016

Plataforma de Simulação de Sistemas Distribuídos de Manufatura

Copyright © Nuno Miguel Nave Alcaçarenho, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Gostaria de agradecer em primeiro lugar ao meu orientador, o Professor Doutor João Rosas, por toda a vontade e disponibilidade demonstradas durante a elaboração desta tese de mestrado.

Queria também agradecer à FCT-UNL pela formação que me foi disponibilizada nestes últimos anos.

Em último lugar mas não sem menos importância gostaria de agradecer aos meus pais por todo o apoio prestado, restante família e amigos.

Da mesma forma que a sociedade vai progredindo, novas necessidades vão sendo criadas. Os consumidores cada vez têm maior preferência por produtos personalizados e assim sendo, as empresas tiveram que se adaptar e corresponder às necessidades e pedidos dos consumidores, na medida de poderem continuar competitivas e interessantes aos olhos quer dos consumidores, quer dos investidores. Neste ambiente global cada vez mais competitivo em que as empresas estão inseridas, onde a tecnologia se encontra em constante evolução, a otimização de processos e a prevenção de falhas torna-se cada vez mais imperiosa de forma a manter os sistemas de produção atualizados e em bom funcionamento. Tendo em conta o carácter global de certas empresas, em que os seus diversos sistemas – produção, controlo, logística – se encontram espalhados pelo globo, é bastante importante que as falhas na cadeia de produção possam ser reduzidas o mais possível.

Para tal, tem-se vindo a recorrer a uma variedade de métodos de previsão e otimização, gestão de risco e identificação de falhas. Tais métodos podem utilizar otimização linear, previsão estatística ou simulação.

Assim sendo, a simulação de processos de manufatura torna-se interessante na medida em que através dela é possível prever comportamentos e realizar alterações necessárias antes de se implementar o sistema.

No entanto, não existem pacotes de simulação no mercado que sejam capazes de simular todos os processos presentes de uma forma abrangente num sistema de manufatura. Por outro lado, o desenvolvimento de um pacote de simulação deste género tornar-se-ia moroso e dispendioso, devido à constante evolução dos sistemas distribuídos de manufatura.

A solução proposta passa então pela criação de uma plataforma de simulação, composta por diversas ferramentas presentes no mercado, que seja capaz de simular o comportamento agregado dos diversos intervenientes que compõem um sistema distribuído de manufatura.

Com a criação de uma ferramenta deste tipo será então possível estudar o comportamento de um sistema de manufatura moderno, podendo prever falhas e otimizar processos.

Palavras-chave: Sistemas Distribuídos de Manufatura, Manufatura, Simulação, Integração de Sistemas, Modelação de Sistemas

Abstract

In the same way that society is progressing, new needs are created constantly. Consumers become keener on personalized products and therefore companies have had to adapt and respond to consumers' needs and requests as long as they remain competitive and interesting to both consumers and investors.

In this increasingly competitive global environment where these companies are inserted and where technology is constantly evolving, process optimization and fault prevention is becoming more and more imperative in order to keep production systems up to date and functioning properly. Given the global nature of certain companies, where their various systems - production, control, logistics - are scattered around the globe, it is very important that the shortcomings in the production chain can be reduced as much as possible. The simulation of DMS (Distributed Manufacturing Systems) can be used to predict industrial behaviors, so they can be corrected and improved – if needed - before the implementation phase.

However, the existing simulation packages available in the market – despite their quality and flexibility - can't broadly simulate every aspect of a modern manufacturing system. Considering the constant evolution of technological systems, developing a system capable of simulate every aspect of a DMS would be a slow and costly process.

The solution proposed in this work combines different simulation tools to create a simulation platform capable of simulate the aggregate behavior of the different components of a DMS. The results provided by this simulation can be used to correct and improve manufacturing processes, saving time and money in the meantime.

Keywords: Distributed Manufacturing Systems, Manufacturing, Simulation, Systems Integration, Systems Modeling

Índice de Conteúdo

1	INTRODUÇÃO.....	1
1.1	MOTIVAÇÃO.....	1
1.2	METODOLOGIA	4
1.3	FORMULAÇÃO DA QUESTÃO DE PESQUISA	6
1.4	PROPOSTA DE DISSERTAÇÃO	7
1.5	CONTRIBUIÇÕES ORIGINAIS	7
2	ESTADO DE ARTE.....	9
2.1	MANUFATURA.....	9
2.1.1	<i>Síntese dos sistemas de manufatura.....</i>	<i>10</i>
2.1.2	<i>Exemplos de sistemas de manufatura.....</i>	<i>10</i>
2.1.3	<i>Tipos de máquinas.....</i>	<i>13</i>
2.1.4	<i>Sistema de gestão de material.....</i>	<i>15</i>
2.1.5	<i>Sistemas de controlo por computador</i>	<i>15</i>
2.1.6	<i>Recursos humanos.....</i>	<i>16</i>
2.1.7	<i>Critérios de classificação de sistemas de manufatura.....</i>	<i>16</i>
2.1.8	<i>Paradigmas de manufatura.....</i>	<i>18</i>
2.2	ALGUNS MECANISMOS DE INTEROPERABILIDADE	20
2.2.1	<i>Web Services.....</i>	<i>20</i>
2.2.2	<i>Serviços REST.....</i>	<i>20</i>
2.2.3	<i>Serviços SOAP</i>	<i>21</i>
2.2.4	<i>Motivação para o uso de Web Services.....</i>	<i>22</i>
2.3	ELEMENTOS DE SIMULAÇÃO.....	24
2.3.1	<i>Processo de simulação</i>	<i>25</i>
2.3.2	<i>Noção de modelo</i>	<i>25</i>
2.3.3	<i>Modelo de simulação</i>	<i>26</i>
2.3.4	<i>Paradigmas de simulação paralela e distribuída.....</i>	<i>28</i>
2.3.5	<i>Simulação cooperativa.....</i>	<i>29</i>
2.3.6	<i>Simulação por computador</i>	<i>29</i>
2.3.7	<i>Simulação com agentes.....</i>	<i>31</i>
2.4	SIMULAÇÃO EM SISTEMAS DISTRIBUÍDOS DE MANUFATURA	31
2.5	PACOTES DE SIMULAÇÃO.....	32
2.5.1	<i>ProModel</i>	<i>32</i>
2.5.2	<i>MODSIM</i>	<i>32</i>
2.5.3	<i>SIMPROCESS.....</i>	<i>33</i>
2.5.4	<i>SIMUL8.....</i>	<i>33</i>
2.5.5	<i>ARENA</i>	<i>34</i>
2.5.6	<i>AnyLogic.....</i>	<i>35</i>
2.6	PACOTES DE SIMULAÇÃO DE ROBÔS.....	36
2.6.1	<i>RobotStudio.....</i>	<i>36</i>
2.6.2	<i>RoboDK</i>	<i>36</i>
2.6.3	<i>RobotExpert</i>	<i>37</i>
2.7	ALGUNS TRABALHOS REALIZADO NA ÁREA DE SIMULAÇÃO DE SDM.....	37
3	DESENVOLVIMENTO DA INFRAESTRUTURA DE SUPORTE.....	41
3.1	MODELAÇÃO DE UM SDM PARA SIMULAÇÃO	41
3.2	IDENTIFICAÇÃO DOS REQUISITOS FUNCIONAIS.....	43
3.3	O PACOTE DE SIMULAÇÃO ARENA	44

3.3.1	<i>Modelação e simulação de sistemas através do ARENA</i>	44
3.3.2	<i>Interoperabilidade com o ARENA</i>	49
3.4	O PACOTE DE SIMULAÇÃO ROBOTSTUDIO	50
3.4.1	<i>Modelação e simulação através da utilização do RobotStudio</i>	50
3.4.2	<i>Interoperabilidade com o RobotStudio</i>	51
3.5	ARQUITETURA DA PLATAFORMA DESENVOLVIDA	52
3.5.1	<i>Infraestrutura de interoperabilidade</i>	52
3.5.2	<i>Modelação dos níveis estratégicos e dos nós locais</i>	54
3.5.3	<i>Interface de controlo</i>	56
3.6	PLATAFORMA DE SIMULAÇÃO OBTIDA	57
4	VERIFICAÇÃO E VALIDAÇÃO DA PLATAFORMA DE SIMULAÇÃO.....	59
4.1	VERIFICAÇÃO DO FUNCIONAMENTO DA INFRAESTRUTURA DE INTEROPERABILIDADE	59
4.1.1	<i>Gestão dos nós de simulação</i>	59
4.1.2	<i>Simulação dos nós locais</i>	60
4.1.3	<i>Simulação dos níveis estratégicos</i>	62
4.2	SIMULAÇÃO DE UM SDM.....	63
4.2.1	<i>Simulação do modelo em ARENA</i>	63
4.2.2	<i>Resultados da simulação dos nós locais</i>	64
4.2.3	<i>Simulação da plataforma de simulação agregada</i>	65
4.2.4	<i>Comparação dos resultados obtidos</i>	67
4.3	ANÁLISE DOS RESULTADOS	68
5	CONCLUSÕES	69
5.1	SÍNTESE DO TRABALHO	69
5.2	RESULTADOS OBTIDOS.....	70
5.3	TRABALHO FUTURO	71
6	BIBLIOGRAFIA.....	73
ANEXO 1	77
ANEXO 2	81
ANEXO 3	83
ANEXO 4	87

Lista de Figuras

FIGURA 1.1: EXEMPLO DE UM ESQUEMA DE UM SISTEMA DISTRIBUÍDO DE MANUFATURA GLOBAL (LAUMER 2008).....	3
FIGURA 1.2: OS NOVE PILARES DA INDÚSTRIA 4.0 (RÜßMANN ET AL. 2015)	3
FIGURA 1.3: SISTEMAS DE MANUFATURA ISOLADOS E SISTEMAS CIBERFÍSICOS (RÜßMANN ET AL. 2015).....	4
FIGURA 1.4: MÉTODO DE PESQUISA UTILIZADO BASEADO NO MÉTODO CIENTÍFICO	5
FIGURA 1.5: COMBINAÇÃO DE FERRAMENTAS DE SIMULAÇÃO DE SISTEMAS DISTRIBUÍDOS DE MANUFATURA	6
FIGURA 2.1: EXEMPLO DE UM ESQUEMA DE UM SISTEMA DE MANUFATURA (GROOVER 2008).....	9
FIGURA 2.2: EXEMPLO DE UM TRABALHADOR A OPERAR UMA ESTAÇÃO DE CÉLULA ÚNICA (FRAUNHOFER IFAM 2014).	10
FIGURA 2.3: EXEMPLO DE UMA INSTALAÇÃO FABRIL COM UM AGLOMERADO DE MÁQUINAS A SEREM OPERADOS POR UM CONJUNTO DE TRABALHADORES (SHENYANG MACHINE TOOL 2010).....	11
FIGURA 2.4: EXEMPLO DE UMA LINHA DE MONTAGEM DE UM AUTOMÓVEL, ONDE UM OPERADOR HUMANO DESEMPENHA, DE FORMA GRADUAL, DIVERSAS OPERAÇÕES DE MONTAGEM ATÉ OBTER O PRODUTO FINAL (LAMBORGHINI 2011).....	11
FIGURA 2.5: PRODUÇÃO DE UM RECIPIENTE NUMA LINHA TOTALMENTE AUTOMATIZADA (MARKO BERIC 2010).....	12
FIGURA 2.6: NESTA FIGURA PODE-SE VER UM CONJUNTO DE MÁQUINAS AGLOMERADAS EM FORMATO DE CÉLULA (BCAG 1997).....	12
FIGURA 2.7: EXEMPLO DE SISTEMA DE MANUFATURA FLEXÍVEL (KUKA 2011).	13
FIGURA 2.8: ESQUEMA DE TRABALHO DE UMA MÁQUINA OPERADA MANUALMENTE – A MÁQUINA RECEBE A MATÉRIA A SER TRABALHADA, O TRABALHADOR OPERA A MÁQUINA E NO FINAL DO CICLO O PRODUTO É DESCARREGADO. (GROOVER 2008).....	13
FIGURA 2.9: ESQUEMA DE TRABALHO DE UMA MÁQUINA SEMIAUTOMÁTICA – NESTA CONFIGURAÇÃO O TRABALHADOR HUMANO USA UM PROGRAMA DE CONTROLO PARA OPERAR A MÁQUINA. A SUA PRESENÇA SÓ É NECESSÁRIA NO INÍCIO E NO FINAL DE CADA CICLO, SENDO QUE A MÁQUINA OPERA DE FORMA AUTÓNOMA NO RESTO DO TEMPO. (GROOVER 2008)	14
FIGURA 2.10: ESQUEMA DE UMA MÁQUINA AUTOMÁTICA – NESTE CASO UM PROGRAMA DE CONTROLO AUTÓNOMO É RESPONSÁVEL POR CONTROLAR AS OPERAÇÕES DA MÁQUINA. O TRABALHADOR HUMANO APENAS TERÁ QUE PRESTAR ASSISTÊNCIA DE FORMA PERIÓDICA. (GROOVER 2008).....	14
FIGURA 2.11: DIFERENTES TIPOS DE ROTAS USADAS NA GESTÃO DO MATERIAL NOS SISTEMAS DE MANUFATURA: NA FIGURA (A) PODE VER-SE UM SISTEMA BASEADO NUMA ROTA FIXA, ENQUANTO QUE NA FIGURA (B) PODE VER-SE UM SISTEMA BASEADO NUMA ROTA VARIÁVEL. (GROOVER 2008).15	15
FIGURA 2.12: DIFERENTES PROCESSOS DE TRATAMENTO DE COMPONENTES: NA FIGURA (A) PODE VER-SE O MODELO SINGULAR; NA FIGURA (B) PODE VER-SE UM MODELO <i>BATCH</i> ; NA FIGURA (C) PODE VER-SE UM MODELO MISTO. (GROOVER 2008).....	18
FIGURA 2.13: EXEMPLO DE <i>WEB SERVICE</i> DO TIPO REST (PAULA BERNIER 2010)	21
FIGURA 2.14: EXEMPLO DE UM <i>WEB SERVICE</i> DO TIPO SOAP (DASUN HEGODA 2014).....	22
FIGURA 2.15: ESQUEMA DE PROCESSOS INTRA E INTER-EMPRESARIAIS – NO CASO APRESENTADO, A COMUNICAÇÃO ENTRE O FORNECEDOR E O ARMAZÉM AINDA É FEITA ATRAVÉS DE FORMULÁRIOS WEB E CORREIO ELETRÓNICO (ALONSO ET AL. 2004).....	23

FIGURA 2.16: ESQUEMA DE PROCESSOS INTRA E INTER-EMPRESARIAIS USANDO WEB SERVICES – NESTE CASO AS LINGUAGENS E OS PROTOCOLOS DE COMUNICAÇÃO SÃO STANDARDIZADOS, NÃO SENDO NECESSÁRIO USAR PROTOCOLOS MIDDLEWARE PARA “TRADUZIR” A INFORMAÇÃO ENTRE OS DIVERSOS INTERVENIENTES (ALONSO ET AL. 2004).....	24
FIGURA 2.17: REPRESENTAÇÃO GENÉRICA DE UM MODELO DE NEGÓCIO (SMITH & SARFATY 1993).....	26
FIGURA 2.18: TIPOS DE MODELO DE SIMULAÇÃO (MARIA 1997).....	27
FIGURA 2.19: EXEMPLO DE SIMULAÇÃO COM O PACOTE DE SIMULAÇÃO PROMODEL (PROMODEL 2016)...	32
FIGURA 2.20: EXEMPLO DE SIMULAÇÃO COM O PACOTE DE SIMULAÇÃO MODSIM (MODSIM 2016)	33
FIGURA 2.21: EXEMPLO DE SIMULAÇÃO COM O PACOTE DE SIMULAÇÃO SIMPROCESS (PAU CASAS 2009)	33
FIGURA 2.22: EXEMPLO DE SIMULAÇÃO COM O PACOTE DE SIMULAÇÃO SIMUL8 (SIMUL8 2016).....	34
FIGURA 2.23: EXEMPLO DE SIMULAÇÃO COM O PACOTE DE SIMULAÇÃO ARENA (ALEX ADAMTSEV 2016)	34
FIGURA 2.24: EXEMPLO DE SIMULAÇÃO COM O PACOTE DE SIMULAÇÃO ANYLOGIC (ANYLOGIC 2016).....	35
FIGURA 2.25: EXEMPLO DE UTILIZAÇÃO DO PACOTE DE SIMULAÇÃO ROBOTSTUDIO (ABB 2016)	36
FIGURA 2.26: EXEMPLO DE UTILIZAÇÃO DO PACOTE DE SIMULAÇÃO ROBODK (ROBODK 2016).....	37
FIGURA 2.27: EXEMPLO DE UTILIZAÇÃO DO PACOTE DE SIMULAÇÃO ROBOTEXPERT (SIEMENS 2016).....	37
FIGURA 2.28: DEMONSTRAÇÃO DE UM SISTEMA DISTRIBUÍDO DE MANUFATURA UTILIZANDO ARQUITETURAS DE ALTO NÍVEL (UYGUN ET AL. 2009)	38
FIGURA 2.29: CENÁRIO DE SIMULAÇÃO DISTRIBUÍDA (STRASSBURGER ET AL. 2003).....	39
FIGURA 3.1: CICLO DE DESENVOLVIMENTO DE UM PROJETO DE <i>SOFTWARE</i> (PROMATICS 2016).....	41
FIGURA 3.2: ESQUEMA REPRESENTATIVO DA DECOMPOSIÇÃO VERTICAL DOS PROCESSOS DESDE O NÍVEL ESTRATÉGICO ATÉ AO NÍVEL DE MÁQUINA	42
FIGURA 3.3: ESQUEMA DO SISTEMA DE MANUFATURA EXEMPLIFICADO (LAW & KELTON 2000).....	44
FIGURA 3.4: DETALHE DO MÓDULO DE CRIAÇÃO.....	46
FIGURA 3.5: DETALHE DO MÓDULO DE PROCESSO.....	47
FIGURA 3.6: DETALHE DO MÓDULO DE RECURSOS	47
FIGURA 3.7: DETALHE DO MÓDULO DE FILAS DE ESPERA.....	47
FIGURA 3.8: MODELO DE SIMULAÇÃO REPRESENTATIVO DE UM PEQUENO PROCESSO DE MANUFATURA	48
FIGURA 3.9: ESTABELECIMENTO DAS CONDIÇÕES DE EXECUÇÃO DE CADA ITERAÇÃO	48
FIGURA 3.10: DETALHE DO RELATÓRIO ESTATÍSTICO GERADO AQUANDO DO TÉRMINO DA SIMULAÇÃO.....	49
FIGURA 3.11: MODELO DE SIMULAÇÃO REPRESENTATIVO DE UM PROCESSO DE MANUFATURA MAIS COMPLEXO	50
FIGURA 3.12: DETALHE DO BLOCO DE PROCESSO COMPOSTO POR UM BLOCO VBA	50
FIGURA 3.13: EXEMPLO DO AMBIENTE GRÁFICO PRESENTE NO PACOTE DE SIMULAÇÃO ROBOTSTUDIO	51
FIGURA 3.14: EXEMPLO DE CÓDIGO RAPID IMPLEMENTADO NO PACOTE DE SIMULAÇÃO ROBOTSTUDIO..	51
FIGURA 3.15: ARQUITETURA PROPOSTA PARA A PLATAFORMA DE SIMULAÇÃO.....	52
FIGURA 3.16: EXCERTO DE CÓDIGO UTILIZADO PARA A INTERLIGAÇÃO ENTRE UM CONTROLADOR VIRTUAL DE UM ROBOT E UMA APLICAÇÃO DESENVOLVIDA NO AMBIENTE DE DESENVOLVIMENTO VISUAL STUDIO	53
FIGURA 3.17: CÓDIGO RELATIVO AO SERVIÇO DE COMUNICAÇÃO WCF	54
FIGURA 3.18: MODELO GENÉRICO DE SIMULAÇÃO NO ARENA	54
FIGURA 3.19: DETALHE DO BLOCO DE PROCESSAMENTO NO MODELO COMPLETO DE SIMULAÇÃO	55

FIGURA 3.20: ESQUEMA DE SIMULAÇÃO DOS NÓS LOCAIS	56
FIGURA 3.21: ASPETO GERAL DA <i>INTERFACE</i> DE CONTROLO IMPLEMENTADA	57
FIGURA 3.22: ASPETO GERAL DA PLATAFORMA DE SIMULAÇÃO DESENVOLVIDA	57
FIGURA 4.1: FUNÇÃO DE TESTE DO SERVIÇO DE COMUNICAÇÃO <i>WEB</i> CRIADO	59
FIGURA 4.2: JANELA DE SELEÇÃO DE CONTROLADOR – NESTA JANELA, PRESENTE NA APLICAÇÃO DE CONTROLO, É POSSÍVEL ESCOLHER UM CONTROLADOR REMOTO	60
FIGURA 4.3: MENSAGEM DE AVISO GERADA PELA APLICAÇÃO DE CONTROLO APÓS A ESCOLHA DO CONTROLADOR	60
FIGURA 4.4: COMANDO DE EXECUÇÃO DO CONTROLADOR – APÓS A ESCOLHA DO CONTROLADOR, É POSSÍVEL EXECUTAR A TAREFA ASSOCIADA	60
FIGURA 4.5: MENSAGEM DE AVISO APÓS A EXECUÇÃO DAS TAREFAS DO CONTROLADOR. ESTA MENSAGEM INFORMA O UTILIZADOR DA CONCLUSÃO DA TAREFA BEM COMO DO TEMPO QUE ESTA DEMOROU.	61
FIGURA 4.6: ROBOT "IRB_140_6KG_0.81M" EM OPERAÇÃO	61
FIGURA 4.7: ROBOT "4600-612345MH" EM OPERAÇÃO	62
FIGURA 4.8: MODELO DE SIMULAÇÃO INTERNO CRIADO NO PACOTE DE SIMULAÇÃO ARENA.	62
FIGURA 4.9: GRÁFICO COMPARATIVO DO DESEMPENHO TEMPORAL DA ATIVIDADE "REBITAGEM".	64
FIGURA 4.10: GRÁFICO COMPARATIVO DO DESEMPENHO TEMPORAL DA ATIVIDADE "SOLDADURA"	65
FIGURA 4.11: GRÁFICO COMPARATIVO DO DESEMPENHO TEMPORAL DA ATIVIDADE "ARMAZENAMENTO".	65
FIGURA 4.12: RELATÓRIO GERADO APÓS O TÉRMINO DA SIMULAÇÃO NO PACOTE DE SIMULAÇÃO ROBOTSTUDIO	65
FIGURA 4.13: MODELO DE SIMULAÇÃO UTILIZADO PARA O ESTUDO DO COMPORTAMENTO AGREGADO DO SISTEMA.	66
FIGURA 4.14: GRÁFICO COMPARATIVO DO DESEMPENHO TEMPORAL DAS ESTAÇÕES DE TRABALHO AQUANDO DO TESTE DO COMPORTAMENTO AGREGADO.....	67

Lista de Tabelas

TABELA 3-1: REQUISITOS FUNCIONAIS CONSIDERADOS NO DESENVOLVIMENTO DA PLATAFORMA PROPOSTA	43
TABELA 4-1: TABELA COMPARATIVA DE TEMPOS DE PROCESSAMENTO DA SIMULAÇÃO EFETUADA NO ARENA	63
TABELA 4-2: TABELA COMPARATIVA DOS TEMPOS DE UTILIZAÇÃO DE CADA ESTAÇÃO.....	63
TABELA 4-3: TABELA COMPARATIVA DOS TEMPOS DE PROCESSAMENTO NO MODELO DE SIMULAÇÃO AGREGADO	66

1 Introdução

1.1 Motivação

Num ambiente económico global caracterizado pela constante pressão dos mercados, pela forte competição entre diversas indústrias e entidades e pela necessidade de apresentar níveis de crescimento cada vez maiores no menor intervalo de tempo possível, todas as indústrias – até mesmo aquelas que estão inseridas em mercados monopolísticos – tiveram que adaptar todos os seus processos de desenvolvimento de forma a poderem competir com os concorrentes que surgiram nas últimas décadas.

À medida que a sociedade progride, vão-se criando novos hábitos e necessidades, sendo que os consumidores têm continuamente vindo a assumir preferências por produtos mais sofisticados e personalizados. As empresas, na sua saga por corresponderem a estas exigências, têm também vindo a adaptar-se, nomeadamente, evoluindo os respetivos sistemas de produção.

Efetivamente, após a Revolução Industrial, o paradigma que persistia era o de produção em massa de produtos homogéneos. Este paradigma é perfeitamente ilustrado pelo célebre adágio de Henry Ford em que dizia que “as pessoas podem escolher a cor do carro, desde seja preta!”. Mas atualmente, as empresas tiveram que se adaptar de forma a melhor usufruir as exigências dos consumidores atuais, para além de se terem que adaptar aos mercados cada vez mais globais e competitivos.

Todos estes aspetos incentivaram e obrigaram as empresas a modificarem e evoluírem os seus sistemas de produção, que se tornaram bastante mais complexos de forma a, conforme mencionado, lidarem com a pressão crescente dos consumidores e a concorrência dos competidores, sendo que um dos tipos de sistemas mais afetados por estas modificações são os sistemas de manufatura distribuídos. Com esta crescente complexidade, tem-se tornado cada vez mais difícil o projeto e implementação de sistemas de manufatura. Um sistema que não esteja bem dimensionado poderá no mínimo apresentar pior desempenho ou mesmo não funcionar, tendo-se como consequência enormes prejuízos e o risco de perder a quota de mercado.

É neste contexto que o papel da simulação reveste-se duma particular importância. Na fase de desenvolvimento de um Sistema Distribuído de Manufatura (SDM), a simulação pode até ser usada como paradigma de projeto. Através da simulação consegue-se imitar o comportamento futuro do sistema e assim, estabelecer estimativas para o seu desempenho. Numa fase em que se pretende efetuar modificações a um SDM existente, mercê da variabilidade de produtos ou otimização de processos de fabrico, a simulação permite observar antecipadamente os efeitos destas mudanças. Efetivamente, a simulação é uma estratégia em crescente utilização, contribuindo até como um dos pilares da base da Indústria 4.0 (Rüßmann et al. 2015).

Para isso, tem sido disponibilizado uma grande quantidade e variedade de ferramentas de simulação, que permitem modelar e simular, nomeadamente, sistemas de produção, no caso particular, sistemas distribuídos de manufatura. Só que os SDM são sistemas, conforme mencionado antes, bastante complexos e que para serem simulados de forma adequada necessitam de ser considerados nas suas variadas dimensões, nomeadamente, o nível físico dos sistemas, a interoperabilidade, a coordenação das distintas partes de um sistema, as estratégias de gestão, o facto de estes serem distribuídos e em muitos casos, a própria cadeia logística.

Uma simulação efetiva de um SDM pode requerer a conjugação de várias destas dimensões na construção de um modelo de simulação. No entanto, os pacotes de simulação disponíveis não possuem a capacidade para simular estes sistemas de uma forma abrangente. Um paco-

te de simulação que consiga abarcar em simultâneo muitas destas realidades é também muito difícil de desenvolver. Na realidade, cada pacote de simulação continua a ser desenvolvido para simular aspetos específicos de um sistema de produção. Por exemplo, alguns pacotes conseguem simular o nível físico de sistemas de produção, enquanto que outros estão melhor adaptados para simular os níveis de gestão desses sistemas. Em alternativa, para se poder simular SDM de forma mais abrangente, e assim se poder efetuar previsões mais precisas e fiáveis do respetivo desempenho, propõem-se a combinação de pacotes de simulação já existentes, para assim se conseguir simulações mais abrangentes. Isto torna-se agora possível devido à disponibilidade de mecanismos de interoperabilidade, nomeadamente através de serviços web e devido ao facto dos pacotes de simulação existentes estarem progressivamente a integrar capacidades de interoperabilidade.

O foco deste trabalho consiste então no desenvolvimento de uma plataforma distribuída de simulação que permita a combinação de várias ferramentas de simulação de forma a poder-se modelar e simular SDM de forma abrangente. Inerentemente mais complexa, a simulação de SDM requer a utilização de uma plataforma com uma arquitetura adequada. De forma a poder-se desenvolver esta arquitetura, é necessário primeiro caracterizar os sistemas que vão ser simulados. Para isso é necessário fazer um levantamento de alguns sistemas de manufatura, culminando com os SDM.

Também é necessário investigar como é que se consegue simular estes sistemas. Para os sistemas de manufatura convencionais, já se possui bastantes ferramentas de simulação. No entanto, ainda são necessários avanços significativos para a simulação de SDM, que é onde este trabalho pretende contribuir. Assim sendo, uma questão importante neste trabalho consiste então em como simular sistemas distribuídos de manufatura. É também necessário estudar o próprio processo de simulação e respetivos paradigmas, por exemplo, simulação centralizada versus simulação distribuída ou simulação contínua versus simulação discreta.

Para se poder desenvolver a plataforma de simulação, é necessário considerar os aspetos técnicos relevantes. Um destes aspetos mais importantes refere-se à necessidade duma plataforma distribuída, pelo que a arquitetura que se vai especificar deverá considerar mecanismos de interoperabilidade. Devido à sua natureza aberta e heterogénea, estes mecanismos deverão basear-se em serviços do tipo REST (Pautasso 2008).

Um SDM é constituído por um lado, por nós heterogéneos e autónomos (Figura 1.1). Por outro lado, cada nó é caracterizado pelos diversos níveis concetuais, desde o nível de gestão até ao nível de manufatura, incluindo os processos que estão a ser executados em cada máquina. Para que a plataforma consiga efetuar uma simulação com esta natureza, é necessário integrar os vários pacotes de simulação disponíveis, em que cada pacote deverá então ser adequado para funcionar de forma autónoma e simular os diversos componentes de um SDM, qualquer que seja o seu nível.

Por exemplo, o pacote de simulação ARENA (Alex Adamtsev 2016) pode ser utilizado para simular os níveis de gestão, logística, etc. O pacote de simulação RobotStudio (ABB 2016) permite simular processos que decorrem ao nível de robôs e outros tipos de recursos.

Com a ferramenta disponível, importa então mostrar o seu valor através da simulação de exemplos ilustrativos de SDM. Um aspeto importante aqui seria investigar os resultados dum estudo de simulação para um determinado SDM feito através dum pacote de simulação genérico (como por exemplo, o ARENA) e simular o mesmo sistema com a plataforma desenvolvida. A expectativa é a de que os resultados obtidos permitirão inferir a importância da plataforma para a simulação de SDM.

Nesta nova revolução os sistemas informáticos, os sistemas de manufatura e os sistemas sensoriais estão interligados num único sistema agregado, denominado de sistema ciberfísico. Estes sistemas interagem uns com os outros através da Internet e analisam dados para prevenir falhas, para se autoconfigurarem e para se adaptarem a mudanças no ambiente em que estão inseridos. Este novo paradigma industrial permite a criação de processos mais rápidos, eficientes e flexíveis, reduzindo custos e diminuindo falhas (Figura 1.3).

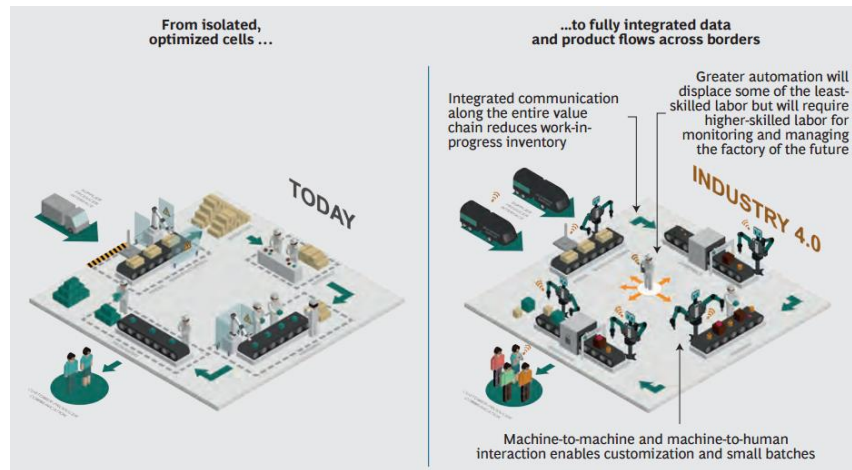


Figura 1.3: Sistemas de manufatura isolados e sistemas ciberfísicos (Rüßmann et al. 2015)

Como referido anteriormente (Rüßmann et al. 2015), um dos pilares essenciais nos sistemas ciberfísicos é a simulação. Esta é usada nos dias de hoje na fase de desenvolvimento de produtos, materiais e processos de produção, mas poderá ser usada de forma mais extensiva nas operações fabris durante os próximos tempos. Estas simulações irão utilizar dados obtidos em tempo real de forma a criar um modelo virtual que espelhe o mundo físico. Este modelo virtual pode incluir máquinas, produtos e operadores humanos.

Através da simulação é assim possível testar e otimizar processos bem como despistar eventuais falhas nos sistemas de produção trabalhando no modelo virtual criado antes da implementação dos sistemas físicos, permitindo a redução dos tempos de configuração destes, aumentando assim a produtividade.

1.2 Metodologia

A metodologia na qual este trabalho assenta baseia-se no Método Científico descrito em (Lakatos & Marconi 1991), sendo que o processo desenvolvido neste trabalho (Figura 1.4) encontra-se dividido em seis seguintes passos:

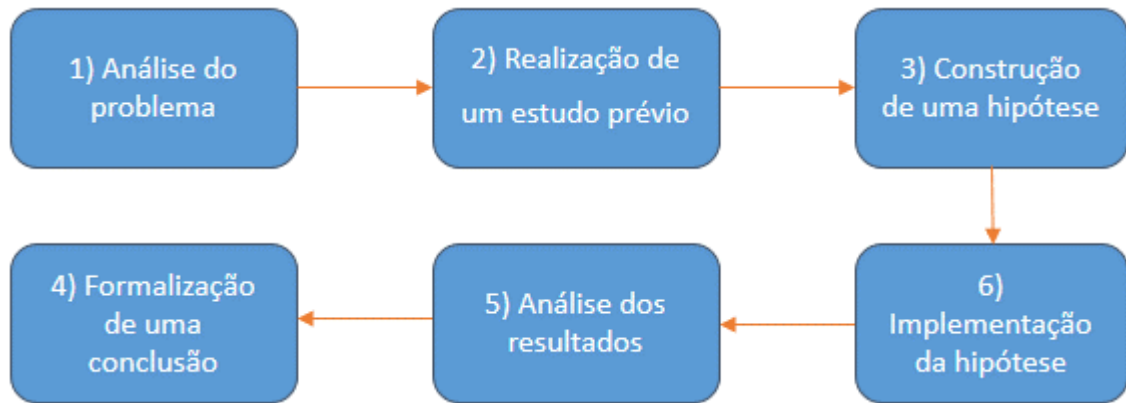


Figura 1.4: Método de pesquisa utilizado baseado no Método Científico

Análise do problema: Numa primeira análise deve-se identificar qual o problema em questão, bem como as respectivas características. No caso descrito, o problema em questão passa pela determinação de uma abordagem adequada para a combinação de diversas ferramentas de simulação de modo a poder criar uma plataforma de estudo e simulação de sistemas distribuídos de manufatura, simulando cada componente de forma individual de forma a poder aferir um resultado global.

Realização de um estudo prévio: Nesta fase serão consultados os diversos trabalhos já realizados sobre o problema em questão, ganhando assim uma base de conhecimento para resolver o problema supracitado.

Construção de uma hipótese: Partindo da base de conhecimento adquirida na fase anterior, será possível compreender de forma mais clara o problema principal deste trabalho, sendo mais fácil formalizar uma hipótese de resolução do problema.

Implementação da hipótese: Numa primeira fase deste passo serão criadas ferramentas para testar a hipótese em questão. Numa segunda fase, as ferramentas criadas serão testadas de forma a obter resultados com vista a poder aferir a validade da hipótese.

Análise dos resultados: Após a conclusão dos testes, nesta fase os resultados serão analisados de modo a verificar a validade da hipótese proposta. Caso a hipótese seja válida os resultados serão reportados, caso a hipótese não seja válida, será formulada uma nova hipótese, sendo implementada e verificada em seguida.

Formalização de uma conclusão: Nesta fase os resultados obtidos na fase de análise – caso seja satisfatórios - serão reportados e publicados.

1.3 Formulação da questão de pesquisa

Nos dias de hoje os sistemas de manufatura estão cada vez mais complexos, sendo difícil prever o seu desempenho de uma forma eficaz. É preciso também ter em conta que a simulação de sistemas distribuídos de manufatura implica simular componentes em diferentes níveis hierárquicos e muitas vezes, em diferentes localizações geográficas.

Uma das alternativas, tal como foi referido anteriormente, consiste na simulação destes sistemas. No entanto, os pacotes de simulação presentes no mercado não são capazes de simular de forma agregada e abrangente todos os aspetos relevantes de um sistema de manufatura distribuído. Logo, seria interessante se fosse possível combinar distintas ferramentas de simulação de forma a realizar simulações de sistemas mais complexos, e com isso, conseguir fazer previsões acerca do seu comportamento e desempenho.

Esta combinação de diferentes ferramentas de simulação começa a tornar-se possível devido ao facto de muitos dos pacotes de simulação presentes no mercado oferecem funcionalidades ao nível de interoperabilidade. Logo, a estratégia para se proceder à combinação destas ferramentas vai consistir na interligação através de mecanismos de interoperabilidade, nomeadamente através de serviços web. Tendo em consideração estes aspetos, a questão de pesquisa é a seguinte:

“Será possível criar uma plataforma de simulação que seja capaz de simular de forma abrangente um sistema distribuído de manufatura?”

Uma hipótese de resposta para esta questão passa, conforme referido antes, pela junção de diversas ferramentas presentes no mercado (Figura 1.5), de forma a criar uma plataforma de simulação distribuída. Esta plataforma, composta por dispositivos locais e remotos, será capaz de simular de forma agregada o comportamento de um sistema distribuído de manufatura.



Figura 1.5: Combinação de ferramentas de simulação de sistemas distribuídos de manufatura

Ao longo deste trabalho vai-se então especificar e contruir esta plataforma agregadora. A validação desta hipótese satisfará assim a questão da pesquisa formulada.

1.4 Proposta de dissertação

Esta dissertação foi estruturada em cinco capítulos distintos que serão caracterizados em seguida:

Introdução: Capítulo inicial desta dissertação, onde é feita uma identificação e caracterização do problema em estudo, a motivação para a resolução deste mesmo problema, a formulação de uma questão de pesquisa e a hipótese de resolução da questão. Neste capítulo é também apresentada a metodologia de trabalho para a hipótese de resolução considerada.

Estado de arte: Nesta secção é feito um estudo sobre as definições de manufatura e simulação, bem como as características e arquiteturas que definem estas duas grandes áreas. Será também abordada a definição de simulação de sistemas distribuídos de manufatura assim como será feito um breve levantamento sobre o trabalho efetuado na área.

Desenvolvimento da ferramenta de suporte: Nesta secção é feito inicialmente um estudo sobre o desenvolvimento de um modelo de simulação de sistemas distribuídos de manufatura, de onde resultará um compêndio de requisitos necessários para o desenvolvimento da ferramenta. Consoante estes requisitos será especificada uma arquitetura que servirá de base ao desenvolvimento da plataforma de simulação distribuída previamente proposta. Esta plataforma será capaz de efetuar simulações tanto localmente como remotamente, sendo esta a arquitetura utilizada para teste da solução proposta.

Validação da ferramenta desenvolvida: Neste capítulo procede-se à validação da plataforma desenvolvida no capítulo anterior, sendo feito um teste dos diversos modos de simulação. Através desta validação será verificado se a questão de pesquisa considerada no segundo capítulo é respondida de forma afirmativa.

Conclusões: Neste quinto e último capítulo são sumarizados os aspetos mais relevantes do projeto desenvolvido, sendo feito um breve sumário de cada etapa do desenvolvimento do trabalho. Na fase final são apresentadas algumas considerações sobre o trabalho futuro que poderá ser desenvolvido tendo como base o trabalho efetuado nesta dissertação.

1.5 Contribuições originais

O presente trabalho pretende oferecer como contribuição original o estudo e desenvolvimento de uma plataforma de simulação de sistemas distribuídos de manufatura. Esta plataforma será constituída por diversos pacotes de simulação que irão cooperar entre si, criando assim um único sistema capaz de simular de forma agregada o comportamento de um sistema distribuído de manufatura.

Para além do desenvolvimento da plataforma, este trabalho pretende clarificar os conceitos de simulação de sistemas distribuídos de manufatura, assim como demonstrar alguns processos para a obtenção de modelos de simulação de sistemas distribuídos de manufatura.

2 Estado de Arte

Neste segundo capítulo serão abordados os conceitos de sistema de manufatura e de simulação, bem como algumas tecnologias presentes neste tipo de sistemas. De acordo com a metodologia e proposta de dissertação delineadas no primeiro capítulo, neste segundo capítulo será feito um levantamento do estado de arte existente relacionado com os conceitos de manufatura e simulação, correspondendo assim ao estudo prévio mencionado no segundo ponto da metodologia. O capítulo prossegue com uma descrição de alguns mecanismos de interoperabilidade e termina com a análise de alguns pacotes de simulação existentes.

2.1 Manufatura

O processo de manufatura é definido como a conversão de matérias-primas e componentes em produtos finais de valor acrescentado, que correspondam às especificações e características pretendidas pelos clientes finais. Este processo é normalmente desempenhado por uma configuração homem-máquina, onde existe uma divisão das tarefas e um nível de produção de larga escala.

Um sistema de manufatura conforme o esquema da figura 2.1 pode ser dividido em quatro sub-componentes presentes na maioria dos sistemas:

- A maquinaria necessária para a manufatura de um dado produto;
- Os mecanismos necessários para o transporte de matérias-primas e/ou material relevante de e para as máquinas afetas aos sistemas de manufatura;
- Um sistema de controlo informático capaz de monitorizar todo o processo de manufatura;
- Operadores humanos para atuarem e controlarem o processo de manufatura.

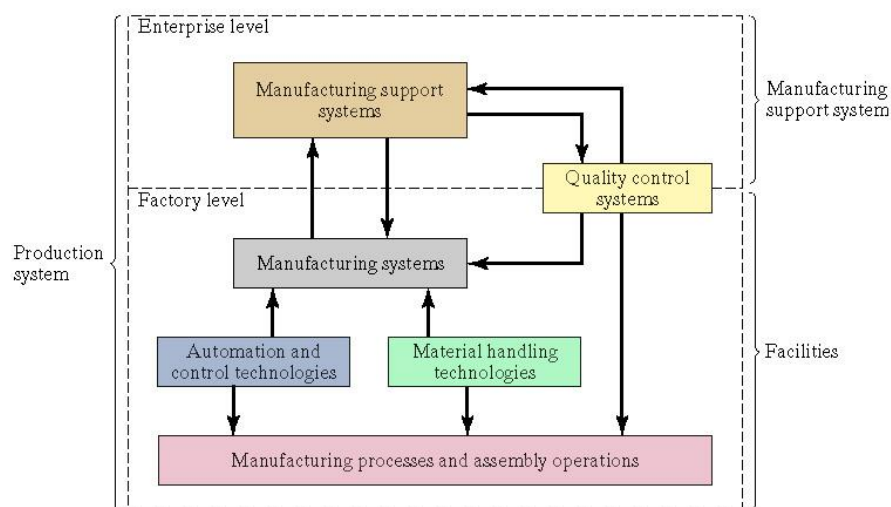


Figura 2.1: Exemplo de um esquema de um sistema de manufatura (Groover 2008)

De forma a poder-se simular SDM, é necessário considerar os aspetos mais relevantes das componentes destes sistemas. Isto inclui os próprios sistemas de produção, que necessitam de uma caracterização adequada, sendo este o foco das secções seguintes.

2.1.1 Síntese dos sistemas de manufatura

Os sistemas de manufatura propriamente ditos nasceram com o advento da Revolução Industrial (Michael George 2002). Com a criação de componentes substituíveis em meados do século XIX foi possível aos sistemas de manufatura darem o passo seguinte – a criação de linhas de montagem no início do século XX, permitindo assim a produção em massa de bens e produtos de valor acrescentado. Com a massificação da tecnologia na segunda metade do século XX, foi possível introduzir robôs e computadores nas linhas de montagem, melhorando o controlo dos sistemas e acelerando os processos de manufatura (EPIC 2013). Tendo em conta a interligação entre sistemas à escala global nos dias de hoje, os sistemas de manufatura não fugiram à regra, estando cada vez mais assentes num paradigma de distribuição de sistemas, podendo assim um sistema de manufatura completo – desde as linhas de produção ao departamento de logística, passando pelas secções de controlo - estar distribuído por diversas partes do globo.

2.1.2 Exemplos de sistemas de manufatura

Os sistemas de manufatura podem ser divididos em diversos tipos consoante a sua configuração, tipo de máquinas utilizadas, entre outros. Em seguida ficam alguns exemplos de sistemas de manufatura fundamentalmente inspirado em (Groover 2008):

Estações de célula única (*single cell station*): É o sistema de manufatura mais utilizado na indústria. Tem como principais características o facto de as suas operações poderem ser independentes das outras estações de manufatura e poder realizar operações tanto de processamento como de montagem. Na figura 2.2 é possível observar um trabalhador a operar uma estação de célula única.



Figura 2.2: Exemplo de um trabalhador a operar uma estação de célula única (Fraunhofer IFAM 2014).

Aglomerado de máquinas (*machine cluster*): Um aglomerado de máquinas pode ser definido como um conjunto de duas ou mais máquinas capazes de produzir produtos com tempos de ciclo idênticos, sendo operadas por um trabalhador. Na figura 2.3 é possível observar um conjunto de trabalhadores a operar um grupo de máquinas semi-automatizadas.



Figura 2.3: Exemplo de uma instalação fabril com um aglomerado de máquinas a serem operados por um conjunto de trabalhadores (Shenyang Machine Tool 2010)

Linhas de montagem manuais: consiste numa série de estações de trabalho onde são realizadas operações de montagem de forma a construir gradualmente um produto, como por exemplo um automóvel (Figura 2.4). Estas linhas permitem que os vários subprocessos de montagem possam ser realizados de forma mais rápida e fácil, assim como facilita a manutenção e a substituição de componentes da linha de montagem.

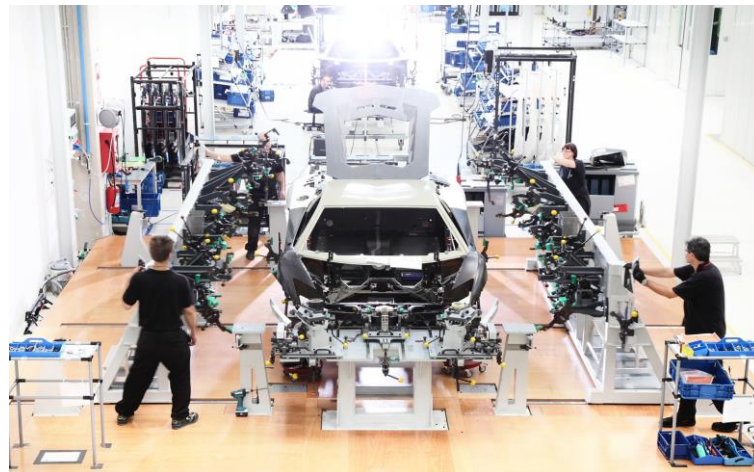


Figura 2.4: Exemplo de uma linha de montagem de um automóvel, onde um operador humano desempenha, de forma gradual, diversas operações de montagem até obter o produto final (Lamborghini 2011).

Linhas de transferência automática: é uma linha de produção que consiste numa série de estações de trabalho que realizam operações de processamento de forma automática, sendo que a transferência de peças entre estações de trabalho é também automatizada (Figura 2.5). Este tipo

de sistemas tende a ser usado em situações que requeiram grandes quantidades de produtos, situações onde o *design* do produto não seja alterado muitas vezes e em situações onde são necessárias diversas operações para produzir o produto. Este tipo de sistemas caracteriza-se por oferecer altas taxas de produção e minimização dos custos de produção.



Figura 2.5: Produção de um recipiente numa linha totalmente automatizada (Marko Beric 2010)

Células de máquinas: é uma série de máquinas de produção e estações de trabalho operadas manualmente, muitas vezes configuradas em forma de “U”, que realizam uma sequência de operações numa família de peças ou produtos que são similares mas não idênticas – é também conhecida como manufatura celular (Figura 2.6).



Figura 2.6: Nesta figura pode-se ver um conjunto de máquinas aglomeradas em formato de célula (BCAG 1997).

Sistemas de manufatura flexíveis: Consistem numa célula de máquinas altamente automatizada que produz famílias de peças ou produtos (Figura 2.7). Este tipo de sistemas caracteriza-se pela capacidade de produzir uma grande variedade de produtos homogêneos (veículos, por exemplo) e por serem bastante flexíveis.



Figura 2.7: Exemplo de sistema de manufatura flexível (KUKA 2011).

2.1.3 Tipos de máquinas

Os sistemas de manufatura referidos anteriormente são constituídos na sua maioria por diversos tipos de máquinas. Estas máquinas podem ser classificadas segundo a participação do operador no processo de manufatura (Groover 2008). Assim sendo, as máquinas podem ser classificadas como: operadas manualmente; semiautomáticas e completamente automáticas:

Operadas manualmente: Controladas ou supervisionadas por um trabalhador ou operador; há uma clara divisão de tarefas, onde a máquina fornece a potência necessária para a operação, enquanto que o trabalhador controla a máquina (Figura 2.8). Ferramentas convencionais, como prensas e máquinas perfurantes encaixam nesta categoria. O operador tem que controlar a máquina continuamente durante todo o ciclo de trabalho.

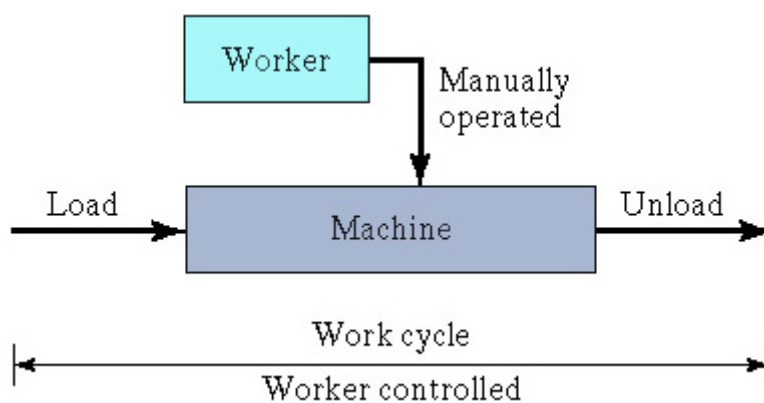


Figura 2.8: Esquema de trabalho de uma máquina operada manualmente – a máquina recebe a matéria a ser trabalhada, o trabalhador opera a máquina e no final do ciclo o produto é descarregado. (Groover 2008)

Máquinas semiautomáticas: estas efetuam uma parte do trabalho sob um programa de controlo, sendo que durante o resto do ciclo de trabalho o controlo da máquina é feito pelo operador humano (Figura 2.9). Um exemplo de uma máquina deste tipo é um torno CNC, onde a máquina processa as suas operações de processamento tal como foi programada, sendo que o operador irá carregar e descarregar a máquina para o próximo ciclo de trabalho. O trabalhador terá que controlar a máquina em cada ciclo, mas não é necessária a sua presença contínua.

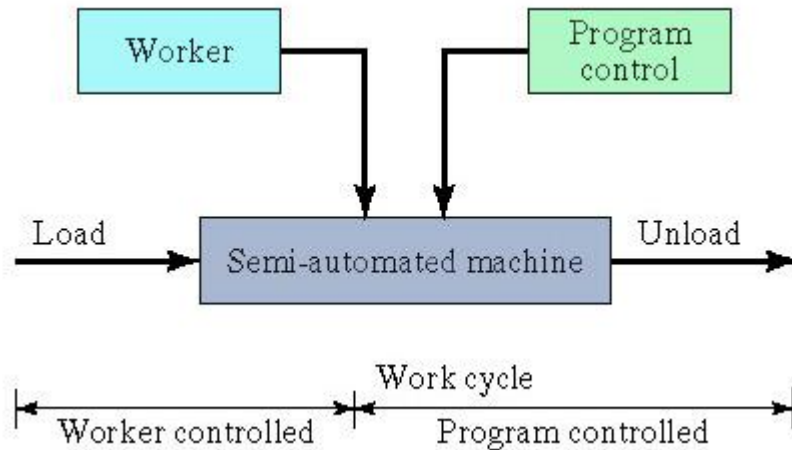


Figura 2.9: Esquema de trabalho de uma máquina semiautomática – Nesta configuração o trabalhador humano usa um programa de controlo para operar a máquina. A sua presença só é necessária no início e no final de cada ciclo, sendo que a máquina opera de forma autónoma no resto do tempo. (Groover 2008)

Máquinas completamente automáticas: Estas máquinas têm a capacidade para operar sem nenhum operador humano presente por períodos de tempo superiores ao ciclo de trabalho da máquina. É contudo necessário um controlo periódico da máquina, quer seja para fornecê-la de matérias-primas ou para efetuar manutenção (Figura 2.10).

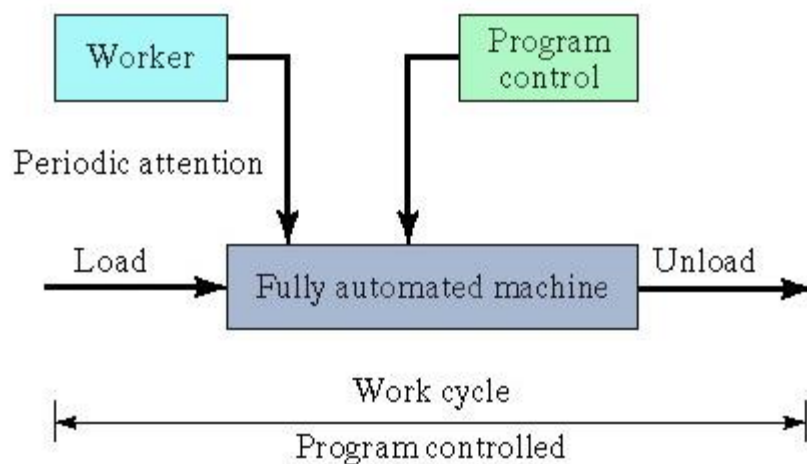


Figura 2.10: Esquema de uma máquina automática – Neste caso um programa de controlo autónomo é responsável por controlar as operações da máquina. O trabalhador humano apenas terá que prestar assistência de forma periódica. (Groover 2008)

2.1.4 Sistema de gestão de material

Um aspeto importante dos sistemas de manufatura sintetizados anteriormente passa pelos sistemas de gestão de material. Na maior parte das operações de montagem a gestão de material pode ser distinguida nos seguintes passos: carregar as unidades de trabalho em cada estação; descarregar as estações de trabalho depois destas processarem os materiais; transportar unidades de trabalho entre estações; e efetuar armazenamento temporário caso seja necessário.

Normalmente existem dois tipos de transporte: **rotas fixas** e **rotas variáveis** (Figura 2.11) (Groover 2008). Nas rotas fixas é usada sempre a mesma sequência de estações de trabalho para processar unidades de trabalho idênticas; nas rotas variáveis as unidades de trabalho são transportadas ao longo de diferentes sequências de estações para permitir um processamento variável nas unidades de trabalho transportadas. Em ambas as rotas é dada ênfase a diferente equipamento de manuseamento de material. As **rotas fixas** baseiam-se em tapetes em movimento contínuo, ao longo do qual existem diversas estações de trabalho para operar sobre as peças.

As **rotas variáveis** usam AGV (*Automated Guided Vehicle*) e sistemas monocarril, seguindo estes uma rota diferente consoante a peça que transportam, de modo a otimizar todo o processo de manufatura.

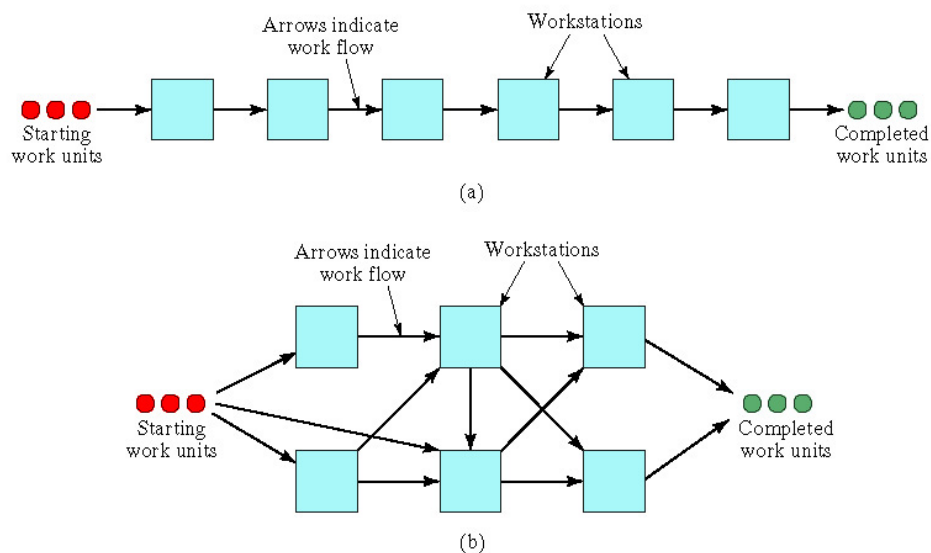


Figura 2.11: Diferentes tipos de rotas usadas na gestão do material nos sistemas de manufatura: Na figura (a) pode ver-se um sistema baseado numa rota fixa, enquanto que na figura (b) pode ver-se um sistema baseado numa rota variável. (Groover 2008)

2.1.5 Sistemas de controlo por computador

Os sistemas controlados por computador são uma parte fundamental dos sistemas de manufatura automatizada, uma vez que estes são necessários para controlar os equipamentos semiautomatizados e totalmente automatizados e participam numa coordenação global e na gestão do sistema de manufatura (Groover 2008).

As funções mais genéricas dos sistemas de controlo por computador são:

- **Comunicar instruções aos trabalhadores:** os operadores devem receber as instruções de trabalho apropriadas para desempenharem as suas funções nas estações de trabalho manuais.
- **Descarregar programas de trabalho:** todas as máquinas controladas por computador são operadas por programas que são enviados por sistemas computadorizados.
- **Controlo dos sistemas de gestão de material:** a disponibilidade de estações de trabalho e dos sistemas de gestão de material devem ser coordenados -para que o processamento das estações de trabalho seja eficiente.
- **Escalonamento de produção:** certas funções de agendamento de produção devem ser cumpridas no local do sistema de manufatura.
- **Diagnóstico de falhas:** diagnóstico de equipamento, de preparação do agendamento de manutenção preventiva e na manutenção do inventário de peças sobresselentes.
- **Monitorização de segurança:** a manutenção do sistema para que este opere em segurança.
- **Manutenção de controlo de qualidade:** envolve a deteção e rejeição de estações de trabalho que sejam efetivas num ambiente de trabalho.
- **Gestão de operações:** gestão geral de operações do sistema de manufatura, seja através do controlo de supervisão ou indiretamente através de relatórios de produção periódicos.

2.1.6 Recursos humanos

Os recursos humanos também têm um papel relevante nos sistemas de manufatura, nem que seja num papel de supervisão global do sistema. Em casos onde o operador humano desempenha algum trabalho de valor acrescentado, este é designado como trabalho direto. Os operadores humanos são também necessários para gerir e dar suporte ao sistema (programadores); operar e dirigir atividades computadorizadas; manter e reparar os sistemas de manufatura automatizados e caso seja necessário, realizar outras tarefas indiretas.

2.1.7 Critérios de classificação de sistemas de manufatura

Segundo (Groover 2008), os sistemas de manufatura podem ser classificados consoante o tipo de operações desempenhadas, o número de estações de trabalho, o *layout* do sistema, o nível de automação e a variedade de componentes e produtos.

Relativamente ao tipo de operações desempenhadas, os sistemas podem ser classificados consoante um certo tipo de características:

- **Tipo de material processado:** diferentes materiais requerem diferentes métodos de processamento.
- **Tamanho ou peso da peça ou do produto:** tem uma grande influência no tipo, alcance e escala do equipamento de manufatura escolhido para processar o item.

- **Complexidade da peça ou produto:** a complexidade do componente está correlacionada com o número de operações de processamento requeridas. Já a complexidade do produto refere-se ao número de componentes que devem ser montados.
- **Geometria do componente:** componentes “maquinados” são rotativos ou não rotativos, o que tem um efeito significativo nas operações de processamento que podem ser efetuadas nos componentes, além de que os sistemas de gestão do material devem ser desenhados de uma forma que se adaptem as partes rotativas e não-rotativas.

Relativamente ao número de estações de trabalho: o número de estações de trabalho exerce uma forte influência no desempenho dos sistemas de manufatura, quer em termos de carga de trabalho, capacidade, taxa de produção e fiabilidade.

Relativamente ao layout do sistema: a configuração do sistema ou o *layout* das estações de trabalho do sistema de manufatura são um fator importante. As estações de trabalho para rotas fixas são normalmente arranjadas de forma linear (e.g. uma linha de produção) enquanto que rotas variáveis os *layouts* podem ter múltiplas configurações.

Relativamente aos níveis de manuseamento e automação: nos sistemas de manufatura os níveis de manuseamento dividem-se em dois níveis para as estações simples de trabalho (manuseado e automatizado) e em três níveis para sistemas multiestação (manuseado, automatizado e híbrido).

Relativamente à variedade de componentes ou produtos: os sistemas de manufatura podem ser afetados pelas diversas propriedades dos materiais processados, seja o tipo de material, a cor, a forma; os diversos componentes eletrónicos colocados nos circuitos; o tamanho dos circuitos. Toda esta variedade pode ser descrita através de três modelos: modelo singular, modelo *batch* e modelo misto (Figura 2.12):

- **Modelo singular:** todas as partes/produtos produzidos são idênticos, o que significa que o sistema de produção é totalmente dedicado à produção do item em questão. O equipamento é especializado e desenhado para uma eficiência máxima.
- **Modelo *batch*:** são fabricadas diferentes componentes ou produtos, mas são feitos em “lotes” (*batches*), sendo assim necessário algumas alterações na configuração física, bem como reprogramações aquando da troca de modelos.
- **Modelo misto:** são produzidos diferentes componentes/produtos pelo sistema de manufatura, mas não existem diferenças significativas entre os componentes produzidos. O sistema é capaz de gerir as mudanças de fabrico sem ser necessário uma reconfiguração e uma reprogramação de todo o sistema. A mistura é assim produzida continuamente e não em lotes.

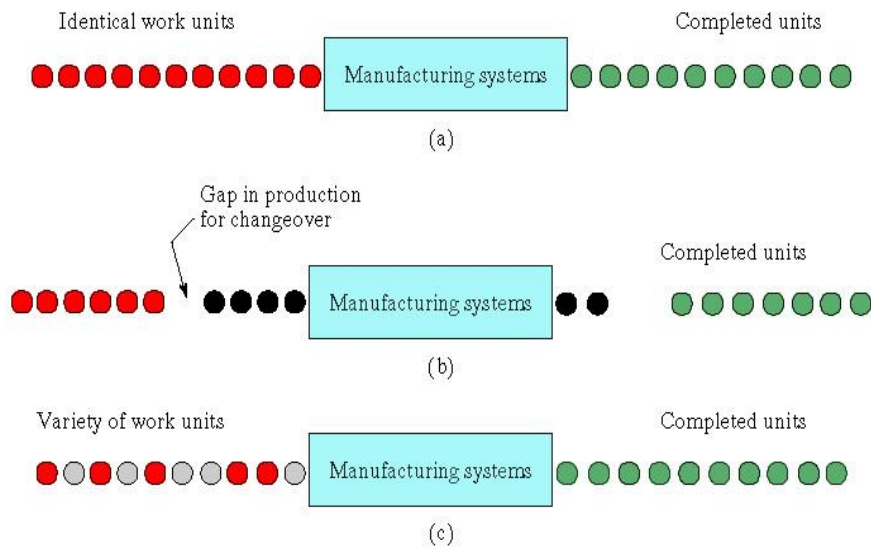


Figura 2.12: Diferentes processos de tratamento de componentes: Na figura (a) pode ver-se o modelo singular; na figura (b) pode ver-se um modelo *batch*; na figura (c) pode ver-se um modelo misto. (Groover 2008)

2.1.8 Paradigmas de manufatura

Os sistemas distribuídos de manufatura podem dividir-se em diferentes paradigmas de manufatura (Chituc & Restivo 2009), consoante os métodos e a tecnologia utilizada:

- **Planeamento de produção centralizado:** este paradigma baseia-se na comunicação centralizada; os sistemas que assentam neste paradigma são rígidos e têm lacunas ao nível da escalabilidade e robustez, tendo um grande custo de integração. Os sistemas de produção em massa dão ênfase à redução dos custos dos produtos e a uma utilização plena da capacidade da fábrica.
- **Projeto Assistido por Computador/Manufatura Assistido por Computador:** estes sistemas integram diferentes ferramentas (tais como sistemas multimédia e desenho a três dimensões) num ambiente multimédia através da Internet. No projeto assistido por computador, os computadores são usados no desenho e análise de produtos e processos. Na manufatura assistido por computador, os computadores são usados diretamente no controlo e monitorização de máquinas/processos em tempo real ou em modo *offline* de forma a dar suporte a operações de manufatura.
- **Manufatura integrada por Computador:** estes sistemas são caracterizados pelo uso de maquinaria controlada por computador e pelo uso de sistemas de automação. Este tipo de manufatura combina diversas tecnologias como projeto assistido por computador e manufatura assistida por computador para providenciar processos de manufatura com poucas ou nenhuma falhas, automatizando tarefas repetitivas e reduzindo assim os níveis de trabalho manual. Esta abordagem aumenta a rapidez dos processos de manufatura.

ra, usando para tal sensores capazes de recolher informação em tempo real, assim como diferentes processos de controlo de forma a manter os sistemas de manufatura otimizados.

- **Arquitetura *lean* e ágil:** os sistemas de manufatura assentes no conceito *lean* são focados no melhoramento contínuo da qualidade do produto enquanto são reduzidos os custos de produção. Os sistemas *flexíveis* dão ênfase à manufatura de uma grande variedade de produtos usando o mesmo sistema. Os sistemas ágeis têm como preocupação principal a capacidade de dar uma resposta rápida às mudanças no sistema.
- **Sistemas reconfiguráveis:** estes sistemas de manufatura caracterizam-se pela sua capacidade de ajustarem rapidamente as suas funcionalidades de produção em resposta a novas circunstâncias no ambiente de trabalho (como por exemplo, alterações ao nível da procura de produto, produção de um novo produto num sistema já existente e a integração de uma nova tecnologia de processamento num sistema de manufatura já existente).
- **Sistemas de manufatura inteligentes:** as capacidades destes sistemas são melhoradas com capacidades que se assemelham ao comportamento humano (como por exemplo, a capacidade de tomar decisões como se de um operador humano se tratasse).
- **Sistemas de manufatura holónicos:** um holon é uma parte identificável de um sistema com uma identidade única e consiste em componentes mais pequenos pertencentes a um sistema maior. Nos sistemas de manufatura holónicos, um holon – que na sua definição filosófica pode ser definido como o conceito de que o todo será maior que a soma das suas partes - é um bloco autónomo e cooperativo capaz de transformar, transportar, guardar e validar informação e objetos físicos. Estes sistemas ainda não têm grande aceitação na indústria uma vez que existe uma falta de *standards* aceites por uma maioria; a sua implementação ainda não cobre todo o ambiente de manufatura e porque o conceito de uma decisão centralizada é difícil de se aceitar.
- **Sistemas de manufatura automaticamente equilibrados:** estes sistemas são caracterizados por uma mistura equilibrada de atividades humanas e autómatos. Estes sistemas são desenhados considerando o contexto socioeconómico onde se inserem bem como outros fatores tais como o custo de produção dos componentes, o nível necessário de flexibilidade e a qualidade do produto desejada.
- **Sistemas de manufatura inspirados biologicamente:** estes sistemas foram criados para tentar resolver os problemas relacionados com os ambientes de manufatura baseados em abordagens “biológicas”, tentando melhorar os aspetos de auto-organização, aprendizagem, evolução e adaptação. Estes sistemas adaptam-se facilmente a mudanças imprevistas nos ambientes de manufatura e são capazes de realizar as suas tarefas com sucesso mesmo que esta não esteja bem definida, conseguindo assim atingir um comportamento global satisfatório através da interação entre os diversos componentes do sistema. Nestes sistemas não existe um sistema “central” mas sim um sistema alternativo onde existem vários subsistemas para resolver um dado problema, selecionando a me-

lhora opção através da seleção natural. Nestes sistemas as tecnologias mais usadas são agentes, computação bio inspirada, computação evolucionária, bio robótica.

- **Sistemas de manufatura *Wise*:** estes sistemas têm a capacidade de resolver problemas e tomar decisões tendo em conta os valores humanos. São sistemas inteligentes, capazes de se automelhorarem (por exemplo, capazes de monitorizar o seu desempenho económico e o seu desempenho de manufatura, diagnosticar problemas de desempenho e tomar decisões necessárias); são autoadaptáveis (capazes de detetar anomalias e tomar decisões para recuperar destas) e autocuráveis (capazes de se automonitorizar, diagnosticar as causas de falhas e recuperar das mesmas, podendo esta abordagem ser feita a uma pequena unidade de processamento ou a uma rede global de manufatura).

2.2 Alguns mecanismos de interoperabilidade

2.2.1 *Web Services*

Os *web services* podem ser genericamente definidos como uma aplicação que está acessível a outras aplicações distintas através da Internet. Sendo uma definição bastante ampla, um programa acessível através da Internet com uma API (*Application Programming Interface*) estável pode ser considerado um *web service*.

Numa definição mais concreta, um *web service* pode ser definido com uma aplicação autónoma, modular, que tem uma *interface* aberta e está orientada à Internet (Alonso et al. 2004). O consórcio World Wide Web define um *web service* como uma aplicação de *software* identificada por um URI (*Uniform Resource Identifier*), onde as suas interfaces e ligações são capazes de estar definidas, descritas e descobertas como “artefactos” XML (*Extensible Markup Language*). Um *web service* suporta interações diretas com outros agentes de software ao usar mensagens baseadas em XML trocadas através de protocolos Internet.

Existem duas classes principais de *web services*, os serviços REST (*Representational State Transfer*) e SOAP (*Simple Object Access Protocol*):

2.2.2 Serviços REST

Os serviços REST constituem a arquitetura de *software* da Internet como a conhecemos (Pautasso 2008) (Figura 2.13). Estes serviços oferecem um conjunto de características pelas quais as arquiteturas se devem reger de forma a que mantenham um desempenho elevado.

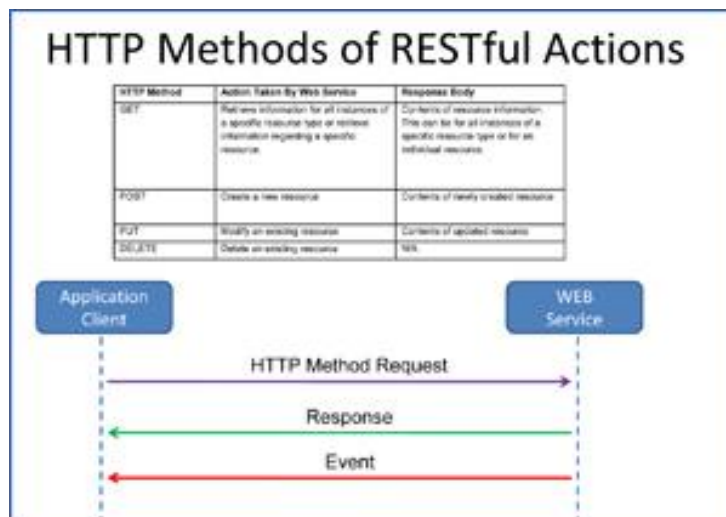


Figura 2.13: Exemplo de *Web Service* do tipo REST (Paula Bernier 2010)

As características dos serviços REST são as seguintes (Amaral 2015):

- **Uma interface uniforme:** onde recursos individuais são identificados nos pedidos através de endereços URI; Cada mensagem possui informação suficiente para descrever como processar a mensagem (em cabeçalhos HTTP (*Hypertext Transfer Protocol*) por exemplo)
- **Não têm estado:** o estado necessário para gerir o pedido está contido dentro do pedido em si; um endereço URI identifica o recurso e o corpo da mensagem contém o estado
- **São cacheable:** os clientes podem “esconder” respostas; as respostas devem definir-se como *cacheable*
- **Distinção entre cliente e servidor:** existe uma interface uniforme que separa clientes de servidores; os servidores e os clientes podem também ser substituídos e desenvolvidos independentemente, desde que a interface não seja alterada.
- **Sistema dividido em camadas:** os clientes não podem dizer se estão ligados diretamente a um servidor final, ou se existe um servidor intermediário pelo caminho. Esta característica pode melhorar a escalabilidade, através do balanceamento de carga.
- **Code on demand:** os servidores têm a capacidade de estender temporariamente ou customizar as funcionalidades do cliente ao transferirem alguma lógica para esse cliente, de modo a que ele próprio execute essa lógica (como por exemplo, *scripts* de JavaScript).

2.2.3 Serviços SOAP

Segundo (Suda 2003), os serviços SOAP podem ser definidos como seguindo um paradigma de troca de mensagens *stateless*, unidirecionais entre nós SOAP (Figura 2.14). Este protocolo é capaz de combinar trocas unidirecionais com características inerentes ao protocolo de transporte que serve de base à comunicação, conseguindo-se assim interações mais complexas entre nós, tais como serviços pergunta-resposta e pergunta-resposta múltipla.

O SOAP é um protocolo leve que é independente da plataforma, do meio de transporte e do sistema operativo, uma vez que é construído através do uso de sistemas como o protocolo HTTP e da linguagem XML.

No SOAP existem dois tipos de pedidos: pedidos RPC e pedidos de documentos. Nos pedidos RPC usa-se uma função localizada numa máquina remota como se esta fosse uma função local. Este processo é muitas vezes síncrono – o cliente envia uma mensagem e espera por uma resposta do servidor ou por uma mensagem de erro.

Nos pedidos de documento, é transmitido entre o cliente e o servidor um documento completo escrito na linguagem XML, documento este que está localizado no corpo da mensagem SOAP.

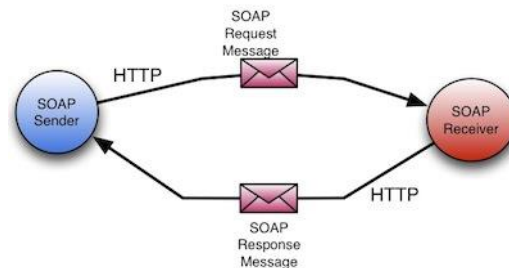


Figura 2.14: Exemplo de um *Web Service* do tipo SOAP (Dasun Hegoda 2014)

2.2.4 Motivação para o uso de Web Services

Uma vez que na área dos sistemas de manufatura existem algumas dificuldades ao nível da integração dos vários tipos de sistemas em jogo, bem como na automação de processos dentro destes sistemas, o uso de *web services* pode ser benéfico na interligação destes sistemas.

Mesmo que a maioria dos processos intra-empresariais estejam automatizados na maioria dos casos (Alonso et al. 2004), os processos inter-empresariais ainda são muitas vezes desenvolvidos de forma manual, sendo necessário o uso de páginas *Web* – quer seja através de *e-mail* ou preenchimento de formulários – para realizar estes processos. (Figura 2.15).

Nos dias de hoje a maioria dos sistemas de manufatura estão espalhados pelo globo, muitas vezes dependentes de companhias diferentes. No cenário apresentado em seguida uma empresa (no lugar de cliente) precisa de encomendar bens doutra empresa (um fornecedor). Este fornecedor irá processar a encomenda, entregando-a diretamente (caso tenha o bem em *stock*) ou pedindo a um terceiro interveniente que entregue o bem. Para todos os intervenientes em jogo, seria bastante benéfico se todo o sistema fosse automatizado, desde o pedido ao pagamento (Figura 2.16).

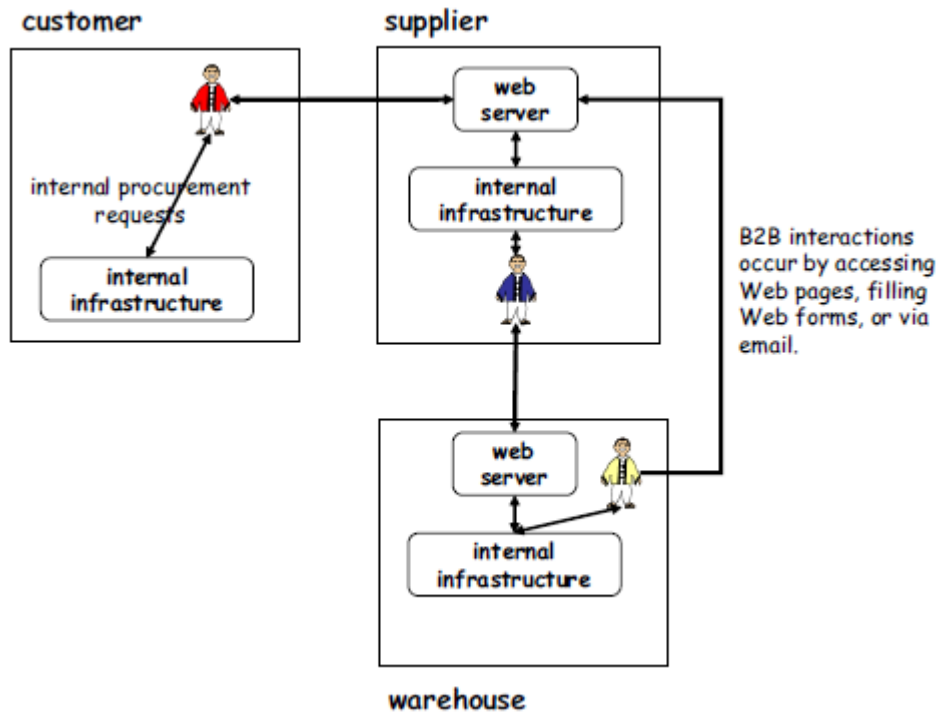


Figura 2.15: Esquema de processos intra e inter-empresariais – No caso apresentado, a comunicação entre o fornecedor e o armazém ainda é feita através de formulários Web e correio eletrônico (Alonso et al. 2004).

O uso de *web services* propõe três tipos de abordagem para resolver este problema: paradigma orientado serviços; protocolos *middleware* e standardização (Alonso et al. 2004).

- **Paradigma orientado a serviços:** neste caso os *Web Services* trabalham assumindo que as funções de uma empresa estão disponíveis e estão expostas como um serviço. Este serviço pode ser um procedimento, um método ou um objeto. A invocação deste serviço é feita por um programa. Com o uso de *Web Services* será possível invocar estes serviços através da Internet. Neste caso os *Web Services* assumem que os serviços têm pouca ligação entre si, uma vez que estes são desenvolvidos e geridos por empresas diferentes.
- **Protocolos *middleware*:** é necessário redesenhar os protocolos *middleware* de forma a que possam trabalhar de forma P2P entre diversas companhias. Os protocolos *middleware* convencionais (como o 2PC – *Two-phase Commit Protocol*) foram desenhados de modo a que não desempenhem interações entre várias empresas – eles assumem um papel de coordenador central de transações. Como este coordenador central pode levantar questões de segurança e confidencialidade, os protocolos 2PC deverão ser desenhados de forma a que exista uma interação direta entre as empresas.
- **Standardização:** uma vez que as interações ocorrem a nível global, a standardização não só é benéfica como é necessária. Os *Web Services* são amplamente aceites e têm bastante sucesso na “ativação” de interações entre operadores humanos e aplicações

(através de *Web Browsers* e *Web Servers*). Assim sendo, torna-se normal que a tecnologia de integração use a web para proceder à standardização.

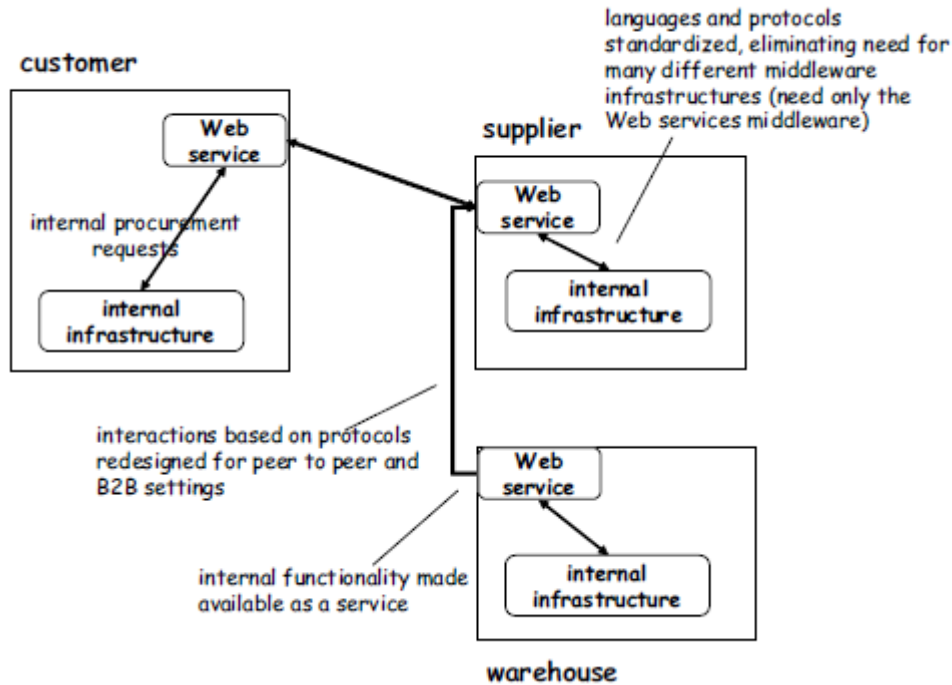


Figura 2.16: Esquema de processos intra e inter-empresariais usando web services – Neste caso as linguagens e os protocolos de comunicação são standardizados, não sendo necessário usar protocolos middleware para “traduzir” a informação entre os diversos intervenientes (Alonso et al. 2004).

2.3 Elementos de simulação

Segundo (Law & Kelton 2000), o conceito de simulação pode ser definido como um conjunto de métodos e aplicações utilizados para imitar o comportamento de sistemas reais. No processo de simulação é utilizado um modelo representativo de um sistema teórico ou de um sistema existente para que possa ser estudado o comportamento destes sistemas (Schwendimann 2010).

Seja de forma manual ou feito através de meios informáticos, a simulação envolve a criação de uma história artificial de um sistema e a observação dessa história artificial serve para inferir sobre as características operacionais do sistema real. (Banks et al. 2004).

Segundo (Maria 1997), uma simulação pode ser descrita como a operação sobre um modelo de um sistema. Este modelo pode ser reconfigurado e experimentado diversas vezes; normalmente isto é impossível de se fazer no sistema em estudo, quer pelo facto de ser muito dispendioso, quer pelo facto de não ser prático.

Um estudo de simulação, especialmente no desenvolvimento de Sistemas Distribuídos de Manufatura, reveste-se de especial importância principalmente na fase de estudo das interações entre as diversas entidades do sistema, na determinação das consequências que um nó pode provocar no resto do sistema, na fase de planeamento logístico do sistema, assim como para determinar a eficiência e a produtividade do sistema a ser desenvolvido (Nagurney 2012).

2.3.1 Processo de simulação

Como referido anteriormente, o processo de simulação assenta na criação de um modelo representativo do comportamento de um sistema. Antes de se criar esse modelo, é necessário analisar previamente o problema e seguir alguns passos de forma a delinear uma estratégia para simular o sistema (Padhya 2013). Estes passos podem ser definidos da seguinte forma:

- Em primeiro lugar, é necessário identificar qual o problema em questão.
- Depois de identificado o problema, é necessário especificar o modelo de simulação apropriado
- Em seguida obtém-se e processa-se dados reais do sistema em estudo e valida-se o modelo de simulação criado.
- Por fim, escolhe-se a abordagem apropriada para simular o sistema em estudo, estabelecem-se os parâmetros de controlo de cada simulação e interpretam-se os resultados obtidos.

2.3.2 Noção de modelo

Segundo (Schwendimann 2010), um modelo pode ser descrito como uma representação de algo – um sistema, um fenómeno ou um mecanismo (Figura 2.17). Apesar de ser mais simplista, ignorando certos detalhes, um modelo é semelhante àquilo que um sistema representa, tentando aproximar da melhor forma possível as características mais importantes de um sistema, e abstraído-se das características irrelevantes.

De uma forma mais detalhada, um modelo é uma representação gráfica, física, matemática ou verbal de um conceito, fenómeno, estrutura ou sistema do mundo real (BusinessDictionary 2016), que tem como objetivos principais a facilitação da compreensão de um sistema ao eliminar componentes desnecessários do mesmo; ajudar na tomada de decisões ao simular cenários “e se”; explicar, controlar e prever eventos baseando-se em observações prévias.

Por norma, o valor de um modelo é diretamente proporcional à qualidade da sua representação de um sistema do mundo real. No entanto, um modelo de um conceito não necessita de uma correspondência exata para ser considerado um bom modelo (Gregory 1998).

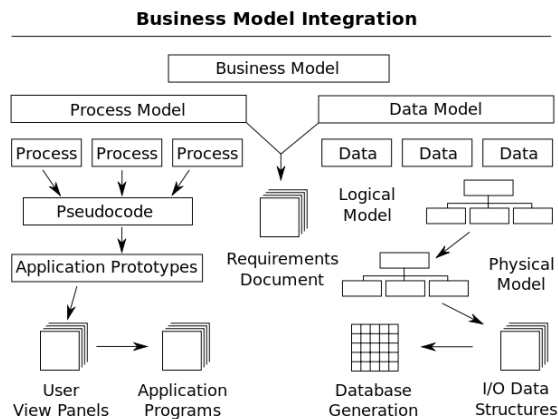


Figura 2.17: Representação genérica de um modelo de negócio (Smith & Sarfaty 1993)

2.3.3 Modelo de simulação

De acordo com (Maria 1997), um modelo de simulação pode ser descrito como um modelo matemático que tem como finalidade a simulação de um sistema do mundo real, sendo que este modelo pode ser desenvolvido com a ajuda de um pacote de simulação.

Este modelo matemático pode utilizar cálculo diferencial, teoria das probabilidades, métodos algébricos ou outras técnicas matemáticas (Banks et al. 2004). Estes modelos usam um ou vários parâmetros numéricos que servem para medir o desempenho do sistema a ser simulado.

Estes modelos podem ser classificados como determinísticos (onde os parâmetros de entrada e de saída têm valores fixos), estocásticos (onde pelo menos um dos parâmetros de entrada ou saída é probabilístico), estático (onde o tempo não influencia o sistema), dinâmico (onde as interações ao longo do tempo são tidas em conta). Normalmente, os modelos de simulação são estocásticos e dinâmicos (Maria 1997).

No entanto, estes modelos matemáticos podem não ser suficientes para simular alguns sistemas do mundo real. Nestes casos, recorre-se a pacotes de simulação informáticos para criar modelos de simulação apropriados ao sistema em estudo. A simulação realizada por estes pacotes gera dados de saída que são usados para estimar as medidas de desempenho do sistema real em estudo. (Banks et al. 2004).

Segundo (Maria 1997), a criação de um modelo de simulação necessita de alguns passos tais como a identificação e formulação do problema, obtenção e processamento dos dados do sistema real em estudo, formulação e desenvolvimento do modelo, validação do modelo desenvolvido, documentação do modelo para utilização futura, escolha dos parâmetros de teste de cada execução da simulação, realização da simulação e interpretação dos resultados obtidos.

Na figura 2.18 serão apresentados alguns modelos de simulação:

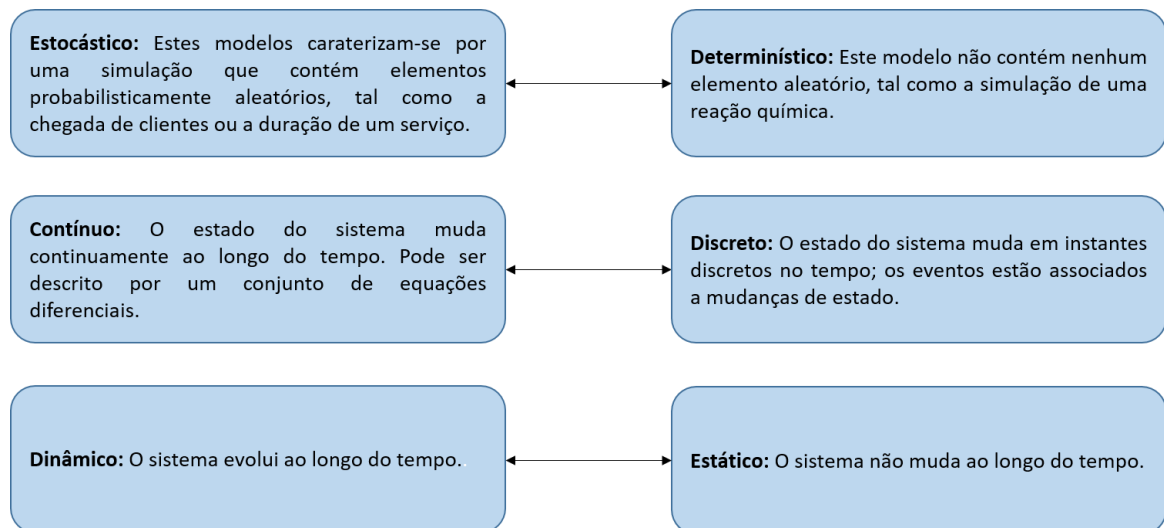


Figura 2.18: Tipos de modelo de simulação (Maria 1997)

Para desenvolver os modelos de simulação supracitados podem-se usar mecanismos como redes de Petri, *Finite State Machines*, linguagens de programação como o Prolog e lógica difusa.

Alguns dos aspetos que motivam a utilização da simulação (Meng 2002):

- A simulação, como metodologia utilizada para a otimização de processos e prevenção de falhas, reveste-se de grande utilidade.
- O comportamento do sistema e os seus efeitos podem ser observados e modelados
- Podem ser simuladas mudanças a nível ambiental, organizacional e informativo, sendo possível observar as mudanças em tais variáveis
- As observações baseadas em simulações fornecem um grande insight do comportamento do sistema, sendo possível determinar quais as variáveis mais importantes, bem como estas interagem entre si
- O modelo pode ser validado e os resultados da simulação podem ser verificados
- A simulação permite experimentar novos *designs* e estratégias e políticas antes de se desenvolver, mudar e melhorar o sistema
- Pode ser usado para treinar um sistema sem ter que se gastar recursos associados ao uso do sistema (como por exemplo, ter que parar o trabalho em progresso)
- Um sistema pode ser tão complexo que o seu comportamento só pode ser estudado através da simulação

No entanto, a simulação pode não ser apropriada nas seguintes situações (Prensky 2002):

- Quando o problema pode ser resolvido com senso comum
- Quando existe uma solução analítica “simples”
- Quando é mais fácil efetuar medidas diretamente no sistema físico
- Quando a obtenção dos dados requeridos demora demasiado tempo
- Quando existe uma falta de informação ao nível de modelação do sistema
- Quando o sistema não pode ser validado e verificado
- Quando existem expectativas irrealistas para os resultados da simulação

- Quando a simulação é demasiado dispendiosa ou quando o sistema é demasiado complexo para ser modelado

2.3.4 Paradigmas de simulação paralela e distribuída

Segundo (Fujimoto 2001), existem dois grandes paradigmas de simulação: a simulação paralela e a simulação distribuída. A simulação paralela (discreta) preocupa-se com a execução de tarefas em plataformas multiprocessador que interagem frequentemente, isto é, milhares de vezes por segundo. Na simulação distribuída a simulação distribuída preocupa-se com a execução de simulações em sistemas sem grande ligação, onde as interações demoram muito mais tempo (podem demorar mais do que um milissegundo) e ocorrem com muito menos frequência. Este conceito de simulação inclui a execução de tarefas em computadores distribuídos globalmente, estando estes computadores interligados através de uma rede de grande alcance (como a Internet, por exemplo).

Quanto à utilização dos modelos de simulação, existem duas categorias principais de simulação – as que são usadas para análise (como para avaliar designs ou processos alternativos de controlo de um sistema complexo, como a gestão de rotas aéreas). Neste caso, o objetivo principal consiste numa obtenção dos dados da simulação o mais rapidamente possível de forma a poder melhorar a eficiência da ferramenta de simulação.

A segunda categoria de simulação consiste na criação de ambientes virtuais nos quais operadores humanos e/ou *hardware* possam ser incorporados.

Estes ambientes são amplamente usados em ambientes de treino (tais como simuladores de voo) ou entretenimento (como videojogos).

Estes ambientes virtuais são também usados de forma extensiva pela indústria e pessoal militar, proporcionando assim uma abordagem mais segura e menos onerosa do que exercícios de campo.

Apesar das diferenças supracitadas entre as simulações paralelas e distribuídas, estas também apresentam algumas vantagens comuns (Fujimoto 2001):

- Os tempos de execução de simulações analíticas podem ser reduzidos ao dividir-se uma simulação de um sistema complexo em diversas simulações de sistemas mais simples.
- Em certos casos, torna-se necessário uma simulação *online* de forma a obter resultados válidos, com o intuito de tomar decisões em tempo útil. Neste caso a simulação paralela fornece os meios necessários para a redução do tempo de simulação, garantindo assim resultados proveitosos.
- A simulação utilizada em ambientes virtuais deve ser executada em tempo real, isto é, o simulador deve ser capaz de simular uma atividade secundária num tempo ínfimo, garantindo assim que o ambiente de simulação se mantém realista e que evolui da mesma forma que o ambiente real. Distribuir a execução da simulação por diversos processadores pode ser útil para desempenhar esta tarefa. Num cenário ideal, uma execução *escalável* pode ser obtida enquanto a simulação distribuída continua a ser executada em tempo real enquanto o ambiente virtual continua a ser simulado e o número de núcleos de processamento continua a ser aumentado.
- As técnicas de simulação distribuída podem ser usadas para criar ambientes virtuais de simulação que estão dispersos geograficamente, tornando assim possível que um opera-

do humano e/ou os dispositivos que este opera possa interagir como se estivessem todos no mesmo local. Estes ambientes virtuais de simulação distribuída apresentam vantagens óbvias em termos de conveniência e redução de custos de deslocamento.

- A simulação distribuída apresenta vantagens a nível da integração de simuladores de máquinas de diferentes fabricantes. Tomando como exemplo um simulador de voo, existem diferentes simuladores para diferentes aeronaves que podem ter sido desenvolvidos em arquiteturas diferentes. A simulação distribuída pode reduzir os custos de integração destes simuladores ao fazer com que cada simulador corra num computador dedicado, criando um novo ambiente virtual com as características de cada simulador.
- Outra vantagem associada à utilização de múltiplos processadores consiste num aumento significativo da tolerância a falhas. Caso falhe um processador, é possível que os outros processadores continuem a simulação, desde que os processadores em falha não possuam elementos críticos à simulação.

2.3.5 Simulação cooperativa

A simulação cooperativa é uma abordagem de simulação onde cada um dos intervenientes do sistema simula uma parte distinta do sistema global e onde é feita uma troca periódica de informação em tempo real, sendo esta troca de informação coordenada através da utilização de *software* adicional, muitas vezes chamado de *middleware*.

Este tipo de simulação é uma abordagem a ter em conta para se efetuarem simulações de sistemas dinâmicos complexos, uma vez que não existe uma plataforma de simulação disponível que seja capaz de simular um sistema com um grande nível de complexidade e de detalhe, tal como no caso dos sistemas distribuídos de manufatura.

Existem algumas vantagens inerentes à utilização da simulação cooperativa, tais como a capacidade de se poder simular em separado cada um dos subsistemas do modelo de simulação, sendo possível aplicar algoritmos e especificações de forma individual, resultando assim numa simulação mais eficiente em termos de tempo (Bleicher et al. 2014).

A principal desvantagem da simulação cooperativa está relacionada com combinação de modelos de simulação com intervalos de comunicação periódicos. Esta combinação gera erros numéricos, uma vez que os dados são obtidos através da extrapolação de conjuntos de dados realizada entre pontos de troca de informação.

2.3.6 Simulação por computador

Atualmente, os estudos de simulação são naturalmente efetuados com o auxílio de computadores. Neste tipo de simulação, um programa de computador modela o comportamento de um sistema ao longo do tempo. Nestes modelos de simulação as variáveis de estado representam o estado atual do sistema em cada instante, sendo que, o programa vai modificando as variáveis de estado enquanto decorre uma simulação.

A simulação pode ser usada nas mais diversas áreas tais como redes de telecomunicações, sistemas de transportes, sistemas eletrónicos, simulação de batalhas, sistemas ecológicos, sistemas de manufatura e logística.

Na tabela 2-1 apresenta-se alguma da terminologia apresentada na definição de simulação:

Tabela 2-1: Terminologia apresentada nos modelos de simulação (Maria 1997)

Nomenclatura	Definição
Sistema	Consiste num conjunto de elementos interligados que interagem uns com os outros de forma a efetuar algum comportamento
Modelo	É uma representação abstrata de um sistema, especificado através de relações lógicas e matemáticas, de forma a descrever a estrutura do sistema e o seu comportamento. O estado e o comportamento é descrito através de variáveis de estado, atributos, eventos e atividades
Estado do sistema	É representado por um conjunto de variáveis, que contêm toda a informação que é necessária para descrever um sistema num dado instante
Entidade	Qualquer objeto ou componente de um sistema, que requer uma representação explícita de um modelo e é usada para definir um estado ou um comportamento de um sistema
Atributos	Representam as propriedades das entidades, como por exemplo, a prioridade de um certo elemento numa fila de espera ou o tempo de processamento de uma dada tarefa
Filas de espera	É uma lista de entidades distribuída de uma forma lógica
Evento	É uma ocorrência instantânea que muda o estado do sistema
Atividade	Carateriza-se pelo tempo em que um sistema permanece num estado
Atrazo	É um intervalo de tempo sem uma duração específica
Clock	Representa o tempo durante uma simulação

Numa simulação discreta existe um *clock* e um agendamento de eventos futuros. O relógio avança até ao tempo do próximo evento, as variáveis de estado são atualizadas e este processo é repetido até ao evento final.

Numa simulação contínua o foco principal está nas atividades e não nos eventos; o comportamento do sistema depende do tempo; as respostas ao sistema envolvem equações diferenciais.

2.3.7 Simulação com agentes

A utilização de agentes permite a modelação de comportamentos autónomos; estes “sentem” o ambiente envolvente, sendo capazes de mudar o estado do ambiente. Têm objetivos a alcançar e podem comunicar e cooperar uns com os outros de modo a alcançar um objetivo global. Podem também ser criados e destruídos dinamicamente

2.4 Simulação em Sistemas Distribuídos de Manufatura

A simulação de sistemas distribuídos de manufatura pode ser considerada como um tipo de simulação composta por múltiplos processos que são executados de forma independente em diferentes locais mas que interagem entre si (Uygun et al. 2009). Estes processos de *software* de simulação podem ser utilizados tanto para modelar uma complexa cadeia de manufatura como para simular o comportamento de uma máquina individual. No entanto, a simulação de sistemas de manufatura de larga escala compõem a maioria das aplicações da simulação.

No contexto económico local, apenas em casos muito pontuais um produto complexo é fabricado num único local por uma única empresa. Normalmente os componentes são fabricados por empresas diferentes e montados em diferentes locais. De forma a otimizar custos a capacidade dos armazéns tende a diminuir, fazendo com que os componentes sejam entregues na altura necessária (não havendo assim compassos de espera), o que aumenta a dependência entre as empresas. Assim sendo, torna-se necessário que os atuais aspetos a serem simulados não abordem só os aspetos de manufatura, mas também os aspetos relacionados com a logística dos componentes necessários nos processos de manufatura. Estes aspetos de simulação de SDM são abordados com mais profundidade no capítulo 3.

Vantagens inerentes

Existem algumas vantagens inerentes à utilização da simulação em sistemas distribuídos de manufatura, entre as quais se destacam a possibilidade de avaliação das diferentes alternativas dos sistemas de manufatura; ter a capacidade de tomar decisões importantes relativas a equipamentos e maquinaria; ser capaz de alocar recursos e reduzir custos de trabalho; otimizar os *layouts* de fábricas; balancear linhas de produção; reduzir os tempos de ciclo de trabalho e o *Work in Progress*; otimizar as cadeias de fornecimento dos sistemas de manufatura; ser possível dimensionar cada um dos nós dos sistemas de manufatura; simular a componente logística de um sistema distribuído de manufatura (Uygun et al. 2009).

No entanto, é preciso ter em conta alguns fatores de forma a que estas vantagens sejam alcançadas, tais como a autonomia e o comportamento de cada interveniente no sistema em estudo; a possibilidade de serem tomadas decisões conflituosas; a ocorrência de distúrbios no ambiente de simulação durante as interações entre os diversos intervenientes; é necessária uma grande componente cooperativa e colaborativa entre os intervenientes no sistema.

2.5 Pacotes de simulação

Um pacote de simulação pode ser definido como um programa que permite ao seu utilizador poder observar uma operação através da simulação sem ter que executar essa mesma operação. Em teoria, qualquer fenómeno que possa ser replicado por um modelo matemático poderá ser simulado por um pacote de simulação. Estes pacotes de simulação são usados no *design* de produtos de forma a que o produto final seja o mais fiel possível às especificações originais, evitando assim mudanças dispendiosas durante o processo de fabrico. Outra das utilizações mais comuns para este tipo de pacotes prende-se com o desenvolvimento de instalações industriais (fábricas químicas, estações de energia, centrais nucleares), onde estes pacotes de simulação são utilizados em tempo real permitindo a recolha de dados importantes sobre as limitações das instalações, evitando assim potenciais acidentes. Nesta secção apresenta-se e descreve-se de forma sucinta alguns pacotes de simulação existentes.

2.5.1 ProModel

É uma ferramenta de simulação de eventos discretos que permite avaliar, planear e desenhar aplicações de manufatura, armazenamento, logística e outras aplicações estratégicas (Figura 2.19).

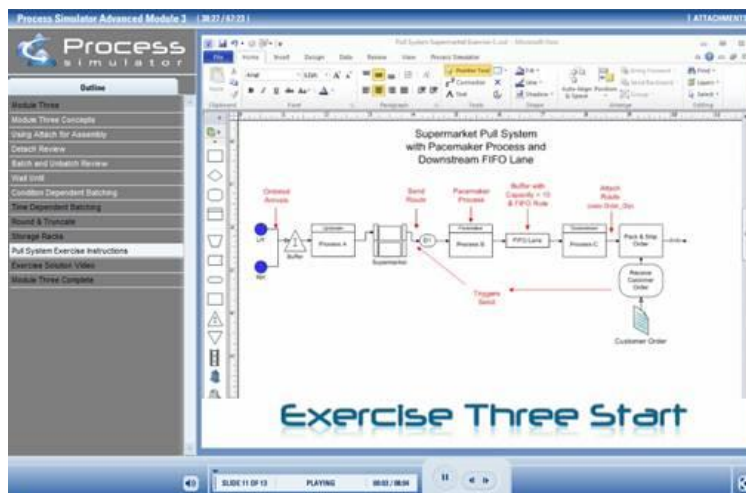


Figura 2.19: Exemplo de simulação com o pacote de simulação ProModel (ProModel 2016)

2.5.2 MODSIM

É uma linguagem de programação estruturada por blocos e que fornece suporte à simulação de eventos discretos e à programação orientada a objetos. É utilizada para construir modelos de simulação de eventos discretos a larga escala e com um grande nível de processamento. Este pacote de simulação suporta múltiplas heranças, *templates*, tipos de referências, polimorfismo e simulação orientada a processos, bem como atividades síncronas e assíncronas (Figura 2.20).

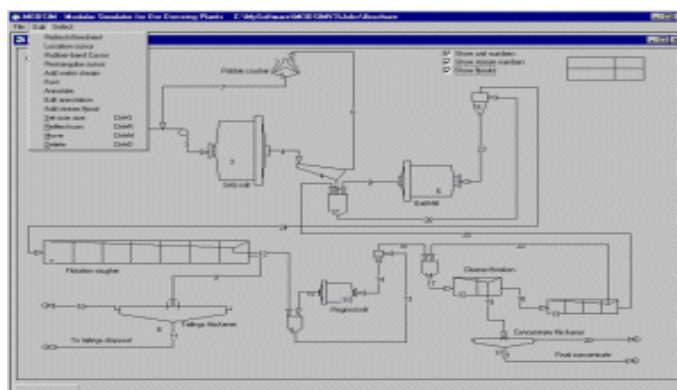


Figura 2.20: Exemplo de simulação com o pacote de simulação MODSIM (MODSIM 2016)

2.5.3 SIMPROCESS

É uma ferramenta de simulação hierárquica que combina mapeamento, simulação de eventos discretos e *Activity-Based Costing*. Possui também modelação orientada a objetos, bem como a capacidade customizar modelos através de linguagens de atributos e expressões (Figura 2.21).

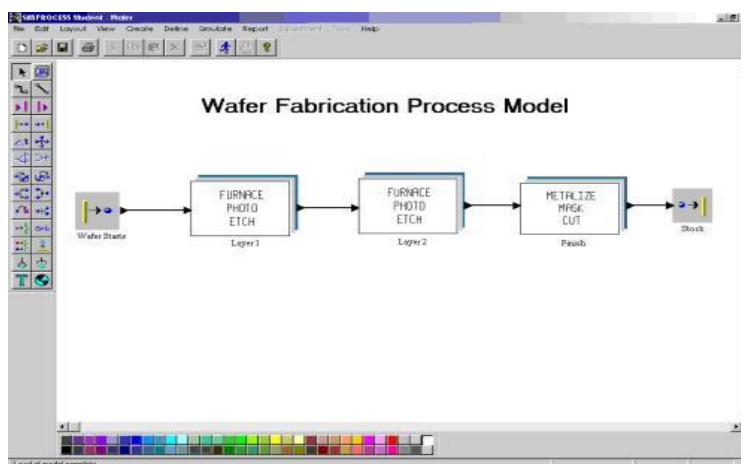


Figura 2.21: Exemplo de simulação com o pacote de simulação SIMPROCESS (Pau Casas 2009)

2.5.4 SIMUL8

Usado para simular sistemas que envolvam o processamento de entidades discretas em tempos discretos. Este programa permite que os seus utilizadores criem um modelo de simulação em computador que tem em conta as limitações encontradas em sistemas de manufatura reais, tais como uma falha súbita num componente da linha de montagem ou uma mudança de turnos de trabalho inesperada (Kieran et al. 2007). Através deste modelo é possível testar cenários de manufatura reais em ambientes virtuais, testando a capacidade do sistema até ao limite de forma a poder encontrar a solução ótima. A construção de modelos neste *software* não cos-

tumam ser baseados em dados estatísticos ou em programação, mas sim através do desenho de esquemas organizacionais (Figura 2.22).

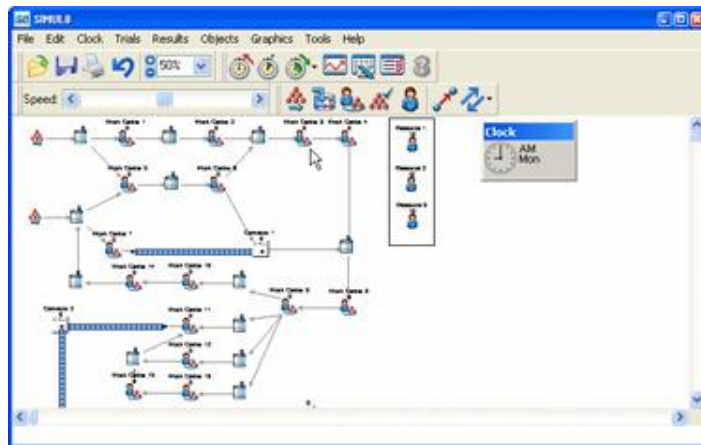


Figura 2.22: Exemplo de simulação com o pacote de simulação SIMUL8 (SIMUL8 2016)

2.5.5 ARENA

Este pacote de simulação permite a simulação de eventos discretos bem como a automação de processos, usando para tal o processador/linguagem de programação SIMAN. Neste pacote de simulação o utilizador constrói um modelo de simulação, no qual são colocados *módulos* que representam processos ou lógica (Figura 2.23). Estes módulos são interligados através de linhas de conexão, linhas que irão especificar o fluxo de *entidades*. É possível recolher dados estatísticos – tais como tempos de ciclo ou níveis de *Work in Progress* - ao longo da simulação.

Este pacote de simulação pode ser integrado com tecnologias Microsoft, tais como VBA (Visual Basic for Applications), assim como *flowcharts* do Microsoft Visio, *spreadsheets* de Excel e bases de dados no Access.

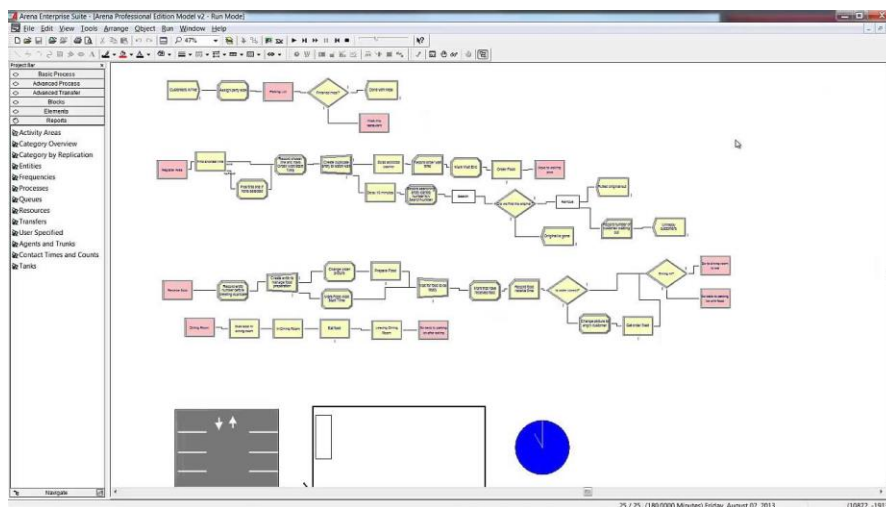


Figura 2.23: Exemplo de simulação com o pacote de simulação ARENA (Alex Adamtsev 2016)

2.5.6 AnyLogic

O AnyLogic é um pacote de simulação desenvolvido pela AnyLogic Company (Figura 2.24). Esta ferramenta de simulação permite simulações em três principais paradigmas: simulações baseadas em agentes, simulações de eventos discretos e simulações de dinâmicas de sistemas (AnyLogic 2016). Este pacote de simulação é amplamente utilizado na simulação de modelos de negócio nas áreas da manufatura, saúde, defesa e logística, na simulação de dinâmicas sociais, infraestruturas de Tecnologias de Informação e na simulação de dinâmicas de tráfego.

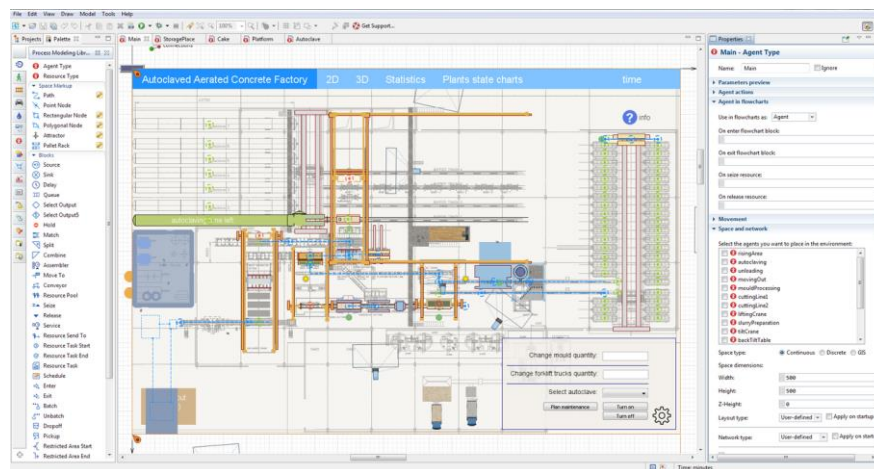


Figura 2.24: Exemplo de simulação com o pacote de simulação AnyLogic (AnyLogic 2016)

Na tabela 2-2 é feita uma breve comparação entre os pacotes de simulação descritos previamente:

Tabela 2-2: Comparação dos pacotes de simulação analisados

Sistema	Tipo de sistema	Tipo de modelos	Alguma aplicações
ProModel	Pacote de simulação	Eventos discretos	Manufatura, armazenamento, logística
MODSIM	Linguagem de simulação	Eventos discretos, programação orientada a objetos	Indústria de processamento mineira
SIMPROCESS	Pacote de simulação	Eventos discretos, programação orientada a objetos	Saúde, logística e recursos humanos
SIMUL8	Pacote de simulação	Eventos discretos	Manufatura, logística
ARENA	Pacote de simulação	Eventos discretos	Manufatura, logística
AnyLogic	Pacote de simulação	Eventos discretos, modelos dinâmicos, simulações baseadas em agentes	Manufatura, TI, saúde, entre outras

2.6 Pacotes de simulação de Robôs

Os pacotes de simulação mencionados previamente são bastante úteis para desenvolver modelos de simulação de sistemas distribuídos de manufatura. Porém, estes não conseguem, por exemplo, simular as ações de um robô presente numa linha de produção. Para tal, existem pacotes de simulação de robôs dedicados, que são capazes de simular diversos processos associados a um robô presente numa linha de produção.

2.6.1 RobotStudio

O pacote de simulação considerado neste trabalho é o RobotStudio da ABB (ASEA Brown Boveri). Este pacote de simulação permite a programação *offline* de robôs, permitindo realizar alterações sem que seja necessário parar a linha de produção (Figura 2.25). Este pacote de simulação fornece as ferramentas necessárias para realizar tarefas de otimização, treino e programação sem provocar distúrbios na produção.

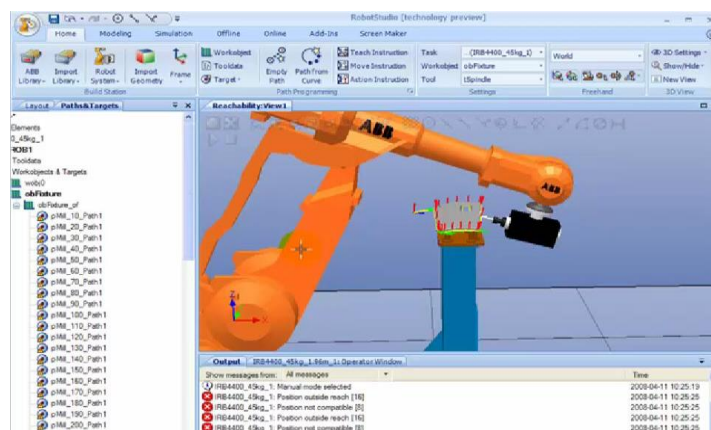


Figura 2.25: Exemplo de utilização do pacote de simulação RobotStudio (ABB 2016)

2.6.2 RoboDK

O RoboDK é um pacote de simulação de robôs *offline* utilizado na simulação de robôs industriais. Este pacote de simulação pode ser utilizado para simular diversas operações de manufatura tais como perfuração, soldadura ou pintura (Figura 2.26). Esta ferramenta caracteriza-se também por possuir uma biblioteca com mais de duzentos robôs de quinze fabricantes diferentes, tais como a ABB, Fanuc ou Kuka (Montaqim 2015).

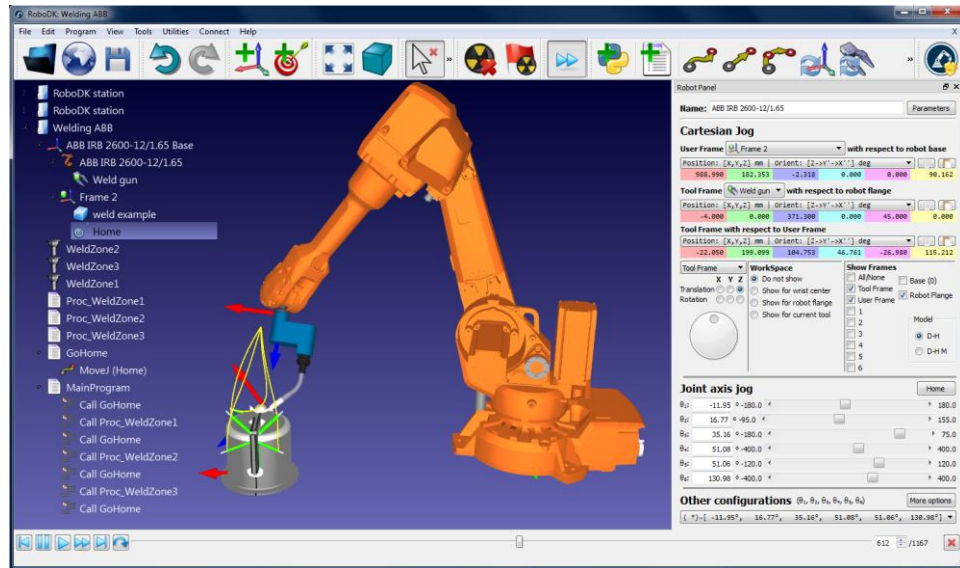


Figura 2.26: Exemplo de utilização do pacote de simulação RoboDK (RoboDK 2016)

2.6.3 RobotExpert

O RobotExpert é um pacote de simulação da Siemens (Figura 2.27). Esta ferramenta caracteriza-se pela capacidade de simulação de células completas de manufatura assim como pela programação *offline* de robôs, das suas ferramentas e dos equipamentos periféricos (Siemens 2016).

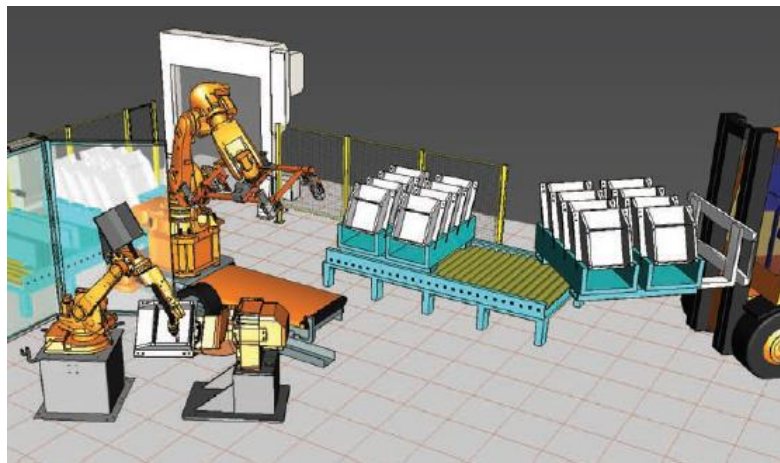


Figura 2.27: Exemplo de utilização do pacote de simulação RobotExpert (Siemens 2016)

2.7 Alguns trabalhos realizado na área de simulação de SDM

Dentro da área da simulação de sistemas distribuídos de manufatura existem diversos trabalhos realizados. Por exemplo (Uygun et al. 2009) descreve um trabalho na área da simulação de sistemas distribuídos de manufatura através da utilização de arquiteturas de alto nível. Neste

trabalho foi considerado um cenário de um sistema distribuído de manufatura composto por três unidades fabris, uma unidade de fornecimento e um sistema de gestão de clientes (Figura 2.28). Neste cenário existe uma fábrica principal responsável pela produção do produto final, estando em constante cooperação com as restantes fábricas. O sistema de gestão de clientes gera pedidos que serão tratados pela fábrica principal, que por sua vez irá criar pedidos que serão tratados pelas restantes fábricas e pela unidade de fornecimento. Tal como a fábrica principal, as restantes fábricas irão também comunicar com a unidade de fornecimento para satisfazer os seus pedidos. Depois de processados todos os pedidos pendentes é efetuado um compêndio da informação recolhida e esta é entregue ao sistema de gestão de clientes, que se assegurará que todos os pedidos foram completados.

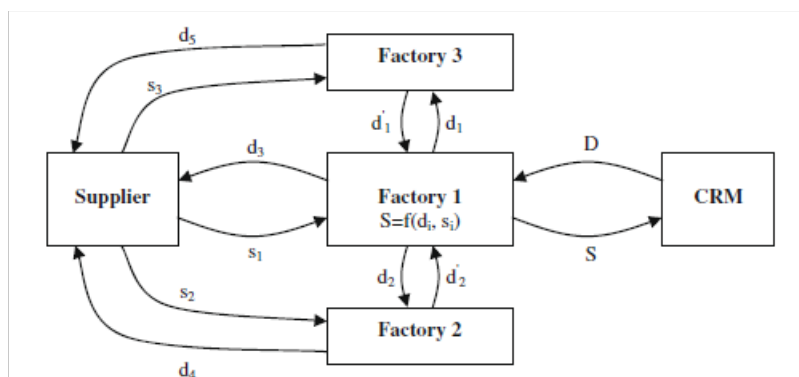


Figura 2.28: Demonstração de um sistema distribuído de manufatura utilizando arquiteturas de alto nível (Uygun et al. 2009)

Uma arquitetura de alto nível pode ser definida como uma combinação de várias simulações efetuadas através de um computador numa simulação mais abrangente (Kuhl, F., Weatherly, R., & Dahmann 1999). Esta fornece um conjunto de regras funcionais e de interfaces comuns para integrar sistemas de simulação separados num sistema de simulação de maior nível. Esta foca-se em dois pontos principais: a interoperabilidade entre sistemas de simulação e a reutilização dos seus modelos em contextos diferentes (IEEE 2000).

Para desenvolver simulações complexas de sistemas de manufatura requiere a colaboração de vários agentes de diversas áreas de conhecimento. De forma a permitir o desenvolvimento de simulações de manufatura de forma cooperativa e integrada utilizando recursos geograficamente distribuídos, a arquitetura de alto nível pode ser utilizada como facilitador entre simulações de sistemas distribuídos de manufatura (Uygun et al. 2009).

No trabalho realizado por (Strassburger et al. 2003), foi abordada a simulação de sistemas distribuídos de manufatura como uma tecnologia presente no desenvolvimento de uma fábrica digital. Neste trabalho foram considerados dois tipos de simulação na área da manufatura virtual: simulações de fluxo de materiais e simulações de robôs, que por sua vez representam duas áreas importantes da simulação na área da manufatura – a simulação de eventos discretos e a simulação de eventos contínuos.

O cenário principal em estudo é constituído por um robô capaz de realizar medições – que será o componente principal do sistema - e por diversas simulações de fluxo de material que representam diferentes componentes de uma fábrica virtual, ou seja, diferentes linhas de manufatura (Figura 2.28). Estes modelos são então acoplados, representando assim a integração horizontal de sistemas de fluxo de material autónomos e geograficamente distribuídos.

O acoplamento destes modelos permite o estudo de fenómenos tanto ao nível local como ao nível global, sendo possível estudar interações, efeitos colaterais e receber diretamente o *feedback* desta interoperabilidade, seja positivo ou negativo.

Para além dos modelos de fluxo de material fazem também parte desta simulação a utilização de simulações de robôs. Estas permitem realizar estudos sobre acessibilidade e controlo de colisões num grande nível de detalhe através do uso de geometria a três dimensões. No cenário considerado na figura 2.29, considera-se uma peça genérica produzida pelo modelo A. A qualidade desta peça é verificada pela simulação do robô. Consoante a qualidade aferida, a peça é enviada para a próxima linha de produção (modelo B) ou então é processada novamente (modelo C). Dependendo do nível de defeitos da peça, o modelo C envia a peça para o modelo D (caso os defeitos sejam de baixo nível), onde estes defeitos serão corrigidos e a peça será reintroduzida na linha de produção do modelo A. Caso os defeitos sejam de alto nível o modelo C envia a peça para o processo de reciclagem no modelo E.

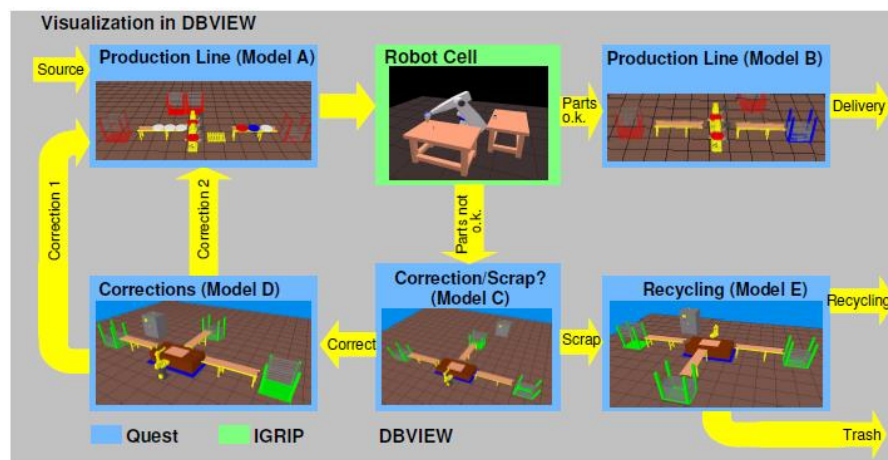


Figura 2.29: Cenário de simulação distribuída (Strassburger et al. 2003)

3 Desenvolvimento da infraestrutura de suporte

Neste capítulo vai-se descrever o desenvolvimento da infraestrutura que vai permitir integrar os diversos pacotes de simulação comerciais, de forma a poder-se simular Sistemas Distribuídos de Manufatura de forma abrangente, conforme o que já foi referido nos capítulos anteriores. Tendo em conta a metodologia abordada no primeiro capítulo, neste terceiro capítulo será elaborada a construção de uma hipótese de resolução da questão de pesquisa formalizada previamente.

Para este caso em particular, importa adaptar o estudo de simulação para a simulação específica de um SDM. Assim sendo, a secção seguinte recai precisamente sobre o problema de como especificar modelos de simulação para SDM. O resultado deste estudo vai permitir determinar quais as características que a infraestrutura de suporte à plataforma de simulação deverá possuir.

Essas características serão então formalizadas através de um processo que engloba algumas fases do projeto de desenvolvimento de *software*, conforme ilustrado na figura 3.1.

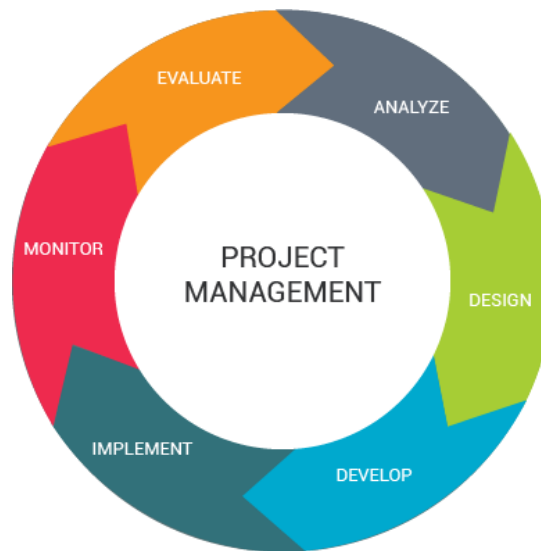


Figura 3.1: Ciclo de desenvolvimento de um projeto de *software* (promatics 2016)

3.1 Modelação de um SDM para simulação

Conforme descrito no capítulo anterior, um SDM é um sistema complexo que engloba diferentes realidades que se manifestam em distintas dimensões que por sua vez se materializam estruturalmente em padrões organizacionais, tanto verticais como horizontais de forma a simular com os processos distribuídos em todos os sistemas em que estes se encontrem. Em termos verticais, os processos que decorrem ao nível estratégico vão-se decompondo em processos cada vez mais detalhados até se chegar ao nível das tarefas que serão executadas num sistema flexível de manufatura (SFM), e até dentro das próprias máquinas, por exemplo os robôs que fazem parte de um SFM. A figura 3.2 ilustra este aspeto de decomposição:

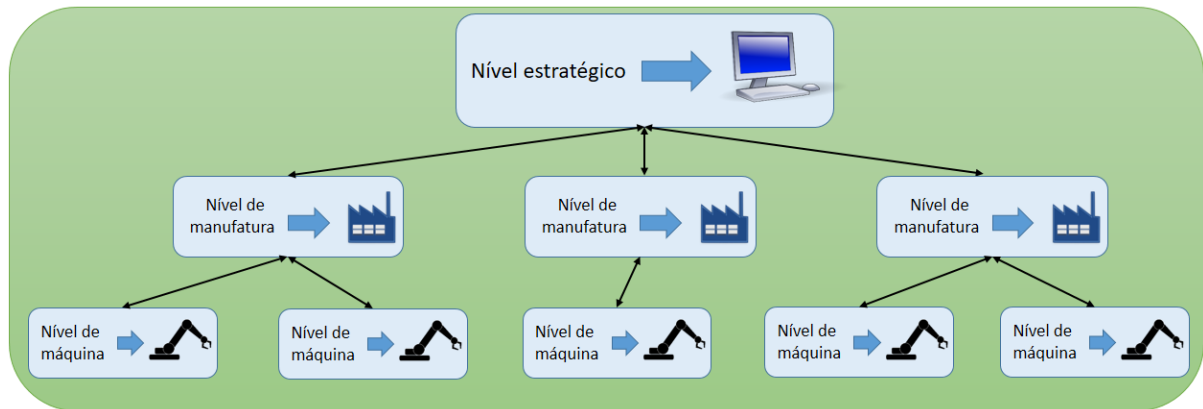


Figura 3.2: Esquema representativo da decomposição vertical dos processos desde o nível estratégico até ao nível de máquina

Em termos horizontais, caso exista uma empresa ou consórcio que seja responsável pelo processo estratégico mais abstrato, assim que se desce para níveis mais detalhados a responsabilidade pela execução dos processos mais específicos pode ser atribuída a entidades (ou outras empresas ou consórcios) distintas a trabalharem em paralelo. Estas entidades possuem normalmente um funcionamento autónomo, pelo que são necessários mecanismos de coordenação para que um processo decorra de forma adequada. Adicionalmente, algumas entidades podem até participar em simultâneo nos diversos níveis hierárquicos, conforme também ilustrada na figura anterior 3.2.

Assim sendo, cada processo concreto irá ter uma natureza específica. Por exemplo, alguns processos relacionam-se com a logística de um SDM, enquanto que outros estão em execução nas respetivas máquinas de um SDM. Cada atividade de um processo pode ter precedências, ou seja, antes de poder iniciar outras atividades deverão ter terminado. Para além disso, uma atividade de um processo necessita de diversos tipos de recursos em quantidades específicas. Esses recursos podem ser necessários para outras atividades, por vezes até em simultâneo.

Os próprios produtos manufacturados, quando complexos, induzem este tipo de organização hierárquica e distribuída. Por exemplo, no fabrico de um automóvel, os diversos módulos constituintes, tais como o motor, a carroceria ou o chassis são produzidos em sistemas distribuídos e autónomos, normalmente, geridos por distintos parceiros.

Tendo em consideração o objetivo deste trabalho, que é de simulação de SDM de uma forma mais abrangente, será necessário a utilização de um pacote de simulação que consiga modelar todos os aspetos atrás considerados. Mas a construção de um pacote deste tipo seria um empreendimento com desafios muito difíceis de ultrapassar.

Portanto, e tal como inicialmente proposto, a melhor alternativa é então combinar adequadamente distintos pacotes de simulação numa plataforma agregadora que, essa sim, permitirá simular efetivamente SDM. Cada pacote de simulação seria utilizado em algumas realidades específicas de um SDM, conforme a adequabilidade de cada um dos pacotes a cada realidade dum SDM.

É assim necessário definir a arquitetura da plataforma proposta que tenha em consideração estes aspetos complexos. Uma das componentes principais dessa plataforma é a infraestrutura de interoperabilidade, que vai permitir que os diversos pacotes de simulação consigam funcionar e interagir de forma harmoniosa. O passo seguinte para o desenvolvimento desta plataforma consiste na análise e identificação dos requisitos, conforme aliás demonstrado na figura 3.1.

3.2 Identificação dos requisitos funcionais

Tipicamente, um requisito funcional pode ser definido como o comportamento pretendido para um dado sistema. Este comportamento pode ser expressado em serviços, tarefas ou funções que o sistema tenha que cumprir (Malan & Bredemeyer 2001).

Para a plataforma proposta, a tabela 3-1 apresenta os requisitos funcionais mais relevantes.

Tabela 3-1: Requisitos funcionais considerados no desenvolvimento da plataforma proposta

Requisito	Importância
Adicionar nós de simulação na plataforma	Dado que a simulação de SDM vai ser efetuada por vários nós em simultâneo, é necessário fazer a gestão desses nós
Remover os nós de simulação	Conforme uma simulação é preparada, assim se deverá adicionar novos nós e eventualmente remover outros
Configuração dos nós de simulação	A plataforma deverá disponibilizar interfaces (GUI) que permitam configurar cada nó, por exemplo, calibrar uma estação composta por um robô
Simulação de processos estratégicos	Ao nível estratégico pretende-se que a plataforma se baseie em pacotes de simulação adaptados para estes níveis, por exemplo o ARENA
Simulação de processos mais específicos (nós locais)	Para os processos que decorrem ao nível de manufatura, são necessários pacotes de simulação mais específicos, por exemplo o RobotStudio

Estes requisitos, embora pudessem ser mais detalhados, servem de guia para uma arquitetura correspondente, tal como descrito nas secções seguintes.

Importa no entanto mencionar que muitas das funcionalidades requeridas na plataforma proposta serão fornecidas pelos próprios pacotes de simulação disponíveis. No contexto deste trabalho vai-se modelar os níveis mais abstratos de um SDM através do pacote de simulação ARENA, enquanto que a simulação dos nós locais será efetuada através do pacote de simulação RobotStudio. Nas secções seguintes, serão introduzidos os pacotes ARENA e RobotStudio com o nível de detalhe adequado tendo em conta os objetivos do trabalho.

A apresentação dos pacotes de simulação vai ser focada em dois aspetos essenciais para este trabalho: Como modelar um nó ou subsistema para simulação e como é que se implementa a interoperabilidade.

3.3 O pacote de simulação ARENA

3.3.1 Modelação e simulação de sistemas através do ARENA

Neste subcapítulo será descrito um exemplo de simulação de um sistema genérico de manufatura, desde a fase de especificação do sistema até à fase da simulação do sistema no ARENA. Neste exemplo, presente em (Law & Kelton 2000), é considerado um modelo representativo de uma porção de um sistema de manufatura. Neste sistema, uma peça genérica é introduzida na linha de produção, processada por uma estação de manufatura e por fim, abandona o sistema (Figura 3.3). Quando a peça chega à estação de manufatura e esta está livre, a peça é processada imediatamente. Caso contrário, esta fica numa fila de espera assente na lógica FIFO (*First In – First Out*).

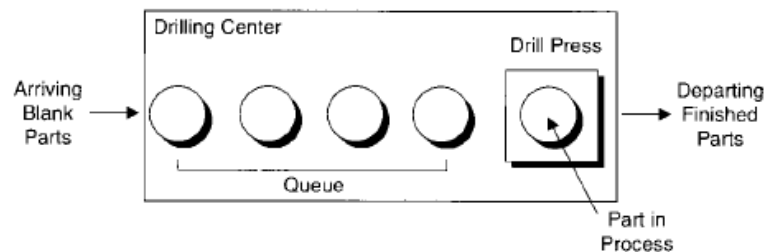


Figura 3.3: Esquema do sistema de manufatura exemplificado (Law & Kelton 2000)

De forma a simular-se este sistema, é necessário especificar os aspetos numéricos do sistema, assim como os critérios de começo e fim de simulação. É também necessário especificar a unidade temporal do sistema – no exemplo em questão será usado o minuto como unidade temporal. O sistema poderá ter mais do que uma unidade temporal – como por exemplo, utilizar minutos para contabilizar um processo de curta duração e horas para contabilizar um processo de longa duração – mas para efeitos aritméticos, todas as unidades terão que ser convertidas para uma única unidade base.

Tendo em conta a observação de três grandezas principais – o tempo de chegada da peça, o tempo entre a chegada de duas peças e o tempo de processamento de cada peça – definiram-se alguns objetivos a serem estudados:

- A produção total de peças durante o intervalo de tempo considerado
- O tempo médio de espera na fila de espera de cada peça
- O tempo máximo de espera na fila de espera
- O número máximo de peças numa fila de espera
- O tempo médio e máximo que cada peça fica na linha de produção
- O tempo de utilização da estação de manufatura

Depois de definidas as constantes numéricas do sistema, é necessário especificar os diversos componentes do sistema:

- Entidades: estes componentes do sistema podem ver o seu estado afetado quer pelo sistema, quer pela ação de outras entidades. As entidades são os objetos dinâmicos da simulação – na maioria dos casos, estas são criadas, afetadas durante algum tempo e des-

cartadas aquando do término do sistema. No sistema em questão, as entidades serão as peças a serem processadas.

- Atributos: serão utilizados para individualizar entidades. Um atributo é uma característica global a todas as entidades, mas que pode ter um valor específico de forma a diferenciá-las. No caso em estudo, alguns dos atributos das peças poderão ser a cor ou o nível de prioridade.
- Variáveis globais: estas variáveis representam uma informação que reflete algumas características do sistema, independentemente do tipo de entidades em jogo. Num sistema de manufatura, estas podem ser utilizadas para definir um tempo de transferência entre duas estações de trabalho diferentes.
- Recursos: representam objetos tais como equipamento ou espaço de armazenamento. As entidades competem entre si para utilizarem um recurso quando este estiver disponível e para o libertarem quando terminarem a sua utilização.
- Filas de espera: Quando uma entidade não consegue utilizar um recurso visto este já estar a ser utilizado por outra entidade, esta é colocada numa lista de espera, onde ficará até poder utilizar o recurso previamente ocupado.
- Acumuladores estatísticos: servem para contabilizar diversos aspetos da simulação enquanto esta decorre. No sistema em estudo, estes acumuladores poderão ser utilizados para contabilizar o número de peças produzidas até ao momento, o tempo de espera numa fila de espera, o número de peças que passaram por uma certa estação de trabalho, entre outros aspetos.
- Eventos: um evento representa um acontecimento durante a simulação, acontecimento este onde poderá existir uma mudança nos atributos, variáveis ou acumuladores estatísticos. No sistema em estudo, serão considerados três eventos principais: chegada (quando uma nova peça chega ao sistema), partida (quando é finalizado o processamento de uma peça e esta abandona o sistema) e a finalização (quando a simulação é terminada, isto é, quando esta atinge o limite temporal estabelecido).

Depois de considerados os componentes do sistema em estudo, é então efetuada a simulação. No exemplo considerado, é feita uma simulação manual em primeiro lugar, cujos passos serão descritos em seguida. Após esta simulação, os aspetos considerados previamente são transpostos para o pacote de simulação ARENA, onde será criado um modelo representativo do exemplo considerado, sendo depois efetuada uma simulação.

De acordo com o exemplo apresentado em (Law & Kelton 2000), a simulação manual efetuada segue os seguintes passos:

- Chegada: Uma nova parte chega à linha de montagem; é agendada a chegada da peça seguinte através da criação de um registo de evento; é atualizado o registo temporal estatístico; é guardado o tempo de chegada de peça num atributo; caso a estação de manufatura esteja livre, a peça acabada de chegar é processada – durante este processo a estação de trabalho é considerada ocupada e é agendado o fim do processo de manufatura. Caso a estação esteja ocupada, a peça acabada de chegar é colocada no final da fila de espera, sendo o tamanho desta incrementado.
- Partida: A peça processada pela estação de manufatura está pronta a ser libertada; o acumulador estatístico das peças produzidas é incrementado; é calculado o tempo total que a peça processada esteve no sistema; caso estejam algumas peças na fila de espera, a primeira é retirada e enviada para a estação de trabalho, o tempo que esta esteve na fila é guardado e a estação de trabalho começa o seu serviço, ficando agendado um evento de partida; em caso contrário caso a fila de espera esteja vazia, a estação de trabalho ficará livre e não será agendado nenhum registo de partida.

- Final: a simulação é dada como terminada; é atualizada a estatística temporal da simulação e são calculadas as medidas de desempenho do sistema.

Após a simulação manual, o próximo passo do exemplo apresentado em (Law & Kelton 2000) passa pela simulação do mesmo modelo em ARENA. Como referido anteriormente no segundo capítulo, os blocos principais nos modelos em ARENA são denominados de módulos. Estes módulos podem ser divididos em duas categorias: os módulos de fluxogramas, onde ocorrem os processos dinâmicos do modelo (onde se encontram os módulos de criação, descarte, processamento e decisão); os módulos de dados, que definem as características dos vários elementos de processamento, como entidades, recursos e filas de espera. Estes módulos também podem ser utilizados para estabelecer variáveis e outros tipos de expressões numéricas que atuem sobre o sistema.

Entrando em detalhe nos blocos principais, o módulo de criação será o ponto de partida do sistema – no exemplo em questão, é neste bloco que se dá a entrada de novas peças na linha de montagem. É também neste bloco que se pode definir o nome da entidade, bem como o tempo entre chegadas de novas peças como é possível verificar na figura 3.4.

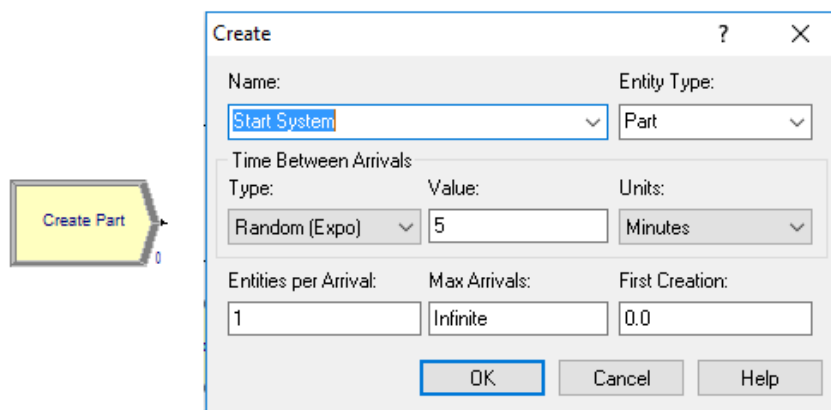


Figura 3.4: Detalhe do módulo de criação

Depois de criado este módulo, o próximo passo é a criação do módulo de processo. Este será o bloco representativo da estação de manufatura, bem como dos recursos, tempos de processamento e filas de espera associados à mesma. Como é possível ver na figura 3.5, uma das ações possíveis para o comportamento do módulo de processamento passa pela ação “*Seize Delay Release*”. Neste tipo de ação, o módulo irá permitir que uma entidade utilize um certo recurso (após uma eventual momento de espera numa fila de espera), libertando-o quando o processamento tiver terminado. Existem outros tipos de ação disponíveis para modelar o comportamento, que podem introduzir atrasos ou criar diferentes sequências de utilização de recursos. Na figura 3.5 é também possível observar mais duas condicionantes do sistema que podem ser definidas: os recursos que podem ser utilizados, bem como a sua quantidade; a distribuição probabilística do atraso do módulo de processamento e a unidade temporal que este segue.

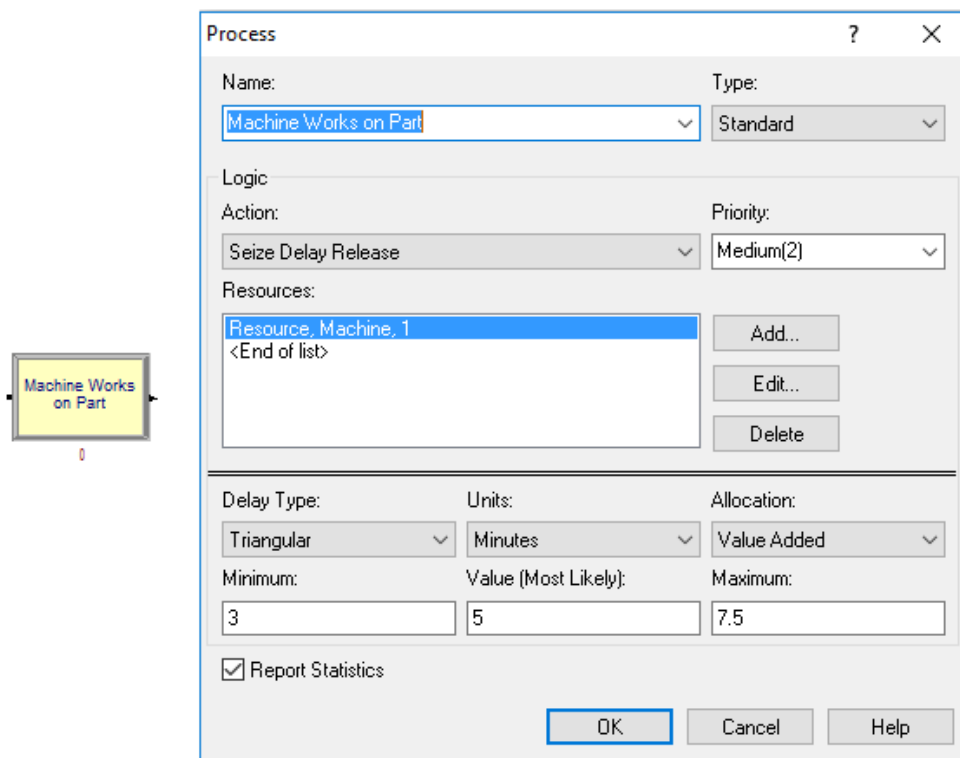


Figura 3.5: Detalhe do módulo de processo

Em relação aos módulos de dados, no exemplo em questão consideraram-se dois módulos: o módulo de recursos e o módulo de fila de espera. Quando for definido um recurso no módulo de processo, uma entrada para o recurso é criada no módulo de recursos. Neste módulo (Figura 3.6) é possível estabelecer algumas características do recurso, tal como a sua capacidade (se é fixa ou se varia de acordo com um certo agendamento) ou algumas condições de paragem de utilização do recurso.

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1 ▶	Machine	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Figura 3.6: Detalhe do módulo de recursos

Se chegar uma entidade ao módulo de processamento enquanto o recurso se encontra em utilização, essa entidade irá para uma fila de espera. No módulo de fila de espera (Figura 3.7) é possível controlar alguns aspetos das filas de espera, tal como o modo em como operam.

Queue - Basic Process				
	Name	Type	Shared	Report Statistics
1 ▶	Machine Works on Part.Queue	First In First Out	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 3.7: Detalhe do módulo de filas de espera

Por fim, é necessário criar um módulo de descarte. Este módulo representa o final do modelo de simulação, sendo possível associar a saída de dados ao relatório estatístico criado no final da simulação. O modelo criado terá um aspeto semelhante à figura 3.8:

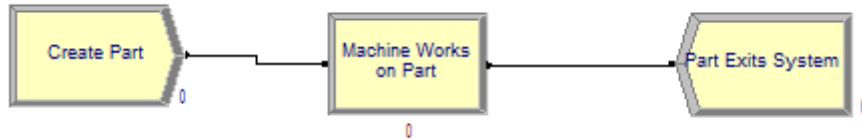


Figura 3.8: Modelo de simulação representativo de um pequeno processo de manufatura

Após a criação do modelo de simulação é necessário estabelecer os parâmetros de cada iteração, como é possível verificar na figura 3.9. Para além da velocidade de animação de cada iteração, o ARENA permite também estabelecer a duração de cada replicação, a unidade temporal destas ou um tempo de aquecimento de cada estação de trabalho.

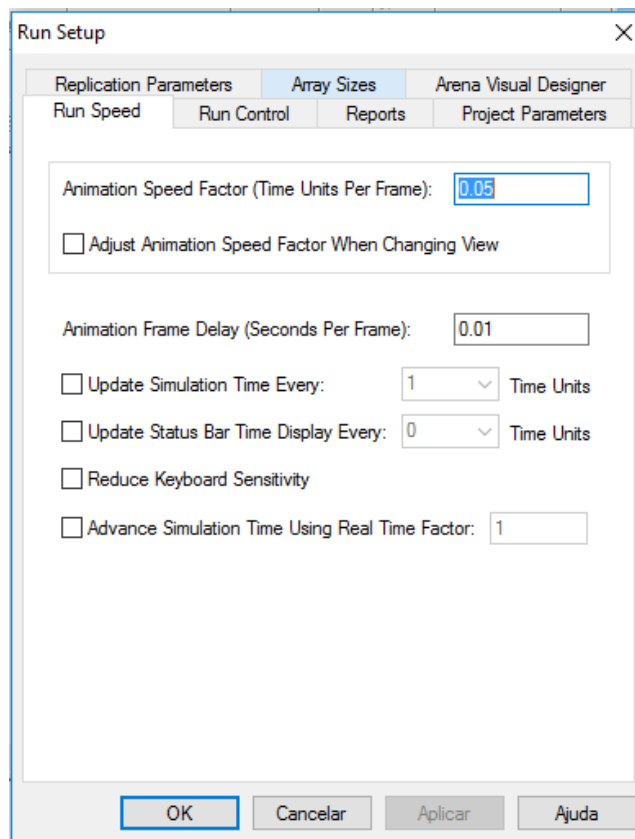


Figura 3.9: Estabelecimento das condições de execução de cada iteração

Depois deste processo, é possível então simular o modelo criado. No final da simulação é gerado um relatório de dados estatísticos correspondentes à produção de cada peça. Neste relatório é possível analisar alguns tempos relevantes e dados significativos da simulação (tempos de espera, número de peças produzidas, tempo médio de processamento de cada peça, entre outros) (Figura 3.10). Estes dados serão analisados de forma mais detalhada no capítulo 4.

Time

VA Time	Average	Half Width	Minimum Value	Maximum Value
Part	61.7626	(Insufficient)	32.3608	86.7656

NVA Time	Average	Half Width	Minimum Value	Maximum Value
Part	0.00	(Insufficient)	0.00	0.00

Wait Time	Average	Half Width	Minimum Value	Maximum Value
Part	0.00	(Insufficient)	0.00	0.00

Transfer Time	Average	Half Width	Minimum Value	Maximum Value
Part	0.00	(Insufficient)	0.00	0.00

Other

Number In	Value
Part	90.0000

Number Out	Value
Part	79.0000

WIP	Average	Half Width
Part	10.5610	(Insufficient)

Figura 3.10: Detalhe do relatório estatístico gerado aquando do término da simulação

3.3.2 Interoperabilidade com o ARENA

Apesar de todas as valências proporcionadas por estes modelos de simulação, nem sempre estes conseguem representar fielmente o comportamento de um sistema. Um sistema de manufatura distribuído não consegue ser simulado tão facilmente como o sistema apresentado – neste tipo de sistemas existem sempre diversas variáveis fora do controlo do utilizador: as estações de manufatura podem nem sempre estar disponíveis para atuar derivado da ocorrência de problemas de logística, avarias ou outras falhas técnicas que nem sempre são fáceis de corrigir – tratando-se de um sistema distribuído, o controlo do sistema e as estações de manufatura podem estar localizadas em regiões diferentes. O modelo apresentado na figura 3.11 é composto por vários processos de manufatura que trabalham de forma paralela, sendo que os processos que compõem o ramo inferior do modelo são compostos por blocos VBA (*Visual Basic for Applications*) (Figura 3.12) que podem ser utilizados para efetuar a comunicação com estações reais de manufatura.

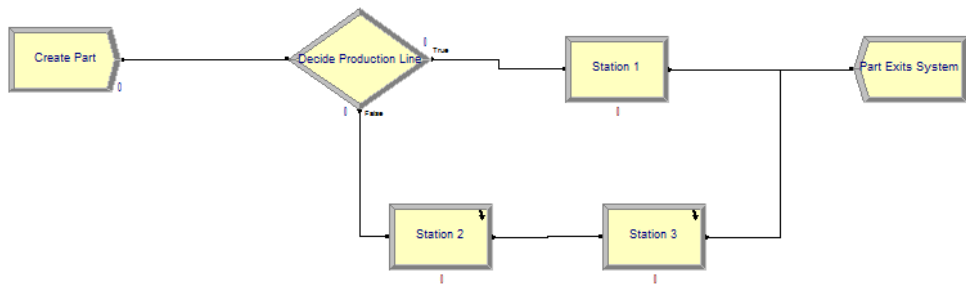


Figura 3.11: Modelo de simulação representativo de um processo de manufatura mais complexo

Tendo em conta que estes processos poderão comunicar com estações remotas, cujo controlo pode fugir do âmbito do utilizador, é importante garantir que estas estão a funcionar corretamente antes de simular o comportamento agregado do sistema. Assim sendo, é imperioso poder simular o comportamento destas estações locais, o que não é possível fazer com o ARENA, sendo necessário utilizar outros pacotes de simulação.

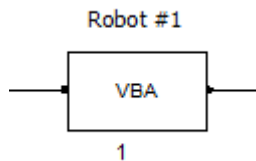


Figura 3.12: Detalhe do bloco de processo composto por um bloco VBA

O bloco VBA, quando integrado num modelo de simulação ARENA, permite então a interação com nós que simulem aspetos mais específicos de um SDM.

3.4 O pacote de simulação RobotStudio

3.4.1 Modelação e simulação através da utilização do RobotStudio

O RobotStudio pode ser definido como um pacote de simulação que permite programar e simular células industriais compostas por robots (Queirós 2016). Este pacote de simulação caracteriza-se também pela capacidade de incorporar um modelo virtual de um controlador de um robot real. Assim sendo, é possível configurar e programar este controlador através de uma linguagem de programação designada de RAPID. Portanto, é possível criar diversas tarefas de simulação quer através do ambiente gráfico do pacote de simulação, criando pontos de ação e movimentos específicos para o robot, como é possível observar na figura 3.13.

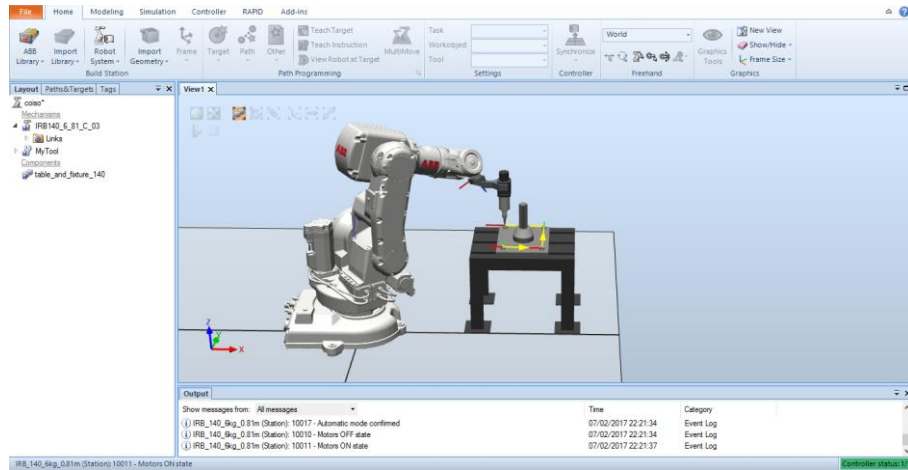


Figura 3.13: Exemplo do ambiente gráfico presente no pacote de simulação RobotStudio

Também é possível programar esses mesmos movimentos através do editor de RAPID embutido no pacote de simulação, como é possível observar na figura 3.14. No anexo 2 é possível consultar de forma mais detalhada o resto do código relativo à tarefa de simulação desenvolvida.

```

T_ROB1/Module1* X
1  MODULE Module1
2
3  VAR socketdev Server1;
4  VAR socketdev Client1;
5
6
7
8  PERS robtarger Target_70:= [[516,125,333],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
9  PERS robtarger Target_80:= [[537.5,75,593],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
10 PERS robtarger Target_90:= [[606,275,293],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
11 PERS robtarger Target_100:= [[556,275,593],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09]];
12
13 PROC Path_20()
14   MoveL Target_70,v100,z100,MyTool\Wobj:=wobj0;
15   MoveL Target_80,v100,z100,MyTool\Wobj:=wobj0;
16   MoveL Target_90,v100,z100,MyTool\Wobj:=wobj0;
17   MoveL Target_100,v100,z100,MyTool\Wobj:=wobj0;
18 ENDPROC
19
20 PROC Path_30()
21   MoveL Target_70,v1000,z100,MyTool\Wobj:=wobj0;
22   MoveL Target_80,v1000,z100,MyTool\Wobj:=wobj0;
23   MoveL Target_90,v1000,z100,MyTool\Wobj:=wobj0;
24   MoveL Target_100,v1000,z100,MyTool\Wobj:=wobj0;
25 ENDPROC

```

Figura 3.14: Exemplo de código RAPID implementado no pacote de simulação RobotStudio

Estas tarefas de simulação podem ser simples, sendo apenas compostas por um procedimento - como um movimento do braço de um robô, ou tarefas mais complexas, compostas por diversos procedimentos.

3.4.2 Interoperabilidade com o RobotStudio

De forma a estabelecer a comunicação entre o RobotStudio e outros pacotes de simulação, recorreu-se ao PC SDK (*Software Development Kit*) da ABB. Esta plataforma, assente pela *framework* .NET, contém as bibliotecas necessárias para a utilização de funções nativas da ABB (relacionadas com os controladores das estações de manufatura) em conjunto com a linguagem de programação C#. Esta linguagem será utilizada no ambiente de desenvolvimento Visual Studio da Microsoft de forma a criar uma *interface* de controlo para o utilizador.

Por intermédio da *framework* .NET, o RobotStudio consegue interagir com o simulador ARENA, através do bloco VBS descrito na secção anterior.

3.5 Arquitetura da plataforma desenvolvida

Tendo em conta os requisitos previamente formulados, a arquitetura proposta cuja construção permite satisfazer esses requisitos é ilustrada na figura 3.15.

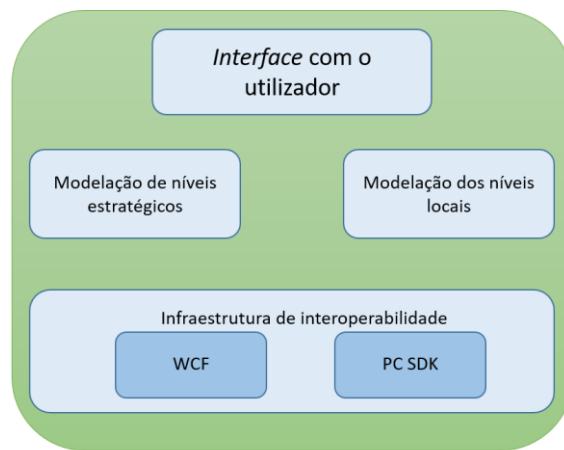


Figura 3.15: Arquitetura proposta para a plataforma de simulação

Ao longo desta secção vai-se descrever o desenvolvimento de cada um dos componentes da plataforma.

3.5.1 Infraestrutura de interoperabilidade

A comunicação realizada entre os diversos componentes da plataforma de simulação é efetuada pela infraestrutura de interoperabilidade. Dela constam o PC SDK como mencionado previamente e um serviço *web* baseado na tecnologia WCF (*Windows Communication Foundation*) da Microsoft. O PC SDK contém nas suas bibliotecas algumas funções que permitem aceder aos controladores virtuais dos robots presentes no pacote de simulação RobotStudio. Na figura 3.16 é apresentado um excerto do código que permite realizar a ligação a um controlador virtual de um robot através de uma aplicação desenvolvida na linguagem de programação C#. No anexo 1 é possível consultar de forma mais detalhada o resto do código.

```

void HandleFoundEvent(object sender, NetworkWatcherEventArgs e)
{
    this.Invoke(new
EventHandler<NetworkWatcherEventArgs>(AddControllerToListView), new Object[] {
this, e });
}

private void AddControllerToListView(object sender, NetworkWatcherEvent-
Args e)
{
    ControllerInfo controllerInfo = e.Controller;
    ListViewItem item = new ListView-
Item(controllerInfo.IPAddress.ToString());
    item.SubItems.Add(controllerInfo.Id);
    item.SubItems.Add(controllerInfo.Availability.ToString());
    item.SubItems.Add(controllerInfo.IsVirtual.ToString());
    item.SubItems.Add(controllerInfo.SystemName);
    item.SubItems.Add(controllerInfo.Version.ToString());
    item.SubItems.Add(controllerInfo.ControllerName);
    this.listView1.Items.Add(item);
    item.Tag = controllerInfo;
}
}

```

Figura 3.16: Excerto de código utilizado para a interligação entre um controlador virtual de um robot e uma aplicação desenvolvida no ambiente de desenvolvimento Visual Studio

Uma vez que esta simulação trata de sistemas distribuídos, era importante que esta pudesse simular sistemas geograficamente distribuídos, estando ligados por uma rede de comunicação (como a internet, por exemplo). Logo, o serviço de comunicação escolhido permite a comunicação entre as entidades do nível estratégico e do nível de manufatura, independentemente da sua localização geográfica. Para tal acontecer é necessário estarem ligados na mesma rede, podendo ser localizados através do seu endereço IP.

O serviço de comunicação criado assenta nas bibliotecas WCF. Estas bibliotecas de comunicação baseiam-se em arquiteturas SOA (*Service-Oriented Architecture*), caracterizadas pela capacidade de criar sistemas compostos por serviços autónomos, sendo possível reutilizar serviços de forma a criar processos mais rápidos e flexíveis (Rosen 2008).

Este serviço de comunicação, também implementado em linguagem C#, que funciona numa lógica de *request-reply*, incorpora funções de envio e receção de dados, que irão receber e enviar dados de controlo entre o nível estratégico e o nível de manufatura. Na figura 3.17 está ilustrado o código utilizado para criar o serviço de comunicação. No anexo 4 é possível consultar de forma mais detalhada o resto do código.

```

namespace WcfService1
{
    public class Service1 : IService1
    {
        public static string send_value;
        public static int ret_val;
        public string hello (string str) {
            return "Hello " + str;
        }

        public void GetDatafromArena(string value)
        {
            send_value=value;
        }

        public string sendDatatoApp()
        {
            string aux;
            aux = send_value;
            send_value = "";
            return aux;
        }
    }
}

```

Figura 3.17: Código relativo ao serviço de comunicação WCF

3.5.2 Modelação dos níveis estratégicos e dos nós locais

O pacote de simulação utilizado na modelação do nível estratégico é o ARENA. Este pacote de simulação permite a criação de modelos de simulação com diferentes níveis de especificações. Como referido no segundo capítulo, os modelos de simulação criados em ARENA são compostos por diversos módulos, tais como os módulos de início e fim de ciclo, módulos de processamento e módulos de decisão. Na figura 3.18 é possível observar um modelo genérico de simulação criado no ARENA.

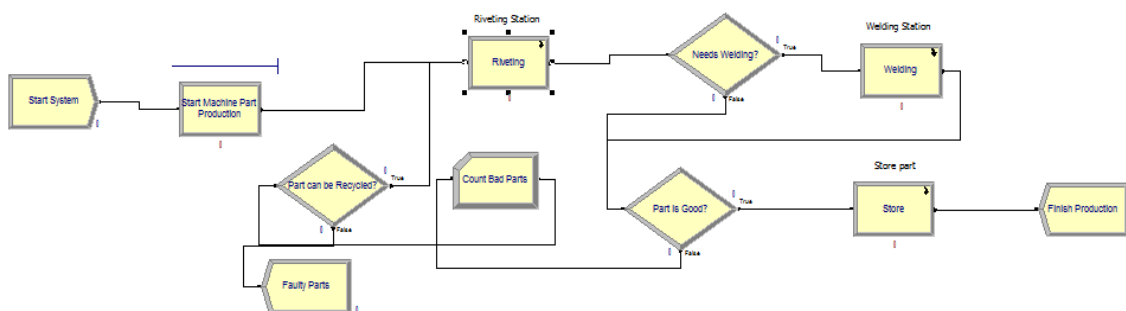


Figura 3.18: Modelo genérico de simulação no ARENA

No modelo de simulação desenvolvido, consideraram-se dois tipos de simulação no nível estratégico: uma simulação realizada de forma interna, ou seja, simular um modelo generalizado uti-

lizando apenas o ARENA, e uma simulação do modelo completo de simulação de SDM, isto é, uma simulação que decorre em distintos níveis.

Em ambos os tipos de simulação, foi considerada uma linha de produção de uma peça genérica. A simulação inicia-se com o pedido de produção de um certo número de peças genéricas no bloco de começo de produção. Estas peças irão seguir as linhas de comunicação – representativas de uma linha de produção real - entre os diversos blocos de processamento – representativas de estações de manufatura – até chegarem ao bloco de fim de produção. Como é possível observar na figura anterior, existem também ao longo da linha de produção alguns blocos de decisão (que irão enviar a peça para estações diferentes de trabalho ou que irão decidir se a peça se encontra em boas condições). Existe também um bloco de contagem, que no caso apresentado, irá contar o número de peças defeituosas. No final da produção é gerado um relatório estatístico, com diversos dados temporais sobre as peças criadas.

Como explicado anteriormente, no primeiro tipo de simulação não se consideraram os níveis inferiores do modelo de simulação. Assim sendo, a simulação da produção de uma peça apenas decorre no ambiente do ARENA e os blocos de processamento apenas afetam a produção com diferentes tempos de processamento e atraso.

No segundo tipo de simulação, o modelo mantém-se mas os blocos de processamento são compostos por subcomponentes (Figura 3.19) que irão ser utilizados para estabelecer a comunicação com os níveis inferiores do modelo de simulação. Neste caso específico irá ser utilizado um bloco VBA programável (*Visual Basic for Applications*) e um bloco contador, que será utilizado para fins estatísticos.

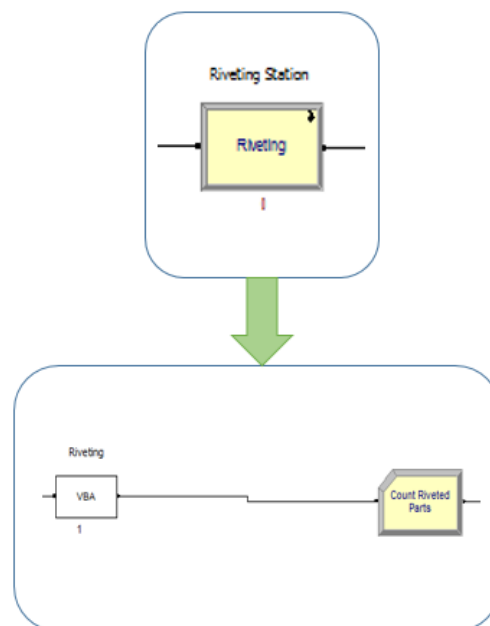


Figura 3.19: Detalhe do bloco de processamento no modelo completo de simulação

O bloco VBA contém funções que irão estabelecer um canal de comunicação com o serviço de comunicação previamente criado, utilizando para tal o endereço do serviço. No anexo 3 é possível consultar de forma mais detalhada o código responsável pelo estabelecimento de uma ligação com arquiteturas exteriores.

Em relação à modelação dos nós locais, esta é tratada pelo pacote de simulação RobotStudio e pela *interface* de controlo (Figura 3.20) que será detalhada na secção seguinte. Neste caso, o utilizador poderá executar diversas tarefas de simulação – sejam estas compostas por um ou mais procedimentos – através da *interface* de controlo. Para tal basta escolher um controlador virtual que esteja disponível. Após a seleção do mesmo, o utilizador poderá então ordenar a execução da tarefa de simulação.

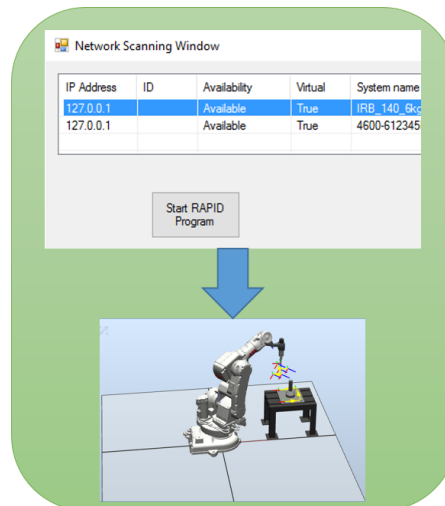


Figura 3.20: Esquema de simulação dos nós locais

3.5.3 Interface de controlo

De forma a gerir a maioria dos processos de simulação de forma intuitiva foi desenvolvida uma *interface* de controlo. Esta *interface*, desenvolvida no ambiente de desenvolvimento Visual Studio com recurso à linguagem C#, permite visualizar quais os controladores virtuais – correspondentes a robots presentes no pacote de simulação RobotStudio – que estão ativos e disponíveis para serem utilizados. Na figura 3.21 é possível observar algumas das funcionalidades fornecidas pela plataforma de simulação: depois de escolhido o controlador desejado na janela de seleção, o utilizador pode executar uma tarefa simples de simulação – simulação ao nível do nó local – ao pressionar o botão *Start RAPID Program*. Caso o utilizador pretenda executar apenas um único procedimento de simulação (normalmente uma tarefa é composta por diversos procedimentos), deverá escolher o nome do módulo – isto é, o nome da tarefa de simulação – e o nome do procedimento a ser executado. Para iniciar a simulação o utilizador deverá pressionar o botão *Start Proc*. Para além destas funcionalidades existe também a possibilidade de executar uma calibração manual de um *target* do robot, utilizando para tal os botões de direção (*Up*, *Down*, *Left*, *Right*, *Front* e *Back*), os valores de desvio pretendidos na caixa de texto *Value* e o *target* pretendido na caixa de texto *Target*. Por fim mas sem menos importância, é possível fazer a execução automática do comportamento da plataforma completa de simulação, utilizando para tal o botão *Auto Start*.

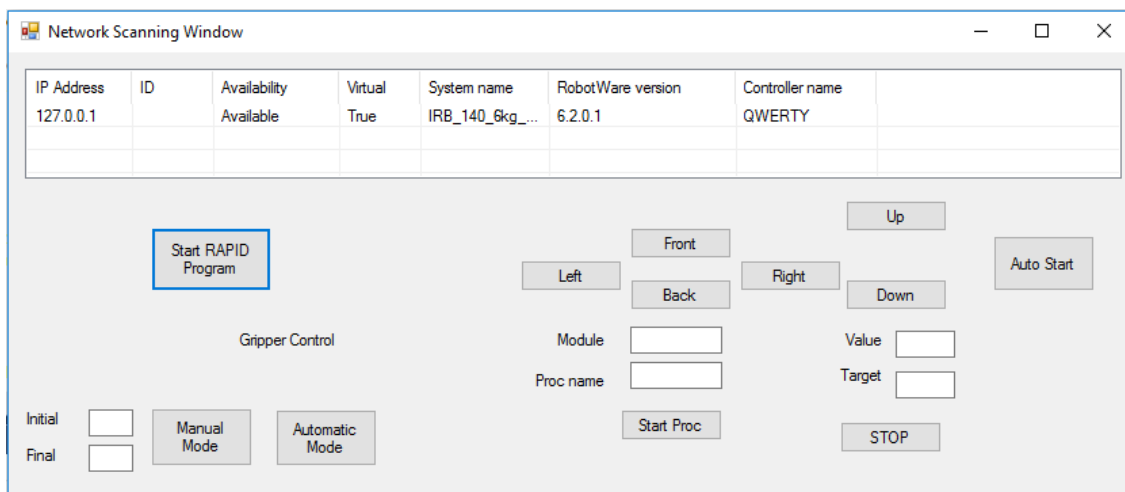


Figura 3.21: Aspeto geral da *interface* de controlo implementada

A arquitetura proposta, desejavelmente simples, destina-se sobretudo a permitir que os diversos pacotes consigam operar uns com os outros, nomeadamente através de serviços *web*. Os respetivos pacotes de simulação escolhidos deverão também minimamente poder interagir através dos serviços *web*.

3.6 Plataforma de simulação obtida

Depois de abordados os diversos componentes da plataforma desenvolvida, resta apresentá-la (Figura 3.22) e descrever o seu comportamento desta vez de uma forma agregada.

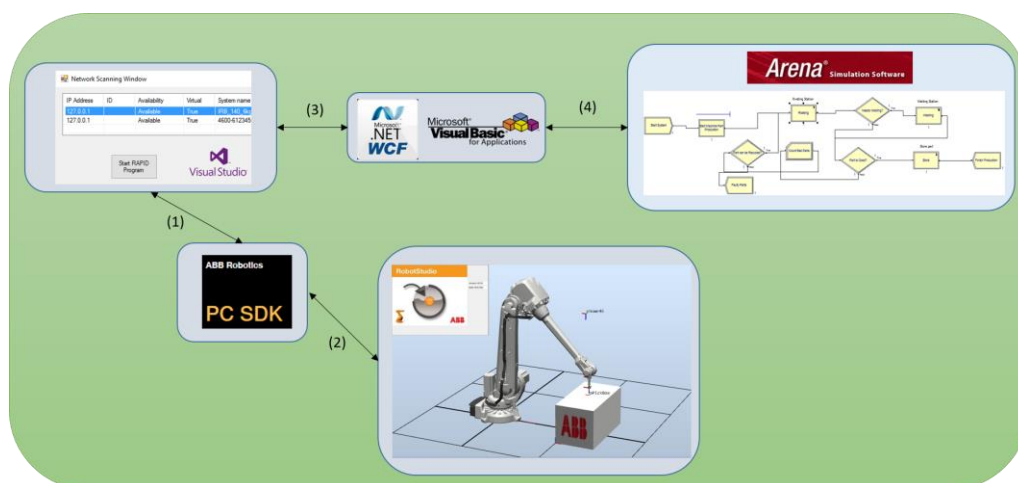


Figura 3.22: Aspeto geral da plataforma de simulação desenvolvida

Tendo em conta as ligações apresentadas na figura anterior, o funcionamento agregado da plataforma pode ser descrito nestes passos: Caso o utilizador apenas pretenda executar uma simulação ao nível dos nós locais, estes utilizará a *interface* de controlo para realizar uma simulação

simples; as bibliotecas da ABB (1) utilizadas no ambiente de desenvolvimento RobotStudio permitirão que o utilizador consiga aceder ao controlador virtual do robot, conseguindo assim simular as tarefas de simulação que deseje (2).

Caso o utilizador pretenda realizar uma simulação do comportamento agregado da plataforma, este deverá utilizar na mesma a *interface* de controlo; porém, neste caso a *interface* irá realizar uma ligação a um serviço de comunicação *web* (3), que por sua vez irá comunicar com o ARENA através dos blocos de comunicação desenvolvidos em VBA. Após o utilizador executar a simulação no modelo desenvolvido no ARENA, os blocos VBA serão ativados de forma a estabelecer uma comunicação com o serviço de comunicação *web* criado previamente.

Após estabelecer a comunicação com o serviço (4), o bloco VBA irá enviar uma variável de controlo para o serviço que por sua vez irá comunicá-la à aplicação responsável pela *interface* de controlo (através do serviço de comunicação criado previamente), sendo que cada bloco de processo envia uma variável com um valor diferente, consoante a tarefa a ser simulada. Nesta altura, o modelo no ARENA entra em modo de espera enquanto a simulação relativa ao bloco de processamento é executada.

A aplicação desenvolvida no Visual Studio, correspondente ao nível de manufatura, ao receber a variável, irá atuar sobre o nível máquina de acordo com a variável recebida, isto é, irá mandar executar uma tarefa de simulação diferente consoante o valor da variável de controlo que recebeu (1)(2). Depois de executada a tarefa, o nível de máquina irá comunicar o sucesso da execução da tarefa ao nível de manufatura, que por sua vez envia uma mensagem de retorno para o nível estratégico que irá prosseguir a simulação. Após o término da simulação, será gerado um relatório estatístico por parte do ARENA.

4 Verificação e validação da plataforma de simulação

A validação da solução proposta passa, numa fase inicial, pelos testes de forma individual dos sistemas implementados. Em seguida, os sistemas implementados serão testados de forma aglomerada, podendo assim aferir as diferenças relativas à simulação de forma individual. Posteriormente, será discutida a validade da hipótese formulada.

4.1 Verificação do funcionamento da infraestrutura de interoperabilidade

Conforme descrito no capítulo anterior, a comunicação entre a aplicação de controlo e o modelo de simulação global foi efetuada com recurso a um serviço de comunicação (*web service* criado através da *framework* WCF da Microsoft).

Este serviço irá permitir a troca de mensagens entre dispositivos que não se encontrem ligados na mesma rede interna, sendo assim possível estabelecer uma linha de comunicação entre o pacote de simulação ARENA e a aplicação de controlo. Neste serviço foram implementadas diversas funções de tratamento de dados de modo a estabelecer a comunicação entre as duas entidades supracitadas. Na figura 4.1 é possível observar o comportamento de uma simples função criada na lógica de *request-response* – depois de se introduzir a palavra “User” na *string* “str”, o serviço irá responder com a *string* “Hello User”.

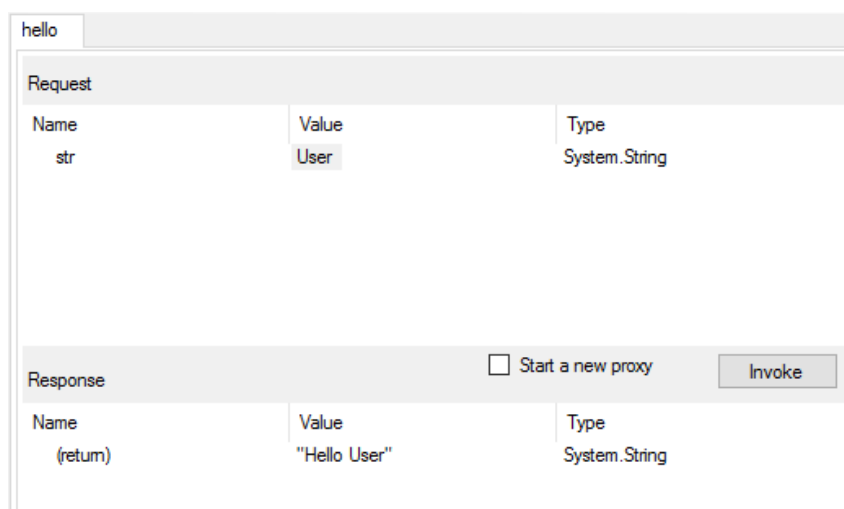


Figura 4.1: Função de teste do serviço de comunicação *web* criado.

4.1.1 Gestão dos nós de simulação

De forma a realizar-se a validação da gestão dos nós locais, foi testada a *interface* de controlo implementada previamente no seu modo mais simples: escolher um controlador válido.

Inicialmente, o utilizador escolhe através da *interface* de controlo qual o controlador que quer simular. A aplicação de controlo apresenta uma lista de todos os controladores ativos que estejam ligados na mesma rede de trabalho, como é possível ver na figura 4.2.

IP Address	ID	Availability	Virtual	System name	RobotWare version	Controller name
127.0.0.1		Available	True	IRB_140_6kg_...	6.2.0.1	QWERTY
127.0.0.1		Available	True	4600-612345MH	6.2.0.1	QWERTY

Figura 4.2: Janela de seleção de controlador – nesta janela, presente na aplicação de controlo, é possível escolher um controlador remoto

O utilizador é então alertado da escolha do controlador através de uma mensagem de controlo (Figura 4.3).

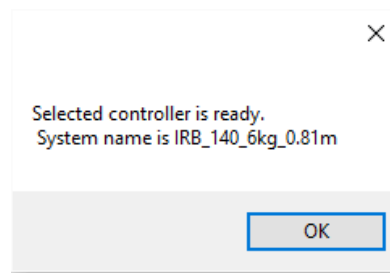


Figura 4.3: Mensagem de aviso gerada pela aplicação de controlo após a escolha do controlador

4.1.2 Simulação dos nós locais

Depois de escolhido o controlador, o utilizador pode mandar executar uma tarefa de simulação associada ao controlador escolhido através da opção *Start RAPID Program*, como é possível ver na figura 4.4.

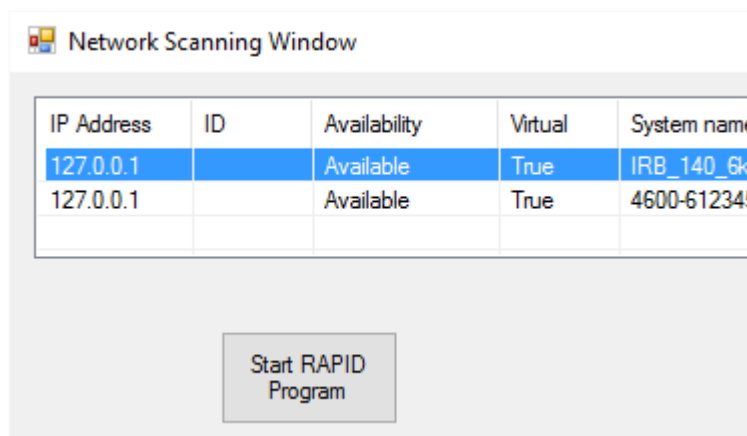


Figura 4.4: Comando de execução do controlador – após a escolha do controlador, é possível executar a tarefa associada

Depois de executada a tarefa de simulação, o utilizador recebe uma mensagem de aviso (Figura 4.5) quando a simulação for terminada, bem como o tempo que esta demorou.

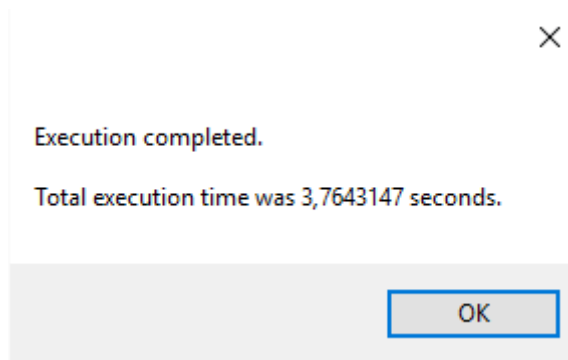


Figura 4.5: Mensagem de aviso após a execução das tarefas do controlador. Esta mensagem informa o utilizador da conclusão da tarefa bem como do tempo que esta demorou.

Cada um dos controladores apresentados representa uma estação de trabalho diferente no pacote de simulação RobotStudio. A figura 4.6 representa o robot “*IRB_140_6kg_0.81m*” da ABB em operação.

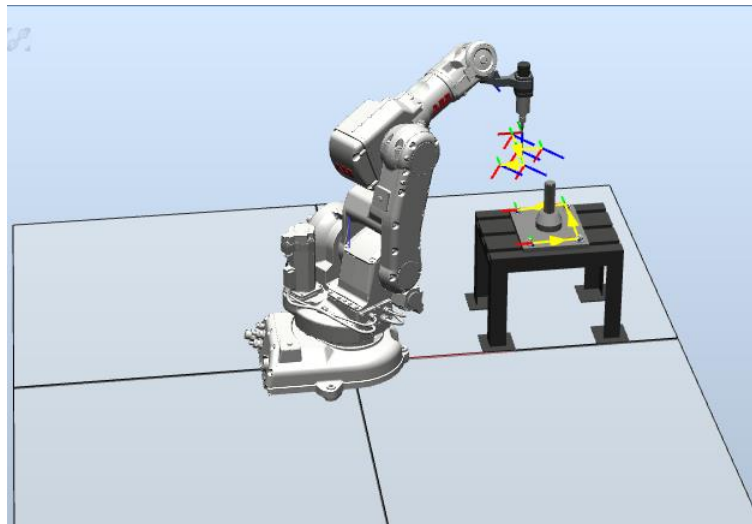


Figura 4.6: Robot “*IRB_140_6kg_0.81m*” em operação

Na figura 4.7 é possível observar o robot “*4600-612345MH*”, também da ABB, em operação.

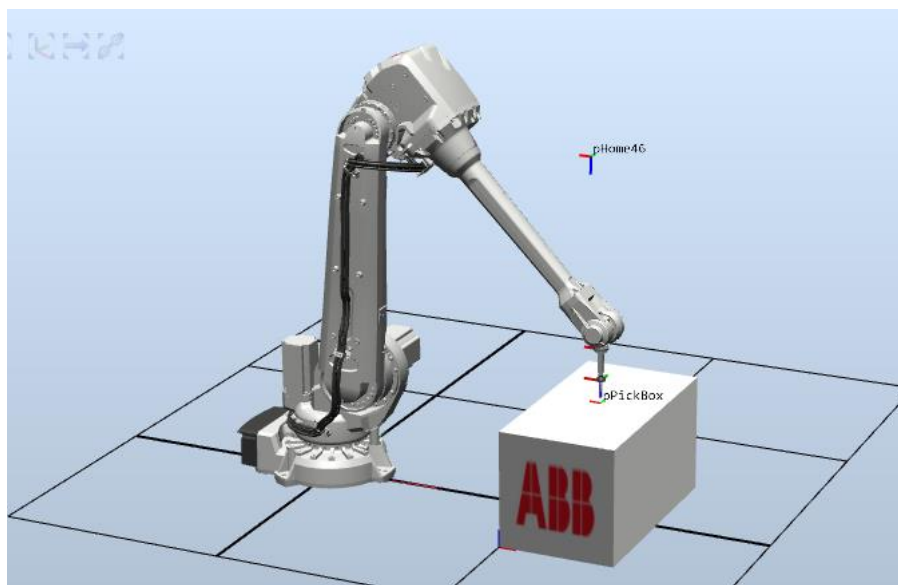


Figura 4.7: Robot "4600-612345MH" em operação

4.1.3 Simulação dos níveis estratégicos

O passo seguinte da verificação consistiu na criação de um modelo de simulação de processos de manufatura, utilizando para tal o pacote de simulação ARENA. Inicialmente foi criado um modelo que realiza uma simulação interna, isto é, sem contato com outros intervenientes do sistema.

Este modelo de simulação – apresentado na figura 4.8 - irá servir de base de comparação com a plataforma de simulação constituída pelas diversas entidades referidas anteriormente.

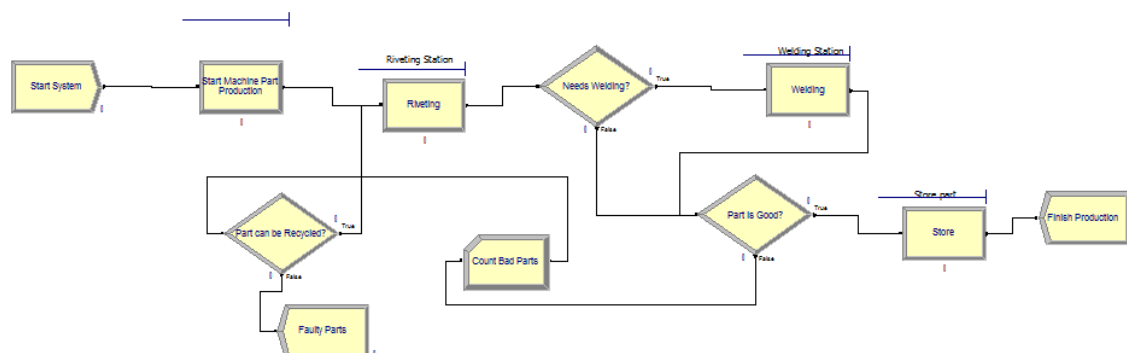


Figura 4.8: Modelo de simulação interno criado no pacote de simulação ARENA.

Nesta simulação são produzidas dez peças genéricas que serão tratadas por diversas estações de trabalho. Ao longo de todo o modelo existem blocos de decisão que irão influenciar o percurso da peça - se precisa de ser tratada por uma estação de trabalho específica ou se está em boas condições. No final da simulação é gerado um relatório de dados sobre a simulação efetuada.

No sistema em questão foram definidos quatro estações de manufatura: produção, rebtagem, soldadura e armazenamento. A estas estações estão associados contadores que irão contabilizar

a utilização de cada estação de manufatura. Os resultados da execução deste modelo são explicados na secção seguinte.

4.2 Simulação de um SDM

4.2.1 Simulação do modelo em ARENA

Depois da realização da simulação com o modelo representado anteriormente no subcapítulo 4.1.3, obtiveram-se alguns dados relevantes relativos à simulação efetuada, que estão representados na tabela 4-1:

Tabela 4-1: Tabela comparativa de tempos de processamento da simulação efetuada no ARENA

Tempo (em minutos)	Valor médio	Valor mínimo	Valor máximo
VA	4.5417	0.8631	12.9919
NVA	0.00	0.00	0.00
Espera	7.0166	0.1629	19.6448
Transferência	0.00	0.00	0.00
Outros	0.00	0.00	0.00
Total	11.5584	2.6850	25.9919

Através destes dados é possível verificar que o tempo necessário para produzir as dez peças genéricas cifra-se em 11.584 minutos, sendo que:

- VA (Value Added) corresponde ao tempo que é acumulado quando uma entidade é afetada por um atraso em processos de valor acrescentado.
- NVA (Non-Value Added) corresponde ao tempo que é acumulado quando uma entidade é afetada por um atraso em processos sem valor acrescentado.
- Espera corresponde ao atraso que ocorre quando a entidade entra num processo designado como um processo de espera ou quando esta permanece numa fila de espera
- Transferência corresponde ao atraso que ocorre quando a entidade entra num processo designado como processo de transferência

O tempo total de processamento será igual à soma de todos os tempos referidos previamente (Mo 2009).

Em relação ao tempo utilização das estações de manufatura, o relatório de dados também fornece alguns dados relativos à utilização das mesmas, como é possível ver na tabela 4-2.

Tabela 4-2: Tabela comparativa dos tempos de utilização de cada estação

Utilização instantânea (em minutos)	Tempo médio	Tempo mínimo	Tempo máximo
Estação de produção	0.02239074	0.00	1.00
Estação de rebtagem	0.00316970	0.00	1.00

Estação de soldadura	0.02985350	0.00	1.00
Estação de armazenamento	0.03542104	0.00	1.00

Quanto à utilização de cada estação de manufatura e ao número de peças defeituosas, foram consideradas:

- Doze peças foram processadas pela estação de rebitagem
- Nove peças foram armazenadas
- Duas peças foram soldadas
- Três peças foram consideradas defeituosas

4.2.2 Resultados da simulação dos nós locais

Aquando da verificação da simulação dos nós locais, foram feitas algumas medições temporais em dez iterações de cada tarefa de simulação – designadas como “rebitagem”, “soldadura” e “armazenamento” – de forma a poder estabelecer um padrão de comparação com a simulação do comportamento agregado que será demonstrada nas próximas secções. De forma a tornar os tempos de execução mais realistas, foi introduzido um tempo de espera aleatório (balizado entre os 500 e os 3000 milissegundos) na simulação do robot. Este tempo será representativo do tempo de espera de uma estação de trabalho real, consoante a tarefa que desempenha, a carga que usa, entre outras condicionantes.

Assim sendo, as medições temporais para a atividade “Rebitagem” são apresentadas na figura 4.9:

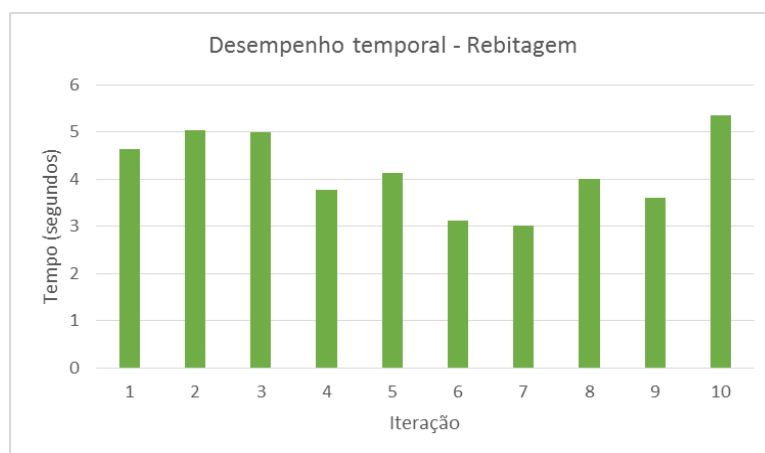


Figura 4.9: Gráfico comparativo do desempenho temporal da atividade “Rebitagem”.

Em relação às medições temporais relativas à atividade “Soldadura”, os resultados são apresentados na figura 4.10:

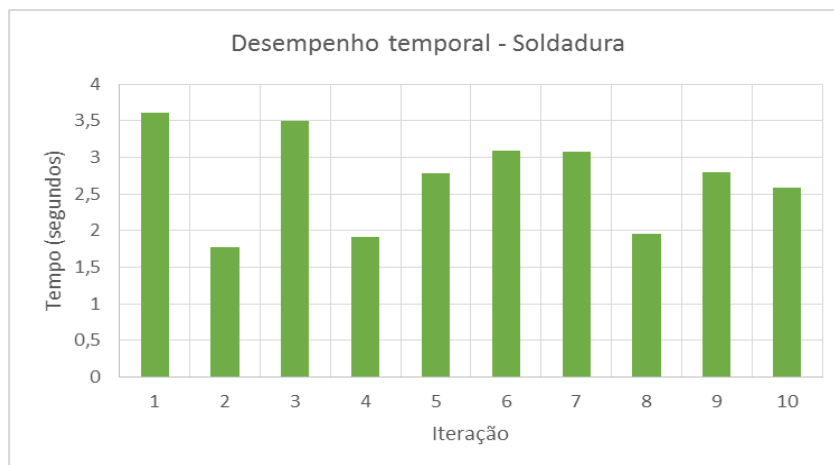


Figura 4.10: Gráfico comparativo do desempenho temporal da atividade “Soldadura”.

Por fim, os resultados temporais relativos à atividade “Armazenamento” são apresentados na figura 4.11:

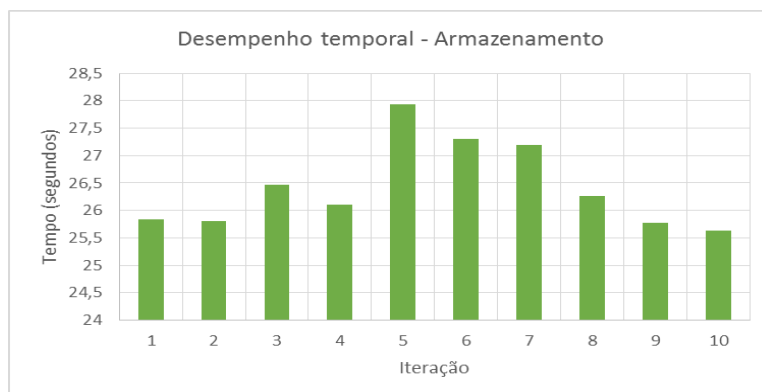


Figura 4.11: Gráfico comparativo do desempenho temporal da atividade “Armazenamento”.

Também o pacote de simulação RobotStudio apresenta algumas informações relativas à simulação efetuada, tais como o início e o fim da simulação, como é possível ver na figura 4.12.

Output		
Show messages from: All messages	Time	Category
IRB_140_6kg_0.81m (Station): 10052 - Regain start	07/09/2016 16:54:21	Event Log
IRB_140_6kg_0.81m (Station): 10053 - Regain ready	07/09/2016 16:54:21	Event Log
IRB_140_6kg_0.81m (Station): 10156 - Program restarted	07/09/2016 16:54:21	Event Log
IRB_140_6kg_0.81m (Station): 50024 - Corner path failure	07/09/2016 16:54:22	Event Log
IRB_140_6kg_0.81m (Station): 10122 - Program stopped	07/09/2016 16:54:23	Event Log

Figura 4.12: Relatório gerado após o término da simulação no pacote de simulação RobotStudio

4.2.3 Simulação da plataforma de simulação agregada

De forma a aferir o comportamento agregado dos sistemas envolvidos, foi testada a plataforma implementada anteriormente. O modelo de simulação (Figura 4.13) baseia-se no modelo desen-

volvido no pacote de simulação ARENA apresentado na secção 4.1.3 mas apresenta algumas alterações, tais como a introdução de blocos VBA (*Visual Basic for Applications*). Estes blocos programáveis, que representam estações de manufatura, irão interagir com o serviço de comunicação *web* implementado anteriormente, que por sua vez irá comunicar com a aplicação de controlo, que irá comandar as estações de trabalho no pacote de simulação RobotStudio.

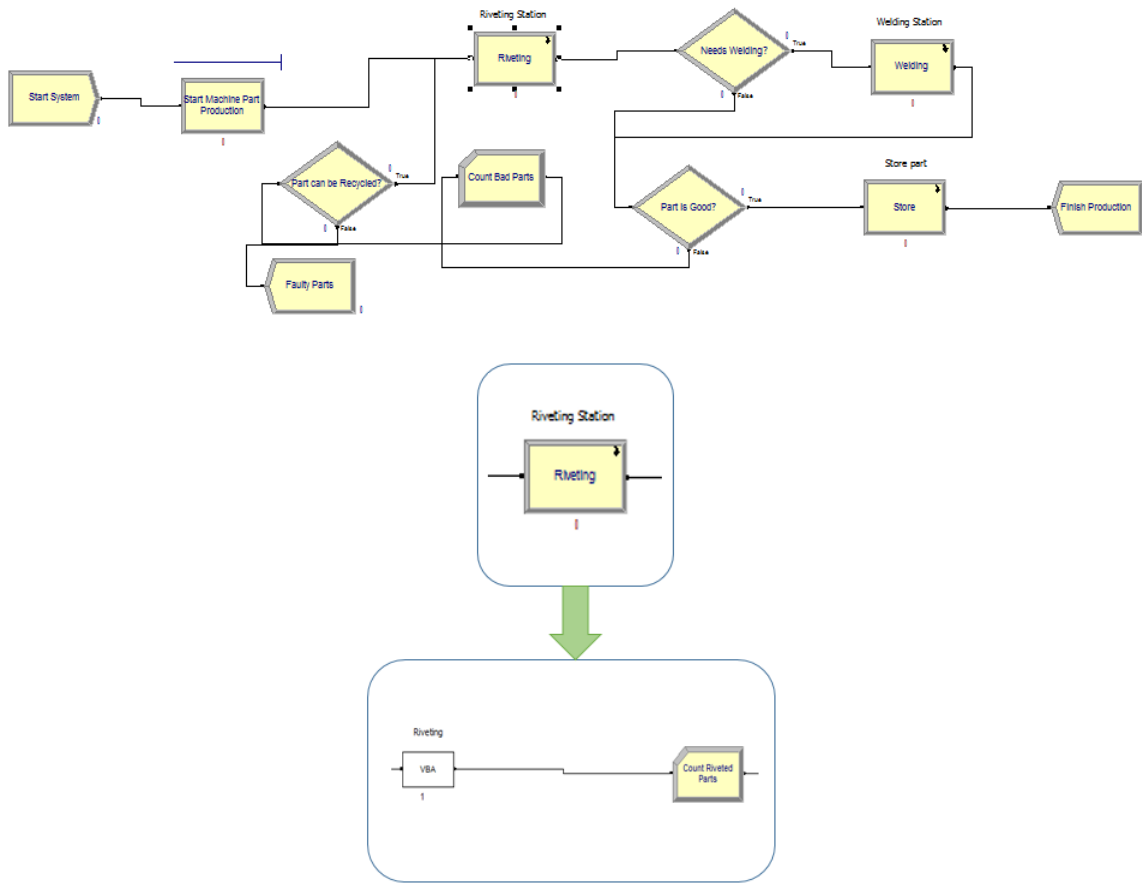


Figura 4.13: Modelo de simulação utilizado para o estudo do comportamento agregado do sistema.

Depois de efetuada uma simulação nos mesmos moldes da simulação efetuada no subcapítulo 4.1.3, obtiveram-se alguns dados relativos ao tempo de processamento, como é possível ver na tabela 4-3:

Tabela 4-3: Tabela comparativa dos tempos de processamento no modelo de simulação agregado

Tempo (em minutos)	Valor médio	Valor mínimo	Valor máximo
VA	5.2772	4.0883	7.3277
NVA	0.00	0.00	0.00
Espera	18.2448	0.00	33.0942
Transferência	0.00	0.00	0.00
Outros	0.00	0.00	0.00
Total	23.5220	4.7983	39.3120

Uma vez que nesta simulação os blocos relativos aos processos de manufatura foram substituídos por blocos que irão comandar uma estação de trabalho noutra ambiente de simulação, foi efetuada uma medição temporal do comportamento agregado de forma a poder-se efetuar uma comparação com a simulação dos nós locais. O resultado pode ser observado na figura 4.14.

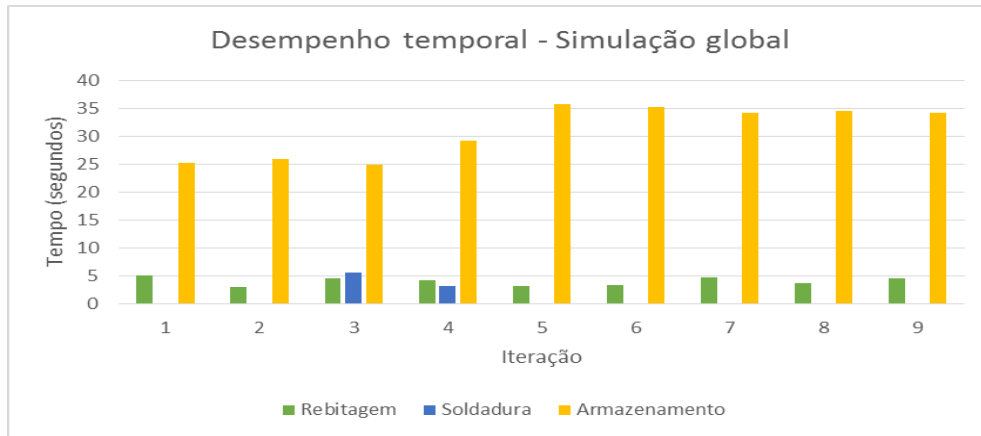


Figura 4.14: Gráfico comparativo do desempenho temporal das estações de trabalho aquando do teste do comportamento agregado

Em relação à utilização de cada estação de manufatura e ao número de peças defeituosas, o resultado foi o seguinte:

- Dez peças foram tratadas pela estação de rebitagem
- Quatro peças foram tratadas pela estação de soldadura
- Oito peças foram tratadas pela estação de armazenamento
- Duas peças foram consideradas defeituosas

4.2.4 Comparação dos resultados obtidos

Após terem sido efetuados os dois ensaios determinantes da plataforma de simulação – um de forma isolada, onde o ARENA e o RobotStudio eram intervenientes únicos e outro onde todos os componentes da plataforma estavam envolvidos – é possível observar que os tempos de processamento nos dois casos apresentam alguma disparidade. No primeiro caso, tratando-se de uma simulação realizada por apenas um único pacote de simulação – fosse ele o ARENA ou o RobotStudio – foi possível atingir tempos de processamento médios na ordem dos onze minutos. Já na simulação da plataforma agregada, os tempos médios de processamento subiram para os vinte e três minutos. Esta disparidade de resultados pode ser explicada pelo facto que no segundo ensaio de simulação, ao contrário do primeiro ensaio onde apenas se utilizou um único pacote de simulação de forma singular, foram utilizados dois pacotes de simulação em simultâneo, bem como serviços de comunicação *web* para estabelecer a interoperabilidade no sistema, sendo que modelo de simulação do nível estratégico passou a ser afetado diretamente pelas simulações dos níveis inferiores. Uma vez que a simulação mais específica dos nós locais é afetada por um tempo de espera aleatório – como fora mencionado na secção 4.2.2 – este valor veio

afetar ainda mais o tempo de processamento da simulação da plataforma agregada, aumentando assim a diferença dos tempos médios de processamento entre os dois ensaios de simulação.

4.3 Análise dos resultados

Após as validações efetuadas à plataforma desenvolvida, é possível observar que os requisitos funcionais previamente identificados no segundo capítulo foram implementados de forma correta. Foi possível observar e simular os diversos comportamentos de um SDM, quer ao nível mais local, quer ao nível mais global.

No entanto, a simulação de SDM nos moldes deste trabalho, apesar de ser relativamente fácil e pouco dispendiosa de se fazer, não é propriamente intuitiva para um operador comum, assim como não é muito aconselhada para sistemas altamente complexos e que necessitem de muita capacidade de processamento.

Assim sendo, resta então responder à questão de pesquisa proposta no primeiro capítulo. Nela foi levantada a questão se seria possível simular de forma abrangente um SDM através da criação de uma plataforma composta por diversos pacotes de simulação. Após a elaboração deste trabalho, onde foram combinados dois pacotes de simulação de forma a criar uma plataforma de simulação de SDM, foi possível verificar o comportamento de um sistema distribuído por diversos níveis, quer verticais como horizontais. Através da plataforma desenvolvida foi assim possível simular os comportamentos de um SDM, tanto a um nível mais global como a um nível local. Tendo em conta que os pacotes de simulação utilizados apresentam arquiteturas diferentes, a interoperabilidade – fornecida por serviços de comunicação *web* – foi crucial para o funcionamento da plataforma de simulação, permitindo assim que arquiteturas diferentes – que podem estar em localizações díspares - comunicassem entre si, representando assim fielmente um SDM real, onde é essencial existir uma boa comunicação entre os diferentes agentes do sistema, independentemente da arquitetura apresentada ou sua localização. Posto isto, é possível afirmar que a questão de pesquisa levantada no primeiro capítulo pôde ser respondida por este trabalho.

5 Conclusões

5.1 Síntese do trabalho

Durante este trabalho, constatou-se que os sistemas de manufatura estão cada vez mais complexos, sendo cada vez mais difícil prever o seu desempenho de uma forma rápida e eficaz. Além disso, é preciso também ter em conta que a observação do comportamento de sistemas distribuídos implica executar processos em diferentes níveis hierárquicos e muitas vezes, em diferentes localizações geográficas.

Conforme se viu, uma das alternativas consiste então na simulação deste tipo de sistemas. Constatou-se que existem no mercado diversos pacotes de simulação industriais, no entanto estes não são capazes de simular de forma agregada todos os aspetos de um sistema de manufatura distribuído. De forma a resolver esta questão, colocou-se a hipótese de combinar diversas ferramentas de simulação de forma a realizar simulações de sistemas mais complexos, e com isso, conseguir fazer previsões acerca do seu desempenho. De facto, esta combinação de diferentes ferramentas de simulação começa a tornar-se cada vez mais possível devido ao facto de muitos dos pacotes de simulação presentes no mercado oferecem funcionalidades ao nível de interoperabilidade.

Assim sendo, propôs-se neste trabalho a criação de uma plataforma de simulação, composta por diversos pacotes de simulação existentes no mercado e que sejam capazes de interagir entre si através de uma infraestrutura de interoperabilidade, composta quer por valências dos próprios pacotes de simulação, quer por serviços de comunicação *web*.

Antes da implementação da plataforma, foi realizado um levantamento teórico com o intuito de clarificar alguns conceitos relativos às áreas de simulação e manufatura, servindo esta informação para delinear uma arquitetura de suporte para a plataforma em questão. Definiram-se vários níveis (por exemplo, estratégico, manufatura e máquina), que englobam tanto as ações a um nível mais global de um sistema distribuído (nível estratégico), como as ações que decorrem a um nível mais local (nível máquina). De forma a que as interações entre os diversos fossem bem sucedidas, considerou-se então uma infraestrutura de interoperabilidade que fosse capaz de estabelecer canais de comunicação entre as diferentes arquiteturas presentes neste sistema.

Para criar a infraestrutura de interoperabilidade foram usadas algumas tecnologias tais como:

- O PC SDK da ABB que permitiu a interligação de uma aplicação (*interface* de utilizador) desenvolvida no ambiente de desenvolvimento Visual Studio da Microsoft com o pacote de simulação RobotStudio da ABB. Foi através das bibliotecas fornecidas por esta tecnologia que se conseguiu aceder a funções nativas dos controladores virtuais dos robots, permitindo assim executar a simulação um processo de manufatura ao nível local.
- O serviço de comunicação *web* criado através da *framework* WCF da Microsoft, possibilitou a comunicação entre o pacote de simulação ARENA e a aplicação de controlo responsável pela simulação ao nível dos nós locais

A combinação da infraestrutura de interoperabilidade supracitada com as ferramentas de simulação referidas ao longo do trabalho (RobotStudio e ARENA) permitiu assim implementar uma plataforma capaz de simular o comportamento agregado de um sistema de manufatura. Destacam-se desta plataforma três elementos principais:

- A *interface* de controlo, que é responsável pela interação do utilizador com o sistema. Esta permite a execução de uma tarefa de simulação de um nó local (através da interligação com o RobotStudio via PC SDK) assim como a automatização do funcionamento da plataforma.
- O pacote de simulação ARENA, que permite criar e simular um modelo de simulação ao nível estratégico. Este modelo de simulação pode ser composto por blocos VBA que permitem a comunicação com um serviço *web*, estabelecendo uma ponte de comunicação com outras arquiteturas.
- O pacote de simulação RobotStudio, que permite criar e simular diversos ambientes industriais. Na plataforma desenvolvida esta ferramenta foi utilizada para a simulação de um nó local, como referido anteriormente.

Considerando a arquitetura proposta para os diferentes níveis (estratégico, manufatura e máquina), foi assim possível simular um nível global (estratégico) com recurso ao ARENA. Este nível global é composto por diversas interações (como por exemplo, tarefas de manufatura numa linha de produção) nos níveis mais locais (manufatura e máquina) que foram simuladas com recurso ao RobotStudio.

5.2 Resultados obtidos

Depois da elaboração deste trabalho foram atingidos alguns resultados práticos, dos quais se destacam:

- Numa fase inicial de estudo teórico, foram clarificados alguns conceitos relativos às áreas da manufatura e simulação
- Após o levantamento teórico, foi elaborado o levantamento dos requisitos funcionais do sistema. Começou-se também a delinear uma arquitetura de suporte para a plataforma de simulação considerada inicialmente
- Tendo em conta a arquitetura considerada, foram implementados diversos processos capazes de dar resposta à simulação de níveis locais e globais de forma separada
- De forma a criar uma plataforma agregadora na sua plenitude, foi desenvolvida uma infraestrutura de interoperabilidade capaz de interligar os diversos agentes do sistema
- Após a junção de todas as estruturas mencionadas previamente, foi assim possível obter uma plataforma de simulação de SDM de forma agregada
- Depois de alguns testes ao sistema desenvolvido, foi possível aferir a sua validade no que à simulação de SDM diz respeito, respondendo assim à questão de pesquisa levantada no primeiro capítulo

5.3 Trabalho futuro

No âmbito deste trabalho, algum do trabalho futuro que seria interessante abordar passaria pela interoperabilidade da plataforma criada com pacotes de simulação diferentes, assim como uma maior utilização de *web services* no que às tarefas de simulação diz respeito – seria assim possível suportar todo um sistema de simulação numa base *web* – como uma plataforma *cloud* – sendo assim possível uma verdadeira interoperabilidade a nível global entre diversos Sistemas Distribuídos de Manufatura.

Este trabalho é de uma natureza interdisciplinar, que envolveu áreas da engenharia distintas, por exemplo, a simulação, os SDM e a informática. Além disso, o desenvolvimento deste trabalho revestiu-se de significativos desafios tecnológicos. Consequentemente, recomenda-se que este trabalho, portanto numa fase futura, seja assumido por uma equipa constituída por peritos de distintas áreas, conforme atrás referidos.

6 Bibliografia

- ABB, 2016. Tutorials for RobotStudio - RobotStudio - ABB Robotics. Available at: <http://new.abb.com/products/robotics/robotstudio/tutorials>.
- Alex Adamtsev, 2016. Push model simulated in Arena software -- Alex Adamtsev. Available at: http://cyberspaceandtime.com/Push_model_simulated_in_Arena_software_-_Alex_Adamtsev/7XBnpoHQRNU.video.
- Alonso, G. et al., 2004. *Web Services: Concepts, Architectures and Applications*, Available at: <http://www.amazon.com/dp/3642078885>.
- AnyLogic, 2016. Online Help — AnyLogic Simulation Software. Available at: <http://www.anylogic.com/anylogic/help/> [Accessed January 22, 2017].
- Banks, J. et al., 2004. Discrete-Event System Simulation. *PrenticeHall international series in industrial and systems engineering*, p.624. Available at: <http://books.google.com/books?id=wWFRAAAAMAAJ>.
- BCAG, 1997. World Class Competition. Available at: <https://www.makino.com/resources/case-studies/World-Class-Competition/129/>.
- Bleicher, F. et al., 2014. Co-simulation environment for optimizing energy efficiency in production systems. *CIRP Annals - Manufacturing Technology*, 63(1), pp.441–444. Available at: <http://dx.doi.org/10.1016/j.cirp.2014.03.122>.
- BusinessDictionary, 2016. What is a model? definition and meaning - BusinessDictionary.com. Available at: <http://www.businessdictionary.com/definition/model.html>.
- Chituc, C. & Restivo, F., 2009. Challenges and trends in distributed manufacturing systems: are wise engineering systems the ultimate answer. ... *Symposium on Engineering Systems MIT*, ..., pp.1–15. Available at: http://www.researchgate.net/publication/228615464_Challenges_and_Trends_in_Distributed_Manufacturing_Systems_Are_wise_engineering_systems_the_ultimate_answer/file/9fd50a2b5f560990.pdf.
- Dasun Hegoda, 2014. Let's make a SOAP request from command line(curl)? Available at: <http://dasunhegoda.com/make-soap-request-command-line-curl/596/>.
- EPIC, 2013. U.S. Manufacturing: A History | Visual.ly. Available at: <http://visual.ly/us-manufacturing-history>.
- Fraunhofer IFAM, 2014. Circuits and sensors direct from the printer - Fraunhofer IFAM. Available at: http://www.ifam.fraunhofer.de/en/Press_Releases/Circuits_Sensors_Printer.html.
- Fujimoto, R.M., 2001. Parallel and distributed simulation systems. In *Simulation Conference, 2001. Proceedings of the Winter*. pp. 147–157 vol.1.
- Gregory, F., 1998. Cause, Effect, Efficiency and soft Systems Models. Available at: http://www.geocities.ws/f_h_gregory/cause_eff_eff_txt.html.
- Groover, M.P., 2008. Automation, Production Systems, and Computer-integrated Manufacturing. In *Automation, Production Systems, and Computer-integrated Manufacturing*. p. 290. Available at: http://bks4.books.google.com.my/books?id=I-dQPgAACAAJ&printsec=frontcover&img=1&zoom=1&imgtk=AFLRE728Hj5LmM5wYsjo-7zW_yiENRUWIJM9kgb-

Kd5DpsiySGkEnvHdugm7zCjFyRowBqIt1xrYRfta506naFXOJ74crbmXzzGPlj7M-PUAzfup_LHNI6GlwcmT6VDcQW0CIZTyj9Wm.

- IEEE, 2000. IEEE Std 1516-2000. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*.
- Kieran, C. et al., 2007. *Simulation Modeling with SIMUL8*,
- Kuhl, F., Weatherly, R., & Dahmann, J., 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*.
- KUKA, 2011. KUKA Flexible Manufacturing Systems - application innovation. Available at: <http://www.kuka-systems.com/NR/exeres/4A50C9C8-4549-4918-92EE-08A5F0AB03C7>.
- Lakatos, E. & Marconi, M.D.A., 1991. Metodologia científica. *Metodologia Científica*, pp.1–20. Available at: http://www.dem.fmed.uc.pt/Bibliografia/Livros_Educacao_Medica/Livro27.pdf.
- Lamborghini, 2011. Lamborghini to Drop Manual Transmissions - Motor Trend. Available at: <http://www.motortrend.com/news/lamborghini-to-drop-manual-transmissions/>.
- Laumer, J., 2008. Big Blue Making Supply Chains Into Green Super-Models: No Floppy Disks Required. : TreeHugger. Available at: <http://www.treehugger.com/corporate-responsibility/big-blue-making-supply-chains-into-green-super-models-no-floppy-disks-required.html>.
- Law, A.M. & Kelton, W.D., 2000. Simulation Modeling and Analysis. *ACM Transactions on Modeling and Computer Simulation*, 2(2), pp.1370–1375. Available at: <http://content.imamu.edu.sa/Scholars/it/VisualBasic/306537540.pdf>.
- Malan, R. & Bredemeyer, D., 2001. Functional Requirements and Use Cases. *White Paper*, (8/3/01), pp.1–10.
- Maria, A., 1997. Introduction to modelling and simulation. *Winter Simulation Conference*, pp.7–13.
- Marko Beric, 2010. Pharmaceutical Production Stock Photos - Image: 33373373. Available at: <https://www.dreamstime.com/stock-photos-pharmaceutical-production-drug-manufacturing-process-factory-image33373373>.
- Meng, X., 2002. When is Simulation the Appropriate Tool? Available at: <https://www.eg.bucknell.edu/~xmeng/Course/CS6337/Note/master/node2.html> [Accessed January 22, 2017].
- Michael George, 2002. *Lean Six Sigma: Combining Six Sigma Quality with Lean Production Speed*,
- Mo, J.P.T., 2009. Modelling and simulation of national electronic product code network demonstrator project BT - 7th Workshop on E-Business, WEB 2008, December 13, 2008 - December 13, 2008. , 22 LNBIP, pp.60–70.
- MODSIM, 2016. Mineral Technologies Inc. - MODSIM. Available at: <http://www.mineraltech.com/MODSIM/>.
- Montaqim, A., 2015. Offline programming software for industrial robots from RoboDK offers hundreds of virtual industrial robots from top robotics companies | Robotics & Automation News. Available at: <https://roboticsandautomationnews.com/2015/07/14/offline-programming-software-robotdk-offers-hundreds-of-virtual-industrial-robots-from-top-robotics-companies/540/> [Accessed January 26, 2017].

- Nagurney, A., 2012. RENEW: How the US Can Compete and Win in Global Supply Chains. Available at: <http://annanagurney.blogspot.pt/2012/01/how-us-can-compete-and-win-in-global.html> [Accessed January 21, 2017].
- Padhya, M., 2013. SIMULATION. Available at: <http://www.slideshare.net/MJ4950/simulation-25117717> [Accessed January 21, 2017].
- Pau Casas, 2009. Simulation tools | Ph.D. Pau Fonseca i Casas. Available at: <http://www-eio.upc.es/~pau/?q=node/20>.
- Paula Bernier, 2010. Web 2.0 Application Development Takes a REST. Available at: <http://www.tmcnet.com/voip/0310/web-2-application-development-takes-a-rest.htm>.
- Pautasso, C., 2008. RESTful Web Services. , pp.1–20.
- Prensky, M., 2002. Why not simulation. *Retrieved August, 6*, pp.1–8. Available at: <http://www.marcprensky.com/writing/Prensky - Why NOT Simulation.pdf>.
- promatics, 2016. Project Management | Software Development Process. Available at: <http://www.promaticsindia.com/project-management> [Accessed February 7, 2017].
- ProModel, 2016. ProModel - Better Decisions Faster. Available at: <https://www.promodel.com/services/university>.
- Queirós, D., 2016. Utilização do software RobotStudio na programação off-line de células robóticas de quinagem.
- RoboDK, 2016. RoboDK help guide. Available at: <https://www.robodk.com/help> [Accessed January 26, 2017].
- Rosen, M., 2008. *Applied SOA Service-Oriented Architecture and Design Strategies*,
- Rüßmann, M. et al., 2015. Industry 4.0. *The Boston Consulting Group*, p.20.
- Schwendimann, B., 2010. Proto-Knowledge: What is the difference between a simulation and a model? [Updated]. Available at: <http://proto-knowledge.blogspot.pt/2010/12/what-is-difference-between-simulation.html>.
- Shenyang Machine Tool, 2010. Liaoning's equipment manufacturers turn to clusters for expansion. Available at: http://www.chinadaily.com.cn/cndy/2010-11/30/content_11627406.htm.
- Siemens, 2016. RobotExpert Free Trial: Siemens PLM Software. Available at: https://www.plm.automation.siemens.com/en_us/products/tecnomatix/free-trial/robotexpert.shtml [Accessed January 30, 2017].
- SIMUL8, 2016. SIMUL8 Simulation Software - for visual process simulation modeling. Available at: <http://www.simul8.com/>.
- Smith, P.R. & Sarfaty, R., 1993. Creating a strategic plan for configuration management using Computer Aided Software Engineering (CASE) tools.
- Strassburger, S., Schmidgall, G. & Haasis, S., 2003. Distributed Manufacturing Simulation as an Enabling Technology for the Digital Factory. *Journal of Advanced Manufacturing Systems*, 2(1), pp.111–126.
- Suda, B., 2003. SOAP Web Services. *Retrieved June*. Available at: <http://pop.suda.co.uk/publications/MSc/brian.suda.thesis.pdf>.
- Uygun, Ö., Öztemel, E. & Kubat, C., 2009. Scenario based distributed manufacturing simulation using HLA technologies. *Information Sciences*, 179(10), pp.1533–1541. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0->

61449245476&partnerID=tZOtx3y1.

Anexo 1

Código C# da aplicação de controlo implementada no pacote de simulação RobotStudio:

```
using System.Windows.Forms;
using ABB.Robotics;
using ABB.Robotics.Controllers;
using ABB.Robotics.Controllers.Discovery;
using ABB.Robotics.Controllers.RapidDomain;
//(...)

namespace Tese
{
    public partial class Form1 : Form
    {
        private NetworkScanner scanner = null;

        private Controller controller = null;
        private Task[] tasks = null;
        private NetworkWatcher networkwatcher = null;
        private RapidData myRD;
        private RobTarget robtarget;

        private Stopwatch sw = new Stopwatch();

        double VarInXdir;
        double VarInYdir;
        double VarInZdir;
        string target_change;
        int time1, time2, total_time;
        Socket socket;
        Socket sender;
        Task proc_tasks;
        ServiceReference1.Service1Client client = new ServiceRefer-
ence1.Service1Client();

        void HandleFoundEvent(object sender, NetworkWatcherEventArgs e)
        {
            this.Invoke(new
            EventHandler<NetworkWatcherEventArgs>(AddControllerToListView), new Object[] {
            this, e });
        }

        private void AddControllerToListView(object sender, NetworkWatcherEven-
tArgs e)
        {
            ControllerInfo controllerInfo = e.Controller;
            ListViewItem item = new ListView-
Item(controllerInfo.IPAddress.ToString());
            item.SubItems.Add(controllerInfo.Id);
        }
    }
}
```

```

        item.SubItems.Add(controllerInfo.Availability.ToString());
        item.SubItems.Add(controllerInfo.IsVirtual.ToString());
        item.SubItems.Add(controllerInfo.SystemName);
        item.SubItems.Add(controllerInfo.Version.ToString());
        item.SubItems.Add(controllerInfo.ControllerName);
        this.listView1.Items.Add(item);
        item.Tag = controllerInfo;
    }

    public Form1()
    {
        InitializeComponent();

        this.scanner = new NetworkScanner();
        this.scanner.Scan();
        ControllerInfoCollection controllers = scanner.Controllers;
        ListViewItem item = null;
        foreach (ControllerInfo controllerInfo in controllers)
        {
            item = new ListViewItem(controllerInfo.IPAddress.ToString());
            item.SubItems.Add(controllerInfo.Id);
            item.SubItems.Add(controllerInfo.Availability.ToString());
            item.SubItems.Add(controllerInfo.IsVirtual.ToString());
            item.SubItems.Add(controllerInfo.SystemName);
            item.SubItems.Add(controllerInfo.Version.ToString());
            item.SubItems.Add(controllerInfo.ControllerName);
            this.listView1.Items.Add(item);
            item.Tag = controllerInfo;
        }

        this.networkwatcher = new NetworkWatcher(scanner.Controllers);
        this.networkwatcher.Found += new
        EventHandler<NetworkWatcherEventArgs>(HandleFoundEvent);
        //this.networkwatcher.Lost += new
        EventHandler<NetworkWatcherEventArgs>(HandleLostEvent);
        this.networkwatcher.EnableRaisingEvents = true;
    }

    //(...)
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            if (controller.OperatingMode == ControllerOperatingMode.Auto)
            {
                tasks = controller.Rapid.GetTasks();
                using (Mastership m = Mastership.Request(controller.Rapid))
                {
                    //Perform operation

                    /*if (tasks[0].Enabled == true)
                    {

                        MessageBox.Show("Task is enabled.");

                    }

                    string task_name = tasks[0].Name;

```

```

        MessageBox.Show("Task name is " + task_name);
        */
        //tasks[0].Start();

        sw.Start();

        wait_random_time();

        controller.Rapid.Start();

        while (controller.Rapid.ExecutionStatus != ExecutionSta-
tus.Stopped)
        {
        }
        sw.Stop();

        MessageBox.Show("Execution completed.\n\nTotal execution
time was " + sw.Elapsed.TotalSeconds + " seconds.");

File.AppendAllText("C:\\temp2\\execution_times_manual.txt",
sw.Elapsed.TotalSeconds.ToString() + Environment.NewLine);

        sw.Reset();
    }
}
else
{
    MessageBox.Show("Automatic mode is required to start execu-
tion from a remote client.");
}

catch (System.InvalidOperationException ex)
{
    MessageBox.Show("Mastership is held by another client." +
ex.Message);
}
catch (System.Exception ex)
{
    MessageBox.Show("Unexpected error occurred: " + ex.Message);
}

}
//(...)

```


Anexo 2

Código RAPID de uma estação de manufatura no *software* RobotStudio:

```
MODULE Module1
```

```
VAR socketdev Server1;  
VAR socketdev Client1;
```

```
VAR string filelocation:="C:/temp2/ficheiro_teste.txt";  
VAR string writestring:="olá";  
VAR iodev file;
```

```
PERS robtarget Tar-  
get_70:=[[516,125,333],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];  
PERS robtarget Tar-  
get_80:=[[537.5,75,593],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];  
PERS robtarget Tar-  
get_90:=[[606,275,293],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];  
PERS robtarget Tar-  
get_100:=[[556,275,593],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
PROC Path_20()  
MoveL Target_70,v100,z100,MyTool\WObj:=wobj0;  
MoveL Target_80,v100,z100,MyTool\WObj:=wobj0;  
MoveL Target_90,v100,z100,MyTool\WObj:=wobj0;  
MoveL Target_100,v100,z100,MyTool\WObj:=wobj0;  
ENDPROC
```

```
PROC Path_30()  
MoveL Target_70,v1000,z100,MyTool\WObj:=wobj0;  
MoveL Target_80,v1000,z100,MyTool\WObj:=wobj0;  
MoveL Target_90,v1000,z100,MyTool\WObj:=wobj0;  
MoveL Target_100,v1000,z100,MyTool\WObj:=wobj0;  
ENDPROC
```

```
PROC Path_40()  
MoveL Target_70,v500,z100,MyTool\WObj:=wobj0;  
MoveL Target_80,v500,z100,MyTool\WObj:=wobj0;  
MoveL Target_90,v500,z100,MyTool\WObj:=wobj0;  
MoveL Target_100,v500,z100,MyTool\WObj:=wobj0;  
ENDPROC
```

```
PROC pwrite()  
  
Open filelocation, file \Append;  
Write file, writestring \NoNewLine;  
Close file;
```

```
ENDPROC
```

```
PROC main ()
```

!pwrite;

!SocketCreate Server1;

!Path_20;

Path_30;

!Path_40;

ENDPROC

ENDMODULE

Anexo 3

Código em VBA no ARENA dos blocos representativos das estações de manufatura que irá comunicar com a aplicação de controlo através do *web servisse*:

```
Private Sub VBA_Block_5_Fire()

Dim val_ext As Integer
Dim syst_name As String
syst_name = "System1"
val_ext = 0

Dim addr As String
addr = "service:mexAddress=""net.tcp://localhost:7891/Test/WcfService1/Service1/Mex"","
addr = addr + "address=""net.tcp://localhost:7891/Test/WcfService1/Service1/"","
addr = addr + "contract=""IService1"", contractNamespace=""http://tempuri.org/"","
addr = addr + "binding=""NetTcpBinding_IService1"", binding-
Namespace=""http://tempuri.org/""

Dim service1 As Object
Set service1 = GetObject(addr)
service1.GetDatafromARENA (syst_name)

Do While val_ext <> 1
val_ext = service1.getDataBack
'Application.Wait (Now + TimeValue("0:00:10"))
'Sleep 10
Loop
End Sub

Private Sub VBA_Block_6_Fire()

Dim val_ext As Integer
Dim syst_name As String
syst_name = "System2"
val_ext = 0
```

```

Dim addr As String
addr = "service:mexAddress=""net.tcp://localhost:7891/Test/WcfService1/Service1/Mex"","
addr = addr + "address=""net.tcp://localhost:7891/Test/WcfService1/Service1/"","
addr = addr + "contract=""IService1"", contractNamespace=""http://tempuri.org/"","
addr = addr + "binding=""NetTcpBinding_IService1"", binding-
Namespace=""http://tempuri.org/"""

```

```

Dim service1 As Object
Set service1 = GetObject(addr)

```

```

service1.GetDatafromARENA (syst_name)

```

```

Do While val_ext <> 2

```

```

val_ext = service1.getDataBack
'Application.Wait (Now + TimeValue("0:00:10"))
'Sleep 10

```

```

Loop

```

```

End Sub

```

```

Private Sub VBA_Block_7_Fire()

```

```

Dim val_ext As Integer
Dim syst_name As String

```

```

syst_name = "System3"
val_ext = 0

```

```

Dim addr As String
addr = "service:mexAddress=""net.tcp://localhost:7891/Test/WcfService1/Service1/Mex"","
addr = addr + "address=""net.tcp://localhost:7891/Test/WcfService1/Service1/"","
addr = addr + "contract=""IService1"", contractNamespace=""http://tempuri.org/"","
addr = addr + "binding=""NetTcpBinding_IService1"", binding-
Namespace=""http://tempuri.org/"""

```

```
Dim service1 As Object
Set service1 = GetObject(addr)

service1.GetDatafromARENA (syst_name)
Do While val_ext <> 3

val_ext = service1.getDataBack
'Application.Wait (Now + TimeValue("0:00:10"))
'Sleep 10

Loop
End Sub
```


Anexo 4

Código em C# do *Web Service* criado:

```
namespace WcfService1
{
    public class Service1 : IService1
    {
        public static string send_value;

        public static int ret_val;

        public string hello (string str) {
            return "Hello " + str;
        }

        public void GetDatafromARENA(string value)
        {
            send_value=value;
        }

        public string sendDatatoApp()
        {
            string aux;

            aux = send_value;

            send_value = "";

            return aux;
        }

        public void GetDataFromApp(int val) {
            ret_val = val;
        }

        public int getDataBack() {
            return ret_val;
        }

        //novo
        public static int[] vect_pos = new int[2];

        public void getPos(int pos_org, int pos_dest)
        {
            vect_pos[0] = pos_org;
            vect_pos[1] = pos_dest;
        }
    }
}
```

```
    }  
    public int[] sendVect()  
    {  
        return vect_pos;  
    }  
}  
}
```