

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA, ARQUITETURA E URBANISMO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**Integração das Informações de Produto,
Processos e Recursos em um Sistema Produtivo
com o uso de Serviços Web**

SANTA BÁRBARA D'OESTE

2012

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA, ARQUITETURA E URBANISMO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**Integração das Informações de Produto,
Processos e Recursos em um Sistema Produtivo
com o uso de Serviços Web**

Antonio Álvaro de Assis Moura

Orientador: Prof. Dr. -Ing. Klaus Schützer

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção, da Faculdade de Engenharia, Arquitetura e Urbanismo, da Universidade Metodista de Piracicaba - UNIMEP como requisito para obtenção do Título de Doutor em Engenharia de Produção.

SANTA BÁRBARA D'OESTE

2012

**Integração das Informações de Produto,
Processos e Recursos em um Sistema Produtivo
com o uso de Serviços Web**

Antonio Álvaro de Assis Moura

Prof. Dr.-Ing. Klaus Schützer, Presidente

PPGEP – Unimep

Prof. Dr. Alexandre Tadeu Simon

PPGEP – Unimep

Prof. Dr. Fernando Celso de Campos

PPGEP – Unimep

Prof. Dr. Sérgio de Mello Schneider

UFU

Prof. Dr. Daniel Capaldo Amaral

EESC - USP

À família Henriques

Agradecimentos

A presente tese foi realizada com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES - Brasil; do Laboratório de Sistemas Computacionais para Projeto e Manufatura - SCPM e da Universidade Metodista de Piracicaba.

DE ASSIS MOURA, Antonio Álvaro. **Integração das Informações de Produto, Processos e Recursos em um Sistema Produtivo com o uso de Serviços Web.** 2012. 182f. Tese (Doutorado em Engenharia de Produção) - Faculdade de Engenharia, Arquitetura e Urbanismo, Universidade Metodista de Piracicaba - UNIMEP, Santa Bárbara d'Oeste, SP.

Resumo

A quantidade de informações envolvidas em um sistema produtivo durante o ciclo de desenvolvimento do produto cresce de maneira contínua, abrangendo mais e mais atividades. Cada uma das etapas envolvidas no sistema produtivo utiliza ferramentas de informática específicas que geram e administram seus dados de forma autônoma. Considerando que o sistema produtivo tem um arranjo que pode se dar de maneira colaborativa por meio do trabalho conjunto de diversos setores e que também pode ser considerado distribuído, pois cada um destes setores está localizado em diferentes lugares, apresenta-se a dificuldade em administrar este grande volume de dados. Uma solução promissora para este problema é trabalhar com bancos de dados federativos, de tal maneira que cada banco de dados preserve a sua autonomia, ao mesmo tempo em que é possível preservar também a flexibilidade, ao se evitar um sistema monolítico, centralizado. Para se trabalhar com bancos de dados distribuídos de modo colaborativo se propõe um modelo de integração baseado em uma arquitetura orientada a serviços, na qual cada atividade do desenvolvimento do produto requisita ou oferece serviços Web das outras atividades. Esta solução permite que um ambiente de aparente complexidade possa ser reduzido a problemas simples e consiga trabalhar com grande volume de dados. Esta tese apresenta uma solução colaborativa e distribuída para se trabalhar com os dados de um sistema produtivo ao longo do ciclo de vida do produto através de um conjunto de serviços Web, estruturados em uma arquitetura orientada a serviços.

Palavras-chave: Sistema Produtivo; Desenvolvimento do Produto; Gerenciamento Federativo; Arquitetura Orientada a Serviços.

DE ASSIS MOURA, Antonio Álvaro. *Integration of Product, Processes and Resources Information on a Productive System using Web Services*. 2012. 182f. Thesis (Doutorado em Engenharia de Produção) - Faculdade de Engenharia, Arquitetura e Urbanismo, Universidade Metodista de Piracicaba - UNIMEP, Santa Bárbara d'Oeste, SP.

Abstract

The amount of information involved in a productive system during the product development cycle grows steadily, covering more and more activities. Each of the steps involved in the productive system uses specific software tools that generate and manage their data independently. Since the productive system is an arrangement that can be done in a collaborative manner through the combined efforts of various sectors and which can also be considered distributed, since each of these sectors are located in different places, There is a the difficulty in managing this huge volume of data. A promising solution to this problem is to work with federative databases, so that each database preserves its autonomy, while it is also possible to preserve the flexibility to avoid a monolithic, centered system. A service-oriented architecture is proposed to work with distributed databases in a collaborative way, where each activity of the productive system requests or offers other services. This solution allows that an environment of apparent complexity can be reduced to simple problems and can work with large volumes of data. This thesis presents a distributed and collaborative solution for working with data from a productive system throughout the product life cycle using a set of Web services, structured in a service-oriented architecture.

Keywords: Productive System; Product Development; Service Oriented Architecture; Federative Management.

Sumário

Lista de Figuras.....	I
Lista de Siglas.....	III
Lista de Institutos e Laboratórios.....	V
1 Introdução.....	1
1.1 Considerações Gerais.....	1
1.2 Objetivos desta Tese.....	6
1.3 Delimitação desta pesquisa.....	6
1.4 Estrutura desta tese.....	7
2 Sistema Produtivo.....	8
2.1 Considerações Iniciais.....	8
2.2 Gerenciamento de Sistemas de manufatura.....	9
2.3 Representação de Sistemas de manufatura.....	12
2.3.1 Visão de Produto.....	15
2.3.2 Visão de Processos.....	18
2.3.3 Visão de Recursos.....	21
2.4 Heterogeneidade dos Dados de Produção.....	23
2.5 Teoria da Complexidade.....	24
2.6 Banco de Dados.....	30
2.6.1 Banco de Dados Relacionais.....	31
2.6.2 Banco de Dados Orientado a Objetos.....	32
2.6.3 Banco de Dados Pós-Relacionais.....	33
2.7 Sistemas de Gerenciamento de Banco de Dados.....	35
2.7.1 Sistemas Centralizados.....	35
2.7.2 Sistemas Heterogêneos.....	36
2.7.3 Sistemas Distribuídos.....	37
2.7.4 Sistemas Federativos.....	38
2.7.5 Integração de Dados.....	39
2.8 Considerações Finais do Capítulo.....	40
3 Tecnologias de Informação utilizadas durante o ciclo de vida do produto.....	42
3.1 Aplicativos <i>Add-on</i> para sistemas CAx.....	42

3.2	Projeto Auxiliado por Computador	42
3.3	Manufatura Auxiliada por Computador	47
3.4	SOA - Arquitetura Orientada a Serviços	49
3.5	Serviços Web	53
3.6	WSDL	56
3.7	Considerações Finais do Capítulo	57
4	Contribuição do Trabalho	58
4.1	Objetivos	58
4.2	Originalidade	59
4.3	Método de Pesquisa	59
4.4	Contextualização.....	60
5	Desenvolvimento do modelo	65
6	Desenvolvimento do Modelo	69
6.1	Aplicativos Internos a Sistemas CAx	72
6.1.1	Aplicativo para sistema CAD.....	74
6.1.2	Aplicativo para sistema CAM - Operações	78
6.1.3	Aplicativo para Sistema CAM - Ferramentas.....	81
6.2	Serviços Web	84
6.2.1	Análise e Recuperação de Dados do Produto	89
6.2.2	Análise e Recuperação de Dados de Processo	92
6.2.3	Análise e Recuperação de Dados de Recursos	93
6.3	Aplicativo para Integração de Dados	94
6.4	Serviços Independentes	97
6.4.1	Estimativa do tempo de manufatura	97
6.4.2	Determinação de pontos para inspeção de superfícies esculpidas.....	104
6.5	Verificação, Validação e Teste dos aplicativos desenvolvidos.	113
7	Conclusões	123
7.1	Sugestões de Trabalhos Futuros.....	124
	Referências Bibliográficas	126
	Apêndice A – Programa Fonte.....	138
	Apêndice B – Projeto <i>FEDMAN</i>	139
	Apêndice C – Modelos para testes	143

Lista de Figuras

Figura 2.1 Etapas do Desenvolvimento e do Ciclo de Vida do produto, adaptado de (ANDERL, 2006).	9
Figura 2.2 Três domínios de informações, adaptado de (NIELSEN, 2003).	13
Figura 2.3 Modelos de níveis hierárquicos para sistemas de manufatura. NIST à esquerda e ISO à direita, adaptado de (BAUER et al., 1991).	14
Figura 2.4 Etapas do ciclo de vida do produto, adaptado de (EIGNER e STELZER, 2009).	17
Figura 2.5 Hierarquização de Processos (SORDI, 2005).	18
Figura 2.6 Ciclo de vida do BPM em comparação com o Gerenciamento do Fluxo, adaptado de (VAN DER AALST et al., 2003).	20
Figura 2.7 Arquitetura para Recurso de Manufatura, adaptado de (CHENGYING, 2003).	22
Figura 2.8 Representação da evolução da gestão (BORGATTI NETO, 2007)	25
Figura 2.9 Modelo de complexidade, adaptado de (URBANIC e ELMARAGHY, 2006)	27
Figura 2.10 Evolução dos Bancos de Dados, adaptado de (KHOSHAFIAN, 1994).	31
Figura 2.11 Exemplo de Banco de Dados Relacional.	32
Figura 2.12 Sistema de Banco de Dados Distribuídos (FILETO e MEDEIROS, 2003).	38
Figura 2.13 Sistemas de Bancos de dados federativos (FILETO e MEDEIROS, 2003).	39
Figura 2.14 Comparação entre sistemas de integração.(ABELN, 1997; MONTAU, 1995; PEDERSON, 2005)	40
Figura 3.1 Ciclo de Projeto Clássico, adaptado de (SHIGLEY, 1977).	44
Figura 3.2 Comparação entre sistema clássico e computacional, adaptado de (GROOVER e ZIMERS, 1983).	45
Figura 3.3 Modelo em V para o desenvolvimento multidisciplinar do produto, adaptado de (ANDERL et al., 2010).....	46
Figura 3.4 Elementos de SOA, adaptado de (KRAFZIG et al., 2005).	51
Figura 3.5 Componentes de um serviço, adaptado de (KRAFZIG et al., 2005).....	52
Figura 3.6 Processo geral de funcionamento de serviço Web, adaptado de (BOOTH, 2004).....	55
Figura 4.1 Curva de comprometimento do custo do produto (ROZENFELD et al., 2006).....	61
Figura 4.2 Relação entre custo provocado e custo do componente, adaptado de (SMITH e REINERTSEN, 1998).....	62
Figura 5.1 Cenário proposto.....	66
Figura 5.2 Processo simplificado de desenvolvimento do produto.....	67
Figura 6.1 Esquema de funcionamento dos aplicativos e serviços desenvolvidos.....	71
Figura 6.2 Caixa de diálogo do aplicativo.	73
Figura 6.3 Árvore de construção.	75
Figura 6.4 Conjunto montado e árvore de construção (subconjunto do eixo com eixo e êmbolo).....	75
Figura 6.5 Estrutograma de leitura dos dados dos componentes.....	76
Figura 6.6 Excerto de arquivo XML com dados dos produto.	77
Figura 6.7 Caixa de diálogo com informação do envio do arquivo XML.	77
Figura 6.8 Sistema CAM com navegador de operações.....	78
Figura 6.9. Superfícies a serem usinadas na peça exemplo.	79
Figura 6.10 Dados da operação de faceamento e a ferramenta associada.	80

Figura 6.11 Estrutograma de leitura dos dados das operações de manufatura	80
Figura 6.12 Excerto de arquivo XML com dados das operações de manufatura	81
Figura 6.13. Caixa de diálogo do sistema CAM com dados da ferramenta.....	82
Figura 6.14 Estrutograma do aplicativo de leitura de ferramentas.....	83
Figura 6.15 Excerto de arquivo XML com dados de ferramentas	84
Figura 6.16 Estrutura de pastas de um serviço Web.	85
Figura 6.17 Apresentação de um serviço em um navegador.....	86
Figura 6.18 Trecho do arquivo WSDL.....	86
Figura 6.19 Informações sobre acesso e requisição do serviço.	87
Figura 6.20 Exemplo de uma resposta a um serviço Web.	88
Figura 6.21 Excerto da estrutura de classe Object	88
Figura 6.22 Sequência para obter o material de um componente.....	91
Figura 6.23. Sequência para obter o tipo de uma operação.	93
Figura 6.24 Telas com a sequência de obtenção de tipo e subtipo de uma ferramenta.	94
Figura 6.25 Tela do aplicativo com dados de produto e recursos.	95
Figura 6.26 Tela do aplicativo com dados de processos e recursos.	96
Figura 6.27 Tela do aplicativo com dados de produto e processo e também as características do componente escolhido.	96
Figura 6.28 Quatro comportamentos para o avanço (HEO et al., 2006).	98
Figura 6.29 Caixa de diálogo com a opção de tempo de manufatura	101
Figura 6.30 Estrutograma para estimativa do tempo de manufatura.	102
Figura 6.31 Estrutograma para o cálculo do tempo de manufatura para um bloco.	104
Figura 6.32 Fluxo de Informação na Integração CAD/CAI com serviços Serviços Web	106
Figura 6.33 Superfície esculpida construída com curvas B-Splines	107
Figura 6.34 Diagrama em Express-G com modelo de uma curva B-Spline. (ISO/IS 10303-238, 2006).....	108
Figura 6.35 Excerto de um arquivo XML com dados de uma curva B-Spline.....	109
Figura 6.36 Estrutograma para cálculo da função de suavização de primeira ordem.....	110
Figura 6.37 Estrutograma para cálculo das funções de suavização de segunda ordem e superiores.....	111
Figura 6.38 Estrutograma para cálculo do ponto a ser medido	111
Figura 6.39 Excerto de um arquivo para medição.....	113
Figura 6.40 Defeito, Erro e Falha (ROCHA e MALDONADO, 2001).....	114
Figura 6.41Conjunto montado de uma garra.....	116
Figura 6.42. Caixa de diálogo com os atributos do componente eixo, existente no conjunto montado.	116
Figura 6.43 Excerto de arquivo XML com dados do produto.....	117
Figura 6.44 Tela do aplicativo com os dados de produto.	118
Figura 6.45 Base do cilindro pneumático com as operações de manufatura.	119
Figura 6.46 Informações da operação de contorno.	119
Figura 6.47. Informações da ferramenta utilizada na operação de contorno.	120
Figura 6.48 Excerto de arquivo XML com dados de manufatura	121
Figura 6.49 Tela do aplicativo com as operações e os dados da ferramenta.	122
Figura 7.1 Níveis em sistema de manufatura e exemplos de sistemas de TI associados.	141

Lista de Siglas

ABNT	Associação Brasileira de Normas Técnicas
ASMX	<i>Active Server Methods</i>
BOM	<i>Bill of Materials</i>
BPM	<i>Business Process Modeling</i>
CAD	<i>Computer Aided Design</i>
CAE	<i>Computer Aided Engineering</i>
CAI	<i>Computer Aided Inspection</i>
CAM	<i>Computer Aided Manufacturing</i>
CAx	<i>Computer Aided ...</i>
CLData	<i>Cutter Location Data</i>
CNC	Controle Numérico Computadorizado
CORBA	<i>Common Object Request Broker Architecture</i>
CRM	<i>Customer Relationship Management</i>
ERP	<i>Enterprise Resource Planning</i>
FTP	<i>File Transfer Protocol</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDEFO	<i>Integrated Definition for Function Modeling</i>
IDL	<i>Interface Description Language</i>
IIS	<i>Internet Information Services</i>
KBE	<i>Knowledge Based Engineering</i>
MANPro	<i>Mobile Agent-based Negotiation Process</i>
MMC	Máquina de Medir por Coordenadas

MRP	<i>Materials Requirements Planning</i>
MRP II	<i>Manufacturing Resources Planning</i>
NC	<i>Numeric Control</i>
NIST	<i>National Institute of Standards Technology</i>
NURBS	<i>Non-Uniform Rational B-Spline</i>
PDM	<i>Product Data Management</i>
PLM	<i>Product Lifecycle Management</i>
SADT	<i>Structured Analysis Design Technique</i>
SCM	<i>Supply Chain Management</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Single Object Access Protocol</i>
SQL	<i>Structure Query Language</i>
STEP	<i>Standard for the exchange of product model data</i>
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>
WSDL	<i>Web Service Description Language</i>
XML	<i>eXtensible Markup Language</i>

Lista de Institutos e Laboratórios

BRAGECRIM	<i>Brazilian-German Collaborative Research Initiative on Manufacturing Technology</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DFG	<i>Deutschen Forschungsgemeinschaft</i>
DiK	<i>Institut für Datenverarbeitung in der Konstruktion</i>
FINEP	Financiadora de Estudos e Projetos
ISO	<i>International Organization for Standardization</i>
PTW	<i>Institut für Produktionsmanagement, Technologie und Werkzeugmaschinen</i>
SCPM	Laboratório de Sistemas Computacionais para Projeto e Manufatura
W3C	<i>World Wide Web Consortium</i>

1 Introdução

Neste capítulo inicial são apresentados os domínios possíveis para um sistema produtivo centrado na manufatura considerando-o como um sistema colaborativo e distribuído. São destacadas as questões relativas às dificuldades na integração dos dados pertinentes a cada uma das partes que compõem o sistema e algumas soluções propostas para superar estas dificuldades. Também estão contidos neste capítulo, os objetivos, a delimitação e a estrutura desta tese.

1.1 Considerações Gerais

A crescente competição entre as empresas, que extrapolou fronteiras e se dá de forma globalizada, tornou dois requisitos essenciais para a sobrevivência, quais sejam: a contínua inovação de produtos e a redução do tempo de lançamento ao mercado. Para atender a esses requisitos foram desenvolvidas diversas ferramentas de informática para a gestão da manufatura que, se por um lado melhoram a eficiência dos processos, por outro geram uma grande quantidade de dados heterogêneos, dificultando o gerenciamento e criando um obstáculo aos requisitos citados.

Para empresas de manufatura esta competição se dá para além da manufatura em si, senão por todo o sistema produtivo, pois inclui todo o ciclo de vida do produto. Nesta tese será considerado como campo de atuação o sistema produtivo tendo de um lado o ciclo de vida do produto e de outro os processos produtivos.

Um sistema produtivo pode ser entendido como o local onde as forças de produção se combinam com os processos de produção, sendo constituído de força de trabalho; os meios de produção; os sistemas nos quais a produção é organizada, a estrutura de controle da atividade produtiva e a estrutura política e social no qual o processo produtivo está inserido (WILKINSON, 1983). Parte do sistema produtivo é o sistema de manufatura entendido como o conjunto de meios e processos utilizados para a transformação de matéria prima em um produto acabado. Em um entendimento ainda mais amplo, um sistema de manufatura pode ser considerado como a organização que se insere dentro da indústria de manufatura para a criação do produto (SLACK, 2002).

Por ser um conjunto, um sistema produtivo pode ser dividido em subsistemas com aptidões específicas; historicamente cada um desses subsistemas teve o desenvolvimento de ferramentas de informática para apoio às suas atividades de forma isolada, com cada área do conhecimento buscando o melhor desempenho e a melhor solução para sua aplicação. Neste contexto surgiram as ferramentas de apoio à engenharia conhecidas pelo acrônimo CAx, na qual CA indica apoiado ou assistido por computador (*Computer Aided*) e “x” pode indicar qualquer área da engenharia como, por exemplo: CAD - Projeto (*Design*); CAM - Manufatura (*Manufacturing*); ou CAE - Engenharia (*Engineering*), entre outras.

O sistema produtivo pode ser dividido em processos específicos para cada uma das atividades componentes. Ferramentas de informática para apoiar processos foram desenvolvidas ao mesmo tempo e também isoladamente, sendo que algumas dessas foram desenvolvidas para apoiar exclusivamente processos de fabricação enquanto outras, mais genéricas, foram desenvolvidas para apoiar os processos empresariais, agregando inclusive o gerenciamento dos recursos para produção (URBANIC; ELMARAGHY, 2006).

A consequência desse desenvolvimento isolado foi a criação de uma barreira entre as aplicações o que estabeleceu uma dificuldade na integração das informações. As soluções apresentadas caminharam em duas direções opostas: i) efetivar esta integração por meio de vínculos indiretos, mantendo cada parte isolada e independente de um modelo central; ii) de buscar a integração total, utilizando vínculos diretos de tal forma que todas as aplicações estejam internalizadas em um modelo central (ANDERL *et al.*, 2009).

Se a solução de utilizar vínculos indiretos tem a facilidade de se trabalhar com aplicações distribuídas e a solução de integração total tem na integração dos modelos parciais uma vantagem, ambas as soluções apresentam problemas na flexibilização do modelo, que se torna um obstáculo à inovação.

Ao explorar a literatura percebe-se que qualquer que seja a solução apresentada para a integração de um sistema produtivo, quatro características sempre estão presentes: a heterogeneidade; a autonomia; a distribuição e a complexidade.

A heterogeneidade é caracterizada pela quantidade e severidade das discrepâncias entre os dados dos componentes do sistema; essas discrepâncias podem existir em qualquer nível de abstração. Como as soluções isoladas podem utilizar diferentes sistemas operacionais, diferentes sistemas de banco de dados e têm diversas visões sobre os domínios de conhecimento é natural que haja discrepâncias entre os componentes do sistema (EIGNER, 2010).

Da mesma forma, o desenvolvimento isolado das ferramentas de informática se deu com total autonomia, visto que cada componente respondia por si e pôde escolher com liberdade qual a maneira de criar e gerenciar seus dados.

A característica de distribuição é percebida de modo imediato como decorrência da especialização das partes contidas no ciclo de vida do produto. Dentro da competição globalizada, setores especializados em determinada área do conhecimento podem estar fisicamente distantes, seja em cidades, estados ou mesmo países. No entanto, considerando o aspecto virtual, esta distribuição pode não ser somente física, senão de diferentes localizações virtuais para as informações pertinentes a um determinado campo de atuação. Desta forma, podem ser considerados distribuídos setores de uma mesma empresa que armazenem as informações em bancos de dados isolados (LITWIN *et al.*, 1990).

Essencialmente por causa da grande quantidade de dados envolvidos em um sistema produtivo, este é apresentado na literatura como complexo e as soluções baseadas na teoria da complexidade são mostradas para o gerenciamento do ciclo de vida.

Dois grandes correntes exploram o tema da complexidade: A corrente que prega a necessidade de um pensamento mais complexo, tendo uma abordagem mais filosófica e subjetiva, portanto relacionada às paixões e complexidades das relações humanas; e a corrente que busca uma formalização científica estando embasada nas ciências

naturais, na modelagem computacional e na matemática. Mesmo que entre estas duas grandes correntes apareçam ideias concorrentes ou mesmo conflitantes, são sustentadas por princípios comuns (BORGATTI NETO, 2007).

A falha na integração das informações em um sistema produtivo pode ocasionar a demora na obtenção de uma informação, o que leva à dilação no tempo esperado para a atividade em questão; ou então a obtenção de uma informação equivocada ou desatualizada, o que pode levar a erro no trabalho desenvolvido e o consequente retrabalho. De toda maneira a falta de integração piora o desempenho das atividades envolvidas.

Ferramentas de tecnologia de informação desenvolvidas isoladamente também têm ciclos de vida diferentes. Pode acontecer que ferramentas descontinuadas, sem manutenção, conhecidos por sistemas legados, necessitem ser agregadas a outras ferramentas. As soluções de integração devem considerar a possibilidade de utilizar dados de sistemas legados (LUBELL *et al.*, 2004).

A integração dos dados em um sistema produtivo também permite a solução antecipada de problemas. No desenvolvimento do produto, durante suas fases iniciais como concepção, especificação ou planejamento, é que são possíveis e menos onerosas as modificações no produto. Por outro lado neste mesmo momento é que se tem pouco conhecimento sobre o produto. Nas etapas finais, quando os custos de modificações são elevados e a possibilidade de redução de custos no produto é menor, é que se tem mais conhecimento do produto. Para influenciar o custo do produto na fonte geradora desses custos é preciso antecipar a solução de problemas. Uma forma de se fazer isso é aumentar a inserção de tecnologia de informação nas fases iniciais do desenvolvimento do produto, dito de outra forma, incluir as informações do sistema produtivo ao desenvolvimento do produto desde o seu início (EIGNER; STELZER, 2009).

Diante disso, o desafio que se apresenta é a integração das informações do sistema produtivo como um todo, de tal maneira que não se percam as vantagens da autonomia dos sistemas isolados e também se preserve a capacidade de flexibilização para

atender os requisitos de inovação, além de permitir a utilização de sistemas legados e prover a capacidade de solução antecipada de problemas.

Uma solução promissora é a divisão das informações do sistema produtivo em três domínios ortogonais entre si, um para o produto, outro para os recursos disponíveis para a manufatura e um terceiro para o domínio dos processos envolvidos. Esses domínios podem se comunicar dois a dois, preservando a autonomia de cada um dos subsistemas e facilitando a comunicação (NIELSEN, 2003). Os componentes dos domínios armazenam seus dados em bancos de dados federativos e a comunicação entre os componentes é feita através de serviços organizados em camadas.

Nos bancos de dados federativos, cada aplicação utiliza um esquema para a obtenção dos dados e os esquemas para importação e exportação são gerenciados externamente ao sistema. O sistema federativo permite um grau de autonomia muito mais elevado para os sistemas componentes do que outros sistemas de bancos de dados como os sistemas distribuídos e centralizados (ANDERL *et al.*, 2009).

A comunicação entre componentes feita mediante serviços organizados em camadas está sendo cada vez mais utilizada, isso porque permite a implementação das mais modernas tecnologias de informática como a computação em nuvem e a arquitetura orientada a serviços que lida com processos distribuídos em sistemas heterogêneos fazendo a interação entre os provedores de serviços e os consumidores de serviços. Essa arquitetura possibilita que os agentes localizem e utilizem as capacidades distribuídas de que precisem, mesmo que cada componente esteja sob controle de diferentes proprietários, o que mantém a autonomia de cada componente (BOOTH, 2004).

Essa solução está inserida como uma proposta no projeto *FEDMAN* desenvolvido em parceria pela *Technische Universität Darmstadt*, através do *DiK - Institut für Datenverarbeitung in der Konstruktion* e pela Universidade Metodista de Piracicaba, através do Laboratório de Sistemas Computacionais para Projeto e Manufatura, conforme descrito no apêndice C.

1.2 Objetivos desta Tese

Diante dessas considerações, dois objetivos estão presentes nesta tese: (1) propor um modelo para integração das informações de um sistema produtivo baseado em um processo de análise e estabelecimento de relações causais lineares, dentro do paradigma da simplicidade e (2) sobre este modelo, propor uma plataforma orientada a serviços Web para a integração digital dessas informações. Em especial para a integração do desenvolvimento do produto aos processos produtivos.

Com o desenvolvimento deste trabalho espera-se demonstrar que o sistema produtivo, posto de aparência complexa e inserido em um ambiente colaborativo e distribuído, pode ser reduzido a um conjunto de sistemas simples, de tal forma que uma plataforma orientada a serviços, lançando mão das modernas tecnologias de informação disponíveis, seja capaz de integrar um grande volume de dados. Como consequência geral espera-se que esta integração permita a solução antecipada de problemas e a preservação de dados de sistemas legados, levando a uma melhoria geral do desempenho do sistema.

1.3 Delimitação desta pesquisa

Esta pesquisa tem seu foco na proposta de uma solução simplificadora para integração digital de dados heterogêneos, dentro de um ambiente colaborativo e distribuído, de tal forma a prover a solução antecipada de problemas e a preservação de dados de sistemas legados. Utiliza os conceitos e ferramentas de tecnologia de informação existentes. Trata-se, portanto da propositura de um modelo e da apresentação de uma ferramenta que o viabiliza.

Não se espera que as ferramentas apresentadas se estendam além do nível necessário para demonstrar a viabilidade do modelo dentro do cenário proposto. Entende-se que o efetivo trabalho com grande volume de dados deve-se dar sobre o conceito apresentado dentro de uma aplicação comercial.

As ferramentas de informática desenvolvidas são aplicadas para um cenário reduzido, porém considerando o modelo como um todo, em todas as suas dimensões. Para a

solução antecipada de problemas são apresentados dois serviços específicos, sendo um para o domínio de produto (disponibilização de pontos para inspeção de superfícies esculpidas em máquinas de medir por coordenadas) e para o domínio de processos (estimativa de tempo de usinagem a partir do arquivo CLData).

1.4 Estrutura desta tese

Esta tese está estruturada em sete capítulos considerando esta introdução.

O capítulo 2 apresenta, como revisão de literatura, o conceito geral de sistemas produtivos e os modelos utilizados para representação do sistema, com destaque para o modelo que considera o sistema produtivo sob o ponto de vista de três dimensões, quais sejam, a dimensão do produto, a dimensão dos processos e a dimensão dos recursos. Adicionalmente discute aspectos da teoria da complexidade e alguns trabalhos que propõem modelos complexos para sistemas de manufatura.

No terceiro capítulo são mostradas as tecnologias de informação, inclusive sistemas de bancos de dados, que podem ser utilizadas na integração digital, também dentro deste capítulo são apresentados sistemas apoiados por computador para cada um dos domínios de um sistema de manufatura.

No capítulo 4 são mostrados os objetivos e a originalidade do trabalho dentro do escopo dessa tese, bem como o método de pesquisa e a definição de contexto.

O capítulo 5 descreve a estrutura de aplicação da solução proposta, enquanto que o capítulo 6 mostra como a solução foi implementada, descrevendo os aplicativos de coleta e disponibilização de dados de cada um dos domínios considerados, dos serviços de acesso a essas informações e do aplicativo que integra as informações.

O capítulo 7 apresenta as conclusões desta tese, suas limitações e possibilidades de futuros estudos e pesquisas na área.

Ao final, estão as referências bibliográficas e os apêndices com o programa fonte, a descrição do projeto *FEDMAN* e o resultado dos testes efetuados.

2 Sistema Produtivo

No capítulo introdutório foram exibidas as questões centrais sobre a quantidade de informações envolvidas em um sistema produtivo, suas características de distribuição e colaboração, bem como as principais soluções apresentadas para gerenciar as informações e os problemas causados pela falta de integração das informações.

O presente capítulo detalha o conceito de sistema produtivo e a divisão em três domínios, avança para as questões relativas aos bancos de dados e seu gerenciamento e também discute a teoria da complexidade como uma solução para o gerenciamento dos dados.

2.1 Considerações Iniciais

Wilkinson (1983) define um sistema produtivo como o local onde as forças de produção se combinam com os processos de produção, sendo constituído de força de trabalho; os meios de produção; os sistemas nos quais a produção é organizada, a estrutura de controle da atividade produtiva e a estrutura política e social no qual o processo produtivo está inserido. No mesmo sentido Birecree *et al.* (1997) definem sistema produtivo como os processos de produção de um produto, ou grupo de produtos, e os vínculos entre os diferentes processos; destaca-se que nesse entendimento nenhum processo pode operar de forma isolada e que se qualquer dos processos não funcionar adequadamente todo o sistema produtivo é prejudicado. Portanto, em um sistema produtivo a cooperação entre os seus agentes é uma necessidade técnica que determina os custos de produção e a qualidade do produto enquanto saída do sistema produtivo e em última instância determina a competitividade do sistema. Com um olhar mais específico para a manufatura Anderl (2006), estuda a intersecção dos processos do ciclo de vida com os processos do desenvolvimento do produto, como apresentado na Figura 2.1. Um sistema de manufatura é o local onde os dois eixos se encontram, pertencendo ao mesmo tempo ao ciclo de vida do produto e ao desenvolvimento do produto. Os recursos utilizados na manufatura podem ser os recursos de transformação (como instalações, máquinas, equipamentos, mão de obra) e os recursos a transformar (materiais, dados).

Ao conjunto de processos (tanto do ciclo de vida de produto, quando do desenvolvimento do produto), recursos (de transformação e a transformar) e do produto em si, define-se um sistema produtivo, ou mais especificamente, para o caso de um produto obtido por manufatura, um sistema de manufatura.

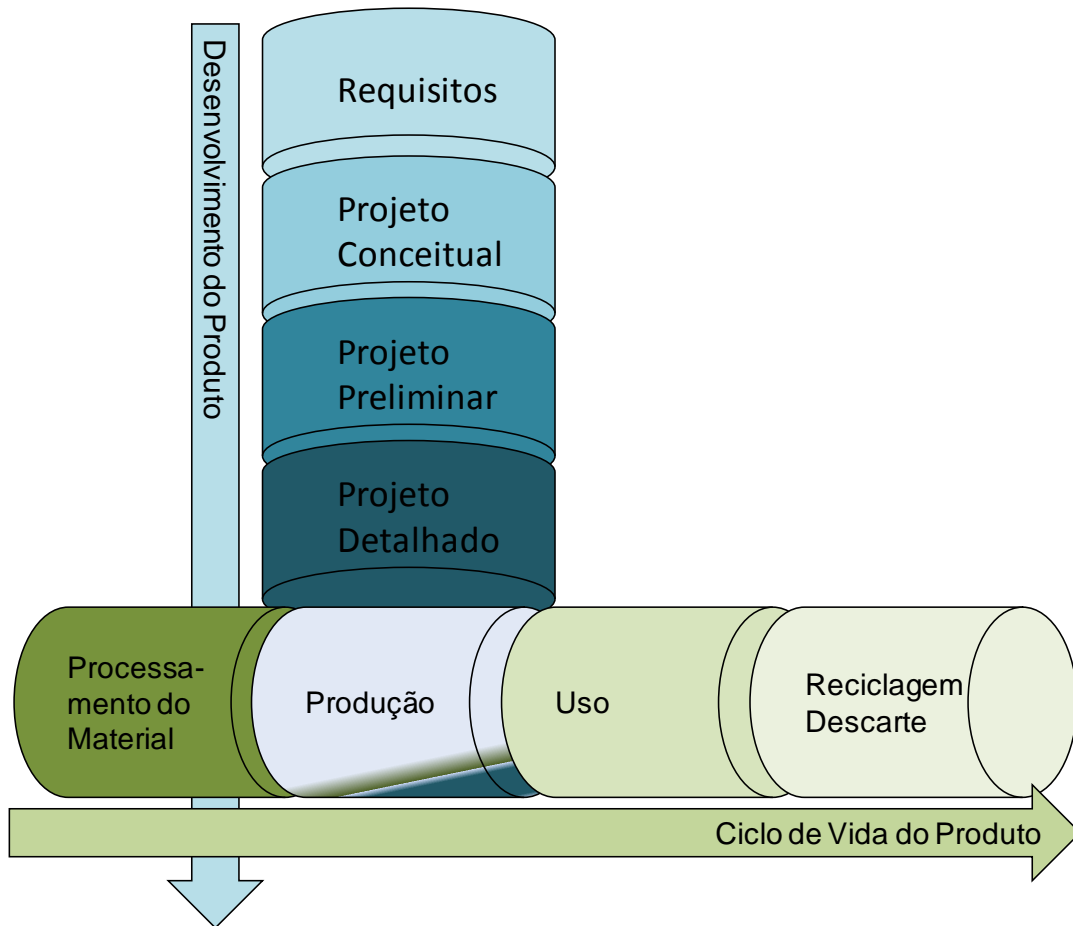


Figura 2.1 Etapas do Desenvolvimento e do Ciclo de Vida do produto, adaptado de (ANDERL, 2006).

Neste trabalho serão consideradas as informações necessárias ao sistema produtivo entendido como a agregação do ciclo de vida do produto; do desenvolvimento do produto e dos recursos associados.

2.2 Gerenciamento de Sistemas de manufatura

Uma definição muito simples de manufatura é a do processo de transformação de matéria prima. Nas empresas que produzem produtos manufaturados a função de transformação de matéria prima tem papel central para o estabelecimento de uma

vantagem competitiva e por esta razão deve ser pensado estrategicamente (SLACK, 2002).

Segundo este mesmo autor, o sistema de manufatura se sustenta sobre as atividades de marketing, desenvolvimento do produto e manufatura em si. Como a manufatura é a responsável direta pela transformação da matéria prima em produto acabado, em um relacionamento direto com o processo físico da produção, é esta atividade que de modo mais perceptível adiciona valor ao produto, daí a grande importância de um gerenciamento adequado desta atividade (ANTUNES JR., 1998).

Dentro da área de conhecimento da engenharia de produção, algumas pesquisas em nível nacional propõem soluções para o gerenciamento de um sistema de manufatura. Especialmente no que concerne ao desenvolvimento do produto destacam-se: o trabalho de Araújo (2003) com a propositura de uma ontologia de linguagem para as fases de concepção e especialização no desenvolvimento do produto; considerando todo o ciclo de manufatura. Boer (2008) propõe um roteiro de gestão da manufatura com o foco no valor do produto. Para uma metodologia estruturada para a aplicação dos conceitos de manufatura enxuta tem-se o trabalho de Machado (2006) e também o trabalho de Zancul (2009) propondo a gestão do ciclo de vida do produto através de modelos de referência.

Com um foco maior em tecnologia da informação Feliciano (2009) propõe um sistema de apoio à tomada de decisão para o sistema de manufatura, baseado em inteligência de negócio; também levando em consideração inteligência de negócio Franco (2003) estuda a base tecnológica para um sistema inteligente de manufatura dentro do paradigma holônico, que considera a junção dos hólons (*holos + on*) como partes componentes do todo. Tálamo (2008) constrói uma rede de colaboração entre empresas, que mesmo não sendo em empresas de manufatura, demonstra os requisitos necessários a um trabalho colaborativo.

Com um olhar específico nas não conformidades em ambiente de manufatura Mikos (2008) apresenta um modelo bastante amplo sobre a utilização de agentes no apoio a solução de problemas, incluindo ontologias e raciocínio baseado em casos; também

usando agentes Nakai (2007) propõe uma coreografia¹ baseada em serviços Web para a coordenação de atividades em cadeias produtivas, que é gerenciada através de conversação entre múltiplos serviços. Borsato (2003), constrói uma plataforma baseada em agentes para apoiar o desenvolvimento rápido de produtos dentro dos preceitos de engenharia simultânea.

No âmbito internacional, diversas pesquisas aplicaram tecnologias de serviços Web para executar tarefas que permitam trabalhar em um ambiente distribuído e colaborativo. Entre as quais, Yuan *et al.* (2008) apresentam uma estrutura genérica para controlar o fluxo de trabalho em sistemas de manufatura controlados por computador, estrutura esta projetada para acomodar as mudanças frequentes no ambiente, tais como falhas em equipamentos, falta de ferramentas e as variações de processamento. Norrie e Maturana (1996) desenvolveram uma arquitetura de planejamento de tarefas distribuídas, incluindo a manufatura distribuída, de tal forma a estruturar ambientes de serviços Web heterogêneos dentro de grupos bem organizados, nos quais são realizadas as atividades de coordenação. Klein (1991) propõe a implementação de um modelo computacional como uma abordagem de apoio à resolução de conflitos no projeto cooperativo que evita muitas das limitações dos trabalhos anteriores nesta área. Blecker e Graf (2003) discutem uma aplicação coordenada para customização em massa com o auxílio de um sistema multi-agente, num ambiente de produção baseado na Web, para o planejamento e controle da produção. Ong e Sun (2003) propuseram uma arquitetura distribuída baseada na Web para o desenvolvimento de um sistema de monitoramento de tempo real de uma plataforma independente por meio de agentes móveis. Shin e Jung (2004) propuseram um mecanismo de negociação chamado de MANPro (*Mobile Agent-based Negotiation PROCESS* - Processo de Negociação Baseado em Agente Móvel), que aplica um sistema de agentes móveis no processo de troca de informação. Kulvatunyou *et al.* (2004) descrevem uma estrutura que facilita a colaboração de tarefas de engenharia, particularmente o planejamento e análise de processo para a atividade de manufatura globalizada.

¹ Conversação entre múltiplos serviços Web descrita por meio de máquinas de estado, diagramas de sequência, de atividades ou por linguagens específicas.

Dos trabalhos apresentados depreende-se que há uma necessidade de organizar o gerenciamento da manufatura e que várias soluções são possíveis, sendo que as mais recentes utilizam agentes como o mecanismo escolhido para lidar com dados heterogêneos (BLECKER e GRAF, 2003), (ONG e SUN, 2003) e (SHIN e JUNG, 2004). Nestes trabalhos é possível perceber a presença do problema de harmonizar colaboração e distribuição que é o que ocorre em um sistema de manufatura cuja atividade é feita com vários atores em diversos locais. Mesmo abordagens com inovação radical (paradigma holônico ou coreografia de agentes, e.g.) surgem para atender esta harmonização.

2.3 Representação de Sistemas de manufatura

A origem da palavra manufatura tem sua etimologia do termo latino *manus factus*, literalmente feito à mão². De modo geral a manufatura pode ser definida como um processo de transformação de matéria prima em produto final. Por sistema entende-se o conjunto de meios e processos para alcançar determinado fim. Portanto, um sistema de manufatura é o conjunto de meios e processos utilizados para a transformação de matéria prima em produto acabado. Dentro de um entendimento ainda mais amplo, um sistema de manufatura pode ser estendido para toda a organização que se insere dentro da indústria de manufatura.

Por ser um conjunto, um sistema de manufatura pode ser dividido em subsistemas com aptidões específicas e pode ser estendido para também compreender as fases precedentes como as que normalmente pertencem ao domínio do desenvolvimento do produto (e.g. Projeto Conceitual) e as que são posteriores ao produto final como o descarte ou reutilização (KURTZ, 1986).

Para o entendimento e maior conhecimento de um sistema real é possível a criação de modelos que permitam: explicar o comportamento do sistema; construir hipóteses sobre o comportamento observado; modificar o desempenho do sistema; auxiliar no projeto de novos sistemas e descrever comportamentos futuros ou modificações de resultados a partir de mudanças nas entradas do sistema (PIDD, 1996).

² Do latim medieval, “manu factus”, literalmente, feito à mão (MERRIAM-WEBSTER, 2008).

Para um modelo que represente um sistema de manufatura devem ser contemplados diferentes aspectos inerentes ao sistema e também ao produto durante todo o ciclo de vida, como isso agrega uma grande quantidade de informações o modelo pode dividido em subsistemas.

Uma proposta feita por Nielsen (2003) é apresentada na Figura 2.2. Onde o conjunto de informações do sistema produtivo é dividido em três grandes domínios ou visões permitindo uma decomposição dos elementos que compõem este sistema de modo que cada uma destas visões seja ortogonal a outra, eliminando assim as interdependências, de tal forma que as relações possam ser analisadas duas a duas. Este modelo apresenta um eixo a mais, o eixo dos recursos, do que o modelo proposto por Anderl (2006) e apresentado na Figura 2.1.

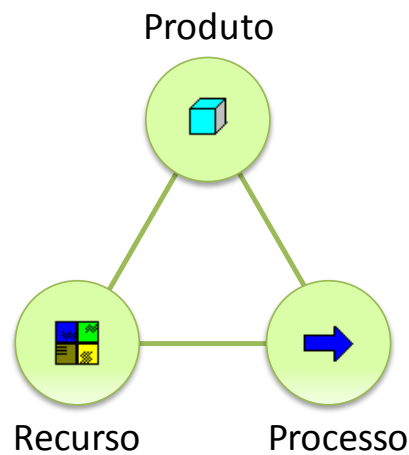


Figura 2.2 Três domínios de informações, adaptado de (NIELSEN, 2003).

Esta divisão em três domínios, separando o que é pertinente a produto, processo e recurso, pela eficiência de sua simplicidade aparece em diversos trabalhos científicos, dentre os quais se destacam: (KIM, G. Y. *et al.*, 2008); (CHRYSSOLOURIS *et al.*, 2009) e (BIN; LEE, I., 2003).

A análise das relações entre as visões, duas a duas, é mais simples, pois quando se analisam as relações entre recursos e produto se excluem as questões relativas ao processo, da mesma forma que ao se olhar os vínculos entre processo e produto, não se consideram os recursos. Com isso é possível que sejam incluídas restrições que

facilitam a construção do modelo central hipotético, neste sim consideradas todas as inter-relações entre os três domínios.

Outra possibilidade para a construção do modelo de um sistema produtivo é a divisão em níveis de planejamento, de tal maneira que o nível inferior esteja subordinado ao nível imediatamente superior. Tanto o *National Institute of Standards Technology* (NIST) quanto a *International Organization for Standardization* (ISO) têm normatizações a respeito de níveis hierárquicos encontrados em um sistema produtivo, como pode ser visto na Figura 2.3, na qual se pode perceber que os dois modelos apresentados são bastante similares, sendo que a ISO inclui um nível a mais, o que permite um maior detalhamento, em contrapartida um modelo com menos níveis pode ser de entendimento mais simples.

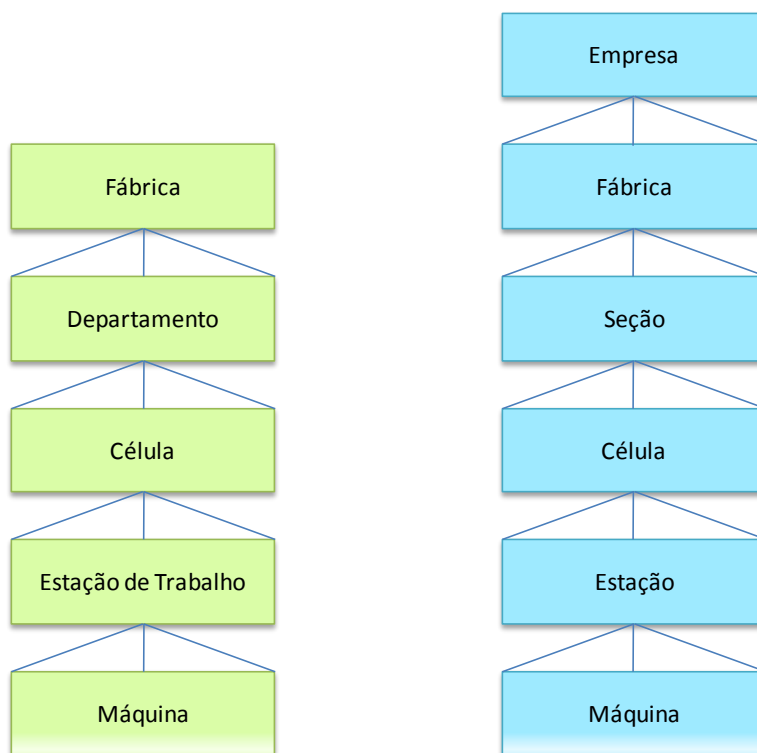


Figura 2.3 Modelos de níveis hierárquicos para sistemas de manufatura. NIST à esquerda e ISO à direita, adaptado de (BAUER et al., 1991).

Esta divisão em níveis hierárquicos não exclui a divisão feita por domínios, mas sim a complementa, já que para cada um dos domínios apresentados pode ser aplicada a divisão por níveis hierárquicos.

2.3.1 Visão de Produto

A visão de produto, considerado o modelo previamente mencionado, se encontra no eixo do ciclo de desenvolvimento do produto e isola as questões relativas a recursos e a processos.

As questões relativas ao produto podem ser reduzidas ao produto em si, definido por Kotler (2003) como “*qualquer coisa que é oferecida ao mercado para uso, aquisição ou consumo com a possibilidade de satisfazer um desejo ou necessidade*”, mas também pode ser ampliada para atender a uma esfera maior do que simplesmente o objeto físico como definido por McCarthy (1997), que inclui acessórios, instalações, instruções sobre uso, embalagem, o nome, além da relação existente com algumas necessidades psicológicas, e a certeza de que estarão disponíveis os serviços de assistência técnica indispensáveis para que as necessidades do consumidor mantenham-se satisfeitas após a compra.

A visão do domínio do produto em si pode, mesmo assim, apresentar vários modelos. Krause *et al.* (1993) distinguiram cinco diferentes tipos de modelos, que mesmo sendo incompletos, permitem uma visão mais aprofundada sobre o tema. Os tipos de modelos propostos são:

- Modelo do produto orientado a estrutura;
- Modelo do produto orientado a geometria;
- Modelo do produto baseado em *features*;
- Modelo do produto baseado no conhecimento; e
- Modelo do produto integrado.

Um modelo de produto baseado na sua estrutura pode ser subdividido em componentes ou funções, atendendo, por exemplo, a uma estrutura de montagem. Já modelos de produtos orientados a geometria são construídos com as bases tecnológicas disponíveis para a representação geométrica do produto. O modelo

baseado em *features* contém, além da informação geométrica, um conjunto de informações de alto nível, que pode definir um grupo de características ou conceitos semânticos, que permitem um nível de automação mais elevado que os sistemas anteriores (ACHTEN e LEEUWEN, 1999).

Modelos baseados no conhecimento são aqueles capazes de agregar ao modelo algum conhecimento sobre o produto de modo a permitir que o projeto tenha vínculos com outros domínios. Esses vínculos são utilizados para orientar e guiar o projetista durante o desenvolvimento do produto.

Modelos como os baseados na norma ISO STEP 10303-214 são capazes de combinar características de geometria, estrutura e *features*, sendo, portanto considerados modelos do produto integrados (LUBELL *et al.*, 2004).

Outros modelos também podem ser considerados, tais como modelos baseados em materiais, ou modelos cinemáticos, com campos de utilização bem específicos. (NIELSEN, 2003).

Para o gerenciamento do ciclo de vida do produto foram desenvolvidos sistemas conhecidos pela sigla PLM (*Product Lifecycle Management*), projetados para gerenciar tanto as questões relacionadas com a engenharia quanto com o desenvolvimento de produto. Segundo Brandão e Wynn (2008), PLM é "*uma abordagem de estratégia de negócio que aplica um conjunto de soluções de negócio no suporte da criação colaborativa, gerenciamento, disseminação e uso da informação de definição de produto por toda a empresa, da concepção ao fim do ciclo de vida - integrando pessoas, processos, sistemas de negocio e informação*". O sistema PLM é sobretudo uma extensão do sistema PDM, advindo da sigla em inglês para *Product Data Management*, que por sua vez atua predominantemente na área da Engenharia de Produto e engloba uma parte da área de Planejamento do Produto e Engenharia de Processo. Já o PLM atua desde o Planejamento do Produto até a sua Manutenção, Reparação e Revisão, envolvendo fornecedores e clientes (EIGNER e STELZER, 2009). É possível, deste modo, afirmar que PDM continua sendo uma base dos sistemas PLM (BRANDAO e WYNN, 2008).

Da mesma forma Eigner e Stelzer (2009) consideram o gerenciamento dos dados do produto como parte integrante do gerenciamento do ciclo de vida do produto, que se inicia com uma ideia, um pedido do cliente, ou uma pesquisa de mercado, evolui para o planejamento, e o desenvolvimento do produto, sendo estas duas etapas contidas no gerenciamento dos dados do produto. As etapas seguintes, de planejamento dos processos, produção; uso e reuso, estão contidas, junto com as demais, no todo que é o gerenciamento do ciclo de vida do produto, como pode ser observado na Figura 2.4.

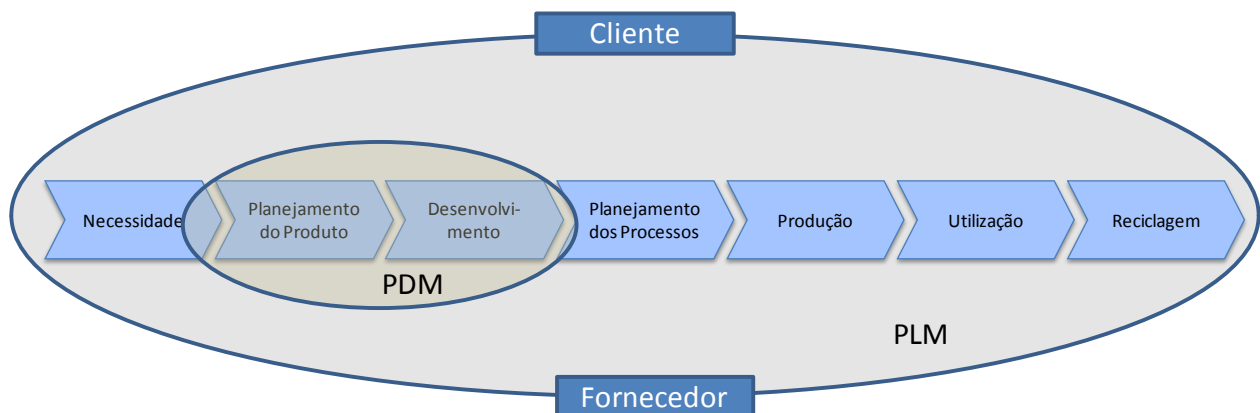


Figura 2.4 Etapas do ciclo de vida do produto, adaptado de (EIGNER e STELZER, 2009).

Dentro da abordagem PLM, os sistemas focados na gestão de dados ao longo do seu ciclo de vida podem ser referenciados tanto como PDM como sistemas PLM (ZANCUL, 2009). No entanto, devido a ser uma tecnologia recente, não há consenso entre os autores sobre a amplitude do uso de cada uma dessas siglas. Sistemas comerciais, por questões de mercado, por vezes usam as siglas de maneira indistinta (DAY, 2004).

É necessário ressaltar que PLM não inclui outros grandes sistemas como CRM (*Customer Relationship Management*), SCM (*Supply Chain Management*) e ERP (*Enterprise Resource Planning*), nem sistemas de apoio a outras funções de negócios, tais como marketing e vendas, distribuição, gestão de recursos humanos e finanças e outras. No entanto cada uma dessas funções ou sistemas interage com muitos componentes e soluções PLM (BRANDAO e WYNN, 2008).

2.3.2 Visão de Processos

Por ser um termo aplicável a vários domínios de conhecimento são várias as definições de processo encontradas na literatura, porém todas apontam para a raiz latina de algo que caminha para frente, que avança. No domínio da engenharia de produção é encontrada a definição de processo para qualquer atividade que recebe uma entrada, agrega algum valor e gera uma saída (BANDARA *et al.*, 2007). Especificamente para a manufatura, sempre mantida a ideia de atividade que avança em direção ao produto final, pode ser definida uma hierarquia de processos como na proposta de Giugliani e Varvakis (2007), atendendo ainda a concepção de níveis de planejamento descrita no item 2.3.

A proposta de hierarquização de processos descrita por Sordi (2005) é mostrada na Figura 2.5, onde o macro processo mais abrangente é decomposto em processos mais simples. Esta hierarquização é plenamente aderente à arquitetura de manufatura apresentada em 2.3.

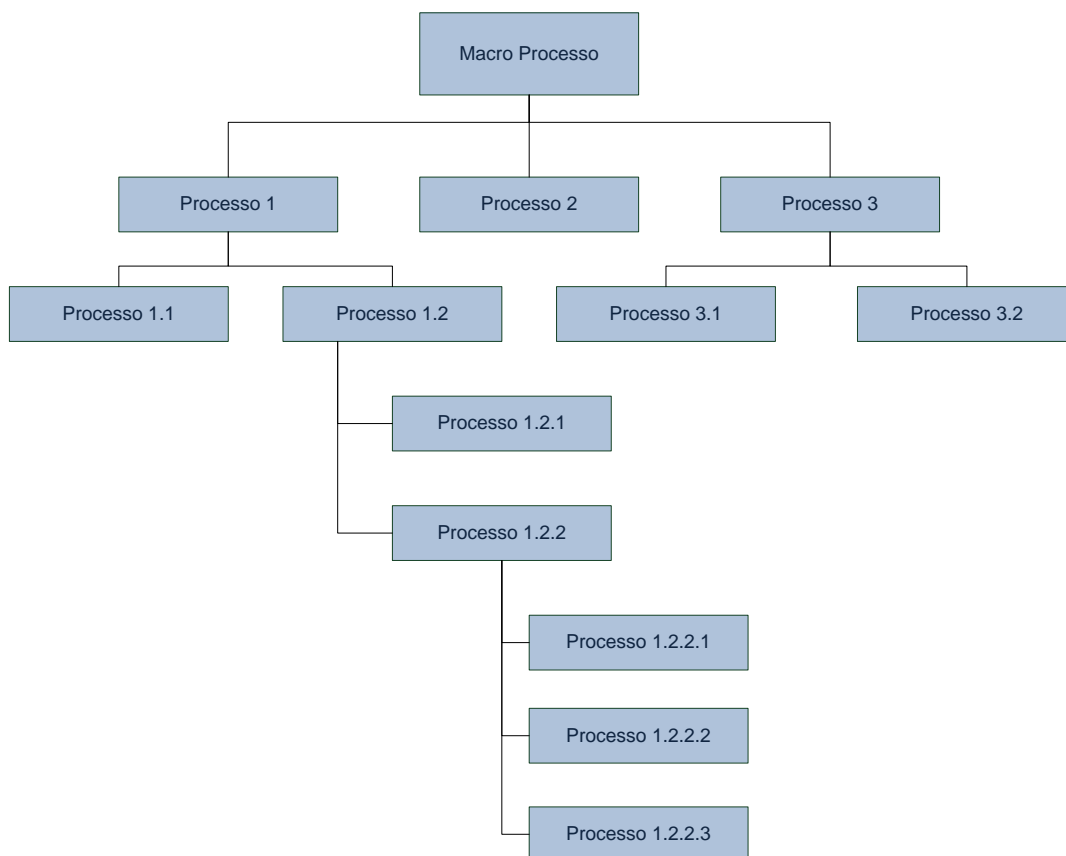


Figura 2.5 Hierarquização de Processos (SORDI, 2005).

A visão hierarquizada de processos gera, para cada nível de gerenciamento, dois tipos de processos, os processos internos no nível em questão e os processos que levam de um nível da hierarquia a outro. Para os processos que descrevem atividades dentro do mesmo nível hierárquico os recursos são aqueles disponibilizados também dentro do próprio nível hierárquico. Para os processos que descrevem atividades que levam de um nível hierárquico a outro, os recursos são os componentes do nível hierárquico imediatamente inferior (NIELSEN, 2003).

Segundo Hans-Erik e Penker (2000), um modelo de negócio é uma visão simplificada, que permite uma melhor compreensão do negócio e funciona como um plano para o seu gerenciamento. Assim um modelo não prevê todas as respostas necessárias à condução do negócio, porém delimita a busca por soluções e apresenta uma estratégia básica a seguir. Neste sentido o modelo de processos deve refletir os processos da organização que são executados por pessoas específicas ou automatizados por sistemas. Organizações são sistemas complexos que envolvem pessoas e equipamentos. Ao se construir o modelo de processos é preciso vincular o modelo à estrutura organizacional e contemplar os seus vínculos para uma descrição precisa das ações a serem consideradas (DUMAS *et al.*, 2005).

Inicialmente a modelagem de negócios no âmbito da manufatura foi introduzida com um olhar para a análise do fluxo de materiais e de suas atividades, com o objetivo de melhorar a qualidade do produto e reduzir os tempos de produção. O desenvolvimento de modelagem de negócios em outras áreas de conhecimento, como tecnologia de informação, automação empresarial, engenharia de software, entre outros, serviu para agregar à manufatura mais elementos (LIN *et al.*, 2002).

A visão do sistema produtivo através dos processos passa obrigatoriamente pela construção de um modelo de processo. Ele é feito usando técnicas de modelagem que descrevem de modo visual as atividades envolvidas nos processo e o fluxo de trabalho envolvido. Podem ser adicionados ao modelo os dados organizacionais, de tecnologia de informação ou de métricas a serem consideradas na avaliação da eficiência do próprio processo (BANDARA *et al.*, 2007).

Existem várias linguagens de modelagem disponíveis para a construção de modelos de processo, das quais se destacam como mais importantes: IDEF0/SADT, *Event-driven Process Chain*, *Process Modeling with UML*, *Business Process Modeling Notation*, *Integrated Enterprise Modeling*, *Process Specification Language* e *Semantic Object Model* (ANDERL *et al.*, 2008).

A linguagem de modelagem unificada (UML para a sigla em inglês de *Unified Modeling Language*) é uma linguagem construída sobre o paradigma da orientação a objetos, com propósito múltiplo e totalmente visual. Apesar de ser inicialmente projetada para modelar sistemas de software, pode ser utilizada para modelagem de negócios. Esta linguagem é, na verdade, um conglomerado de diagramas de vários tipos, o que torna um desafio a construção de uma plataforma única que acomode todos os diferentes diagramas, e seus inter-relacionamentos (DUMAS *et al.*, 2005).

O sistema de tecnologia de informação que gerencia os processos é conhecido pela sigla BPM (*Business Process Modeling*). É definido segundo Van Der Aalst *et al.* (2003) como um modelo de apoio ao processo de negócio usando métodos, técnicas e software para projetar, definir, controlar e analisar processos operacionais que envolvam pessoas, organizações, documentos e outras fontes de informação. Como pode ser visto na Figura 2.6, o gerenciamento dos processos do negócio envolve mais atividades que o gerenciamento do fluxo em uma organização.

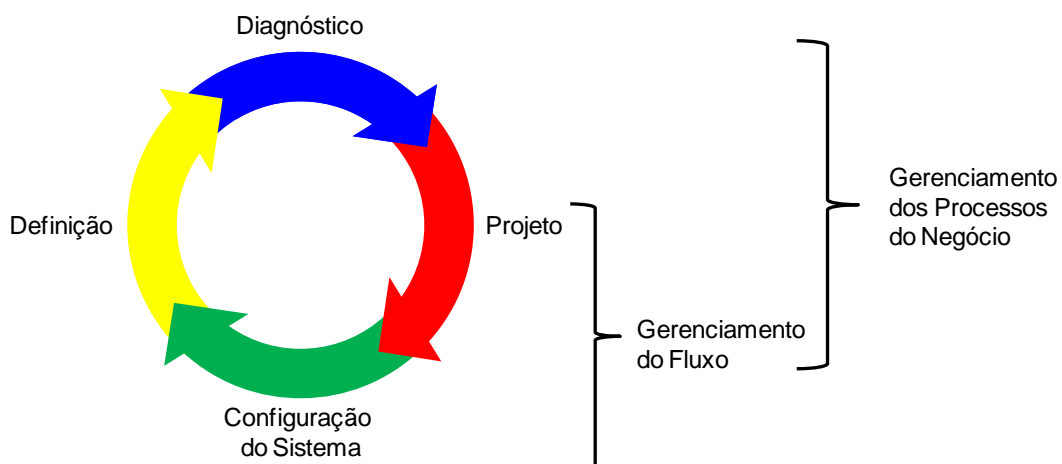


Figura 2.6 Ciclo de vida do BPM em comparação com o Gerenciamento do Fluxo, adaptado de (VAN DER AALST *et al.*, 2003).

A visão de processos é, dentro deste modelo, apresentada como ortogonal à visão de produto e também à visão de recursos, de tal forma que os vínculos se formam entre as visões, duas a duas, o que facilita a compreensão do modelo geral.

2.3.3 Visão de Recursos

Os recursos para a confecção de um produto, entendidos de modo amplo, compreendem matérias, ferramentas, máquinas, recursos humanos e administrativos. Estes recursos estão distribuídos em vários locais e sistemas, cada qual com diferentes fontes e capacidades. Cada um desses recursos desempenha um papel importante e ao mesmo tempo variável dentro do sistema de manufatura, sendo o gerenciamento essencial para o gerenciamento do sistema como um todo.

Segundo Chengying (2003) um modelo de informações dos recursos de produção deve ter, ao menos, as seguintes características: Flexibilidade; Coesão; Integridade e Disponibilidade. Este mesmo autor propõe um modelo de informação cuja arquitetura tridimensional é apresentada na Figura 2.7, onde os recursos de manufatura são classificados de acordo com a estrutura organizacional, o desenvolvimento do produto e a disponibilidade de uso. Este modelo tem, no eixo da Estrutura Organizacional, valores similares aos propostos no tópico 2.3 (Fábrica, Departamento, Célula, Estação de Trabalho e Máquina). Esta similaridade permite compatibilizar ambos os modelos, obedecendo inclusive à característica ortogonal entre os eixos. Em contrapartida, Yuan *et al.* (2008) propõem um modelo de informações para o gerenciamento da manufatura composto de modelos isolados. Nele, o modelo para os recursos contém as definições para os recursos humanos e tecnológicos que estão envolvidos ou disponibilizados para a execução de uma tarefa. Os vínculos com os outros modelos que compõem o modelo central do gerenciamento da manufatura são feitos dinamicamente para cada uma das atividades.

Estas duas propostas são igualmente válidas, sendo que a proposta de modelos isolados é mais aplicada a um sistema integrado, como será visto no tópico 2.7.1. Já a proposta de uma arquitetura tridimensional é melhor aplicada a um sistema distribuído, a ser discutido no tópico 2.7.3.

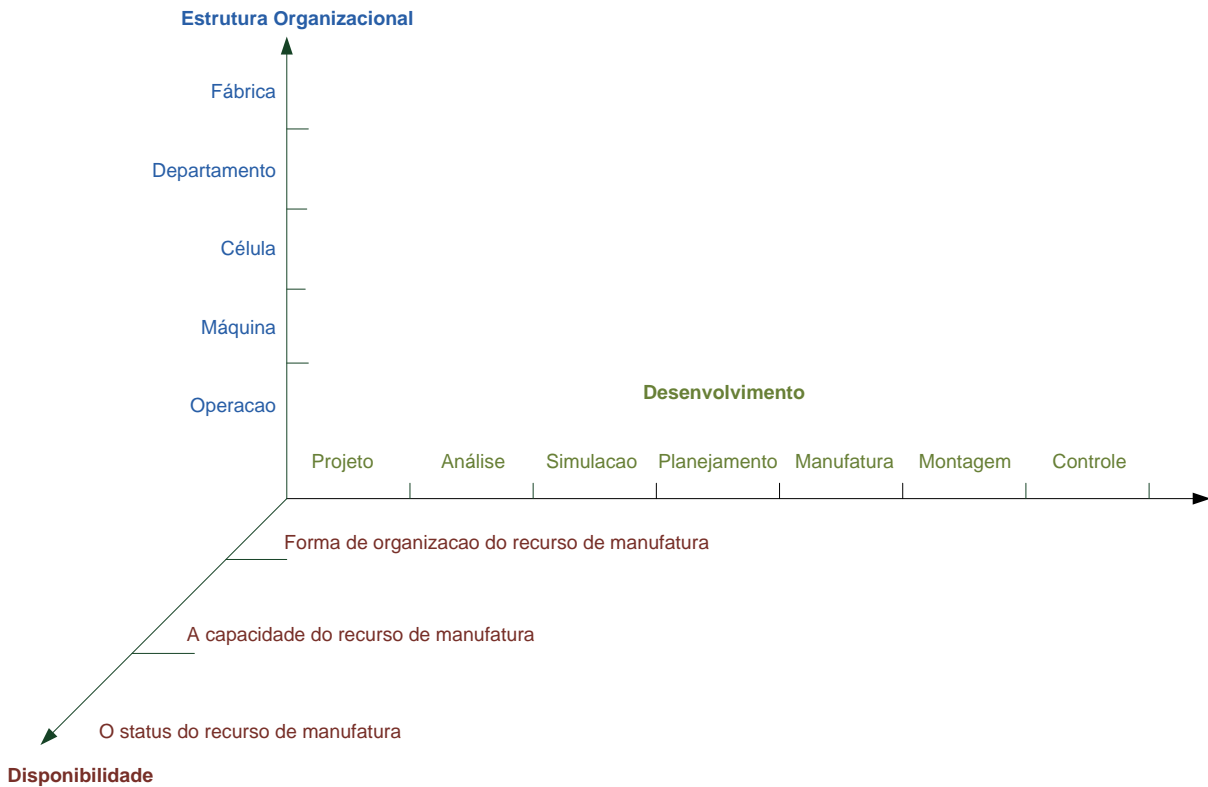


Figura 2.7 Arquitetura para Recurso de Manufatura, adaptado de (CHENGYING, 2003).

Os sistemas de tecnologia de informação para o gerenciamento dos recursos de manufatura tiveram uma evolução constante desde a lista de materiais (BOM - *Bill of Materials*) até os modernos sistemas de planejamento dos recursos empresariais (ERP - *Enterprise Resource Planning*) (CHUNG, S. e SYNDER, 1999). A automatização dos processos, possibilitada pela tecnologia da informação, permitiu que listas de materiais fossem gerenciadas em conjunto com os dados de estoque para um planejamento fabril, gerando os sistemas de planejamento de necessidades materiais ou MRP pela sigla em inglês *Materials Requirements Planning*, que são compostos de três elementos básicos: o plano mestre de produção; a lista de materiais; e as quantidades em estoque.

Com a publicação, em 1981 do livro *Manufacturing Resources Planning*, MRP II, de Oliver Wight (1981), houve um significativo aumento da abrangência do modelo, pois além de incorporar a possibilidade de uma análise grosseira para estabelecer uma relação direta entre o programa mestre de produção e a carga dos centros produtivos, e também a possibilidade dinâmica de ajustes na programação da produção, houve a

inclusão de outros recursos para a produção, como os recursos humanos e orçamentários. Com esta mudança houve também a mudança da denominação de “Planejamento das Necessidades Materiais” para “Planejamento dos Recursos de Produção” (LAURINDO e MESQUITA, 2000).

A evolução dos sistemas MRP II se deu com a incorporação de algumas funcionalidades como a possibilidade de cálculos mais complexos e a utilização de informações de outros sistemas. Neste ponto do desenvolvimento, quando integração passou a ser uma palavra-chave, houve um novo salto tecnológico e uma nova denominação para os sistemas que não somente gerenciam os recursos de manufatura, senão todos (ou pelo menos, a intenção de), os recursos empresariais. Este novo sistema é conhecido pela sigla ERP (*Enterprise Resource Planning*), e incorpora funcionalidades de finanças, custos, vendas, recursos humanos, e ainda outras, antes gerenciadas por sistemas isolados (HYPOLITO e PAMPLONA, 1999).

2.4 Heterogeneidade dos Dados de Produção

Sistemas de informação são desenvolvidos de acordo com o ciclo: análise; projeto; implementação e teste. Normalmente considerando uma única plataforma de banco de dados. Sistemas heterogêneos utilizam grandes quantidades de dados vindos das mais diversas fontes, assim a utilização destes dados enfrenta problemas de interoperabilidade que segundo Härder *et al.* (1999) é definida como a capacidade de cooperação entre sistemas heterogêneos. A solução para a interoperabilidade pode ser decomposta em dois passos: inicialmente é preciso extrair os dados das fontes primárias para a alimentação do modelo central e depois integrá-los. Para lidar com este problema alguns paradigmas foram desenvolvidos e são apresentados a seguir:

Paradigma de Objetos Distribuídos: é o paradigma central para a tecnologia CORBA (da sigla em inglês para *Common Object Request Broker Architecture*), onde cada objeto tem uma identificação. O código para a sua implementação é um estado, que é determinado pelo valor associado com um número de uma variável interna (SUJITHAN, 1995).

Infopipes: Um *infopipe* é uma plataforma de comunicação de fluxo de dados baseada na abstração produtor-consumidor. Este modelo inclui semântica e requisitos do usuário, o que permite o controle de fluxo e otimização de uso (BLACK *et al.*, 2002).

Peer-to-peer: São sistemas distribuídos constituídos de nós interconectados capazes de se auto organizar em uma rede topológica com o propósito de compartilhar recursos, tais como, conteúdo, processamento, armazenagem e distribuição. Também são capazes de se adaptar para suprir falhas e de acomodar novos nós sem que se perca a conectividade e o desempenho, sem que, para isso, seja necessária a interferência de um servidor central (ANDROUTSELLIS-THEOTOKIS; SPINELLIS, 2004).

Serviços Web: São módulos de software, independentes e auto descritos, acessíveis na Web (VIEIRA, 2006).

2.5 Teoria da Complexidade

A grande quantidade de dados e de sistemas isolados trabalhando em conjunto em um sistema produtivo leva a considerar a possibilidade de que se caracterize um sistema complexo, e como consequência as soluções para sistemas complexos devam ser aplicadas.

Durante o século XX a ciência da administração apresentou uma enorme evolução. Especificamente para a gestão de sistemas produtivos esta evolução foi marcante, a Figura 2.8 (BORGATTI NETO, 2007) apresenta uma representação desta evolução considerando as escolas de Administração Geral e de Gestão da Produção. Nota-se que as evoluções ocorrem em ciclos cada vez mais curtos, especialmente após a possibilidade de utilização das ferramentas de tecnologia de informação. Outro detalhe importante, destacado na figura, é que a base competitiva é acumulativa. Ou seja, inicialmente a base competitiva estava na eficiência e depois passou a ser eficiência e qualidade, sendo esta agregada àquela e não substituída, de tal forma que a competição se dá pelo acúmulo dos requisitos, e todos estes requisitos têm que ser atendidos simultaneamente. A influência da abordagem mecanicista vai da década de cinquenta até a década de noventa, quando se usa uma abordagem mais orgânica. A abordagem orgânica é essencial para o entendimento da complexidade.

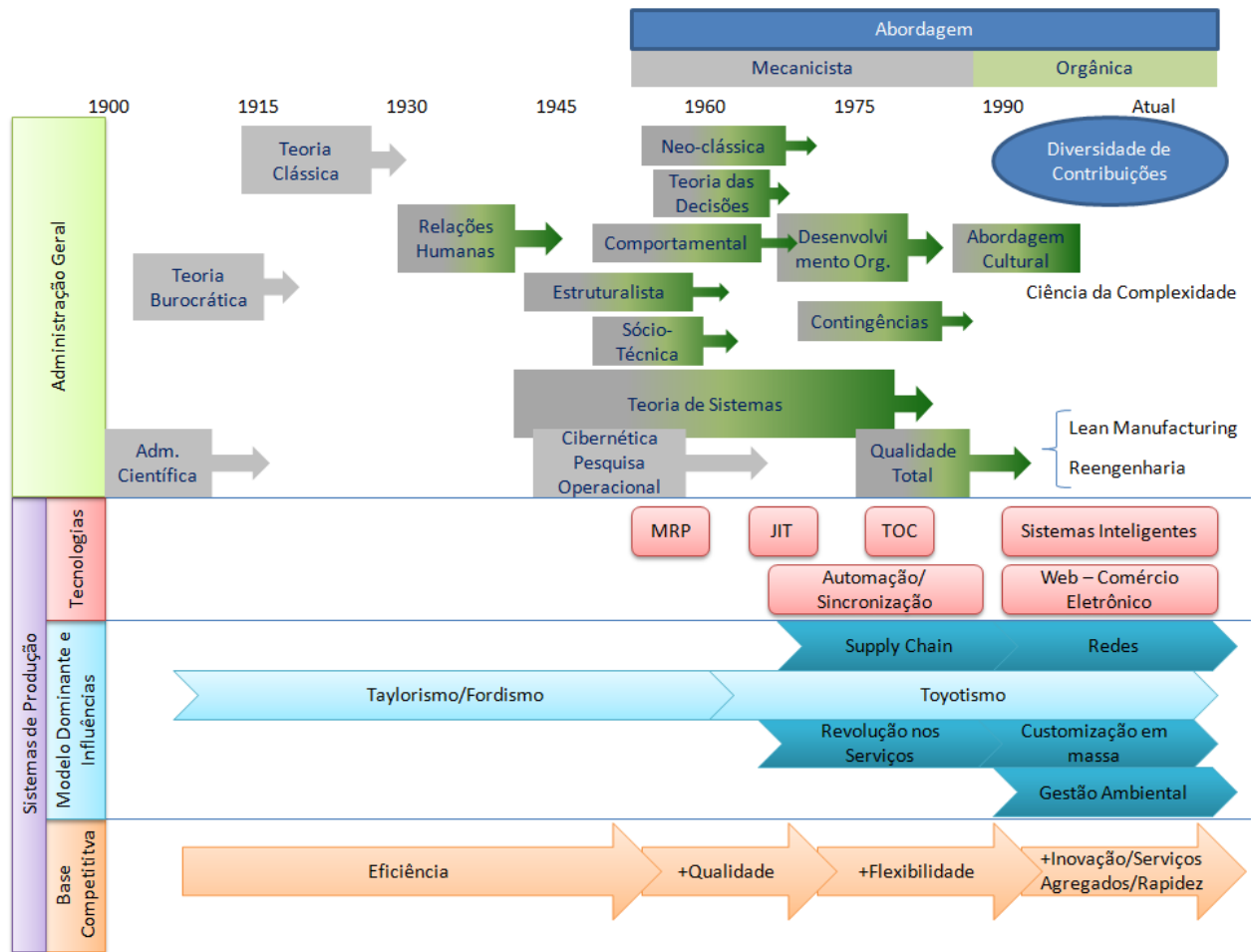


Figura 2.8 Representação da evolução da gestão (BORGATTI NETO, 2007)

A complexidade que teve início com os estudos de Heinz von Foerster em 1956 nos Estados Unidos, ao mesmo tempo em que estudos sobre causalidade circular, auto-referência e o papel organizador do acaso foram feitos por Ross Ashby, Warren McCulloch, Humberto Maturana e Gosdon Pask, entre outros (STADNICK, 2006). Um marco importante no estudo da complexidade é o lançamento, em 1977, do livro “Método 1 - A natureza da natureza”, de Edgard Morin onde são discutidos os temas da complexidade da natureza, da natureza desta complexidade e questões sobre ordem/desordem e organização (MORIN, 2007).

Segundo Morin (1997, p. 7), “a palavra complexidade não possui uma nobre herança filosófica, científica ou epistemológica. Suporta, pelo contrário, uma pesada carga semântica, porque transporta no seu seio, confusão, incerteza e desordem.” Portanto,

remete à ideia de que o conjunto interligado de componentes heterogêneos forma a complexidade. Parte deste entendimento advém da raiz etimológica latina de *complexo* para o que é tecido em conjunto, ou aquilo que é feito por várias mãos, conseqüentemente sem uma ordem inicial pré-estabelecida e assim, de difícil compreensão ou entendimento. Do ponto de vista de processo a complexidade pode ser entendida como um tecido de ações, interações, retroações, acontecimentos e acasos que constituem o objeto do estudo no campo desta teoria.

Este entendimento traz para o conceito de complexo a oposição a simples, porém a teoria da complexidade contempla diversos saberes, transitando de maneira interdisciplinar por várias outras ciências como a cibernética, a computação, a educação, a teoria de sistemas, a biologia, entre outras. Ciências estas que contribuiriam isoladamente para a construção de um saber sobre complexidade mais amplo (BESSEN *et al.*, 2009).

Segundo Borgatti Neto (2007), duas grandes correntes exploram o tema da complexidade; a corrente que prega a necessidade de um pensamento mais complexo, tendo uma abordagem mais filosófica e subjetiva, portanto, relacionada às paixões e complexidades das relações humanas. A outra corrente busca uma formalização científica estando embasada nas ciências naturais, na modelagem computacional e na matemática. Mesmo que entre estas duas grandes correntes apareçam ideias concorrentes ou mesmo conflitantes, são sustentadas por princípios comuns.

Uma propriedade emergente específica dos sistemas adaptativos complexos refere-se à sua capacidade de auto-organização espontânea. Quando ocorre um desajuste no ambiente, em face de alterações deste, os sistemas adaptativos complexos assumem estados de equilíbrio dinâmico. Na direção do limite do caos, onde os agentes buscam adaptar-se a novas situações do ambiente e co-evoluir para uma nova situação de ajuste.

Uma outra interpretação etimológica, porém, indica que o termo complexidade significa a característica daquilo que é composto de parte, o dicionário COLLINS (2010) faz uma recomendação especial para que o termo não seja usado erroneamente, indicando que

o uso correto serve somente para definir a característica daquilo que é composto de muitas partes e não deve ser usado como significado de que algo, só por conter muitas partes, é difícil de ser analisado ou compreendido

As pesquisas sobre complexidade, especialmente as desenvolvidas pelo *Santa Fe Institute*, incorporaram as ideias do caos, fractais, desordem e incertezas, o que fez surgir a proposição de um complexidade geral que congrega as ciências da complexidades sendo estas classificadas como complexidade restrita, já que não há um questionamento epistemológico da complexidade (MORIN, 2007).

Dentro do escopo da manufatura encontram-se muitas pesquisas que destacam alguns aspectos complexos, especialmente relacionando as interações internas e externas, o aumento da variabilidade dos produtos e do sempre crescente número de atores envolvidos no processo.

Segundo Urbanic e Elmaraghy (2006) a manufatura, de maneira ampla, pode ser entendida como um conjunto de relações entre o projeto do produto, os materiais, os equipamentos de produção e os sistemas de suporte, ou mais sucintamente pelo conjunto de relações entre produtos, processos e recursos. Estes elementos e suas ligações em diferentes níveis levam a um cenário de complexidade que pode ser considerada como a soma da compreensão e do gerenciamento de um grande conjunto de informações com grande variedade. A Figura 2.9 apresenta um modelo heurístico para a complexidade da manufatura introduzido pelos autores Urbanic e Elmaraghy (2006), composta de três elementos básicos: a quantidade de informação; a diversidade desta informação; e o conteúdo desta informação, que é a medida do esforço para obter o resultado desejado.

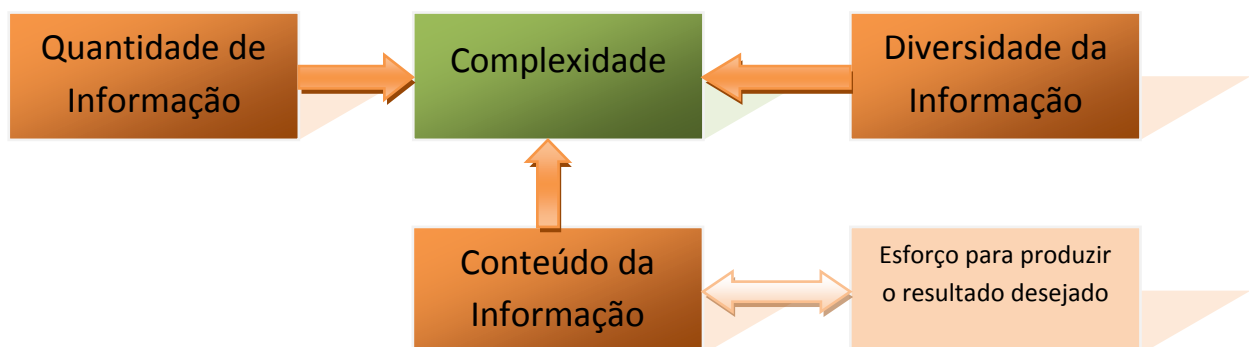


Figura 2.9 Modelo de complexidade, adaptado de (URBANIC e ELMARAGHY, 2006)

A causa desta complexidade pode ser atribuída à complexidade do produto em si que, segundo Tudorache (2006), tem origem na explosão de variantes advindas de um processo de customização em massa. Na literatura encontram-se muitas vezes a variedade com mesmo significado de complexidade e isto porque a maior parte da complexidade está na grande variedade de informações que devem ser gerenciadas (SCHLEICH *et al.*, 2007) e que é intensificada pelo fato de que os dados que descrevem o produto, os processos e os recursos para a manufatura serem originados de diferentes fontes (GRABOWSKI e MEIS, 1997).

A complexidade no ciclo de vida do produto pode ter pelo menos três consequências negativas (GRABOWSKI e MEIS, 1997):

- Falha na integração do desenvolvimento do produto;
- Perda de eficiência no processo;
- Problemas de cooperação entre os diversos atores do processo.

Estes problemas afetam diretamente a confiabilidade e qualidade do produto final, de tal forma que ações proativas devem ser tomadas para o gerenciamento dos dados do ciclo de vida do produto (URBANIC e ELMARAGHY, 2006).

Um sistema produtivo na sua concepção mais ampla considera também fatores transversais, em que agem fatores externos. Disso aumenta o número de interações e também a variedade das informações inseridas no processo, ao mesmo tempo em que exige dos sistemas que gerenciam estas informações um desempenho com maior coordenação. (SCHWARZENBACHER e WAGNER, 2005). Outro fator, não menos importante, é a participação de sistemas legados, já que historicamente um sistema produtivo trabalha com sistemas autônomos que são escolhidos muitas vezes por critérios operacionais e não estratégicos (MASSON, 2006). Estes sistemas legados devem ser incorporados ao sistema de gerenciamento de dados da manufatura fazendo com que estas informações sejam agregadas ao modelo central.

Pela geração dos dados através de sistemas isolados, pelos fatores externos e sistemas legados, o sistema produtivo apresenta um elevado grau de diversidade e de dificuldade de compreensão, o que leva autores, como os citados nestes tópicos, a considerá-la como uma atividade complexa.

Uma tentativa de romper com o determinismo linear de causa e efeito é feita por Morgan (1996) em sua metáfora que considera a organização como fluxo de transformação. Nesta metáfora as organizações são vistas como sistemas dinâmicos complexos operando em ambientes imprevisíveis e também respondendo de maneira imprevisível. Para esta metáfora são incorporados os princípios e conceitos da teoria do caos, da complexidade e da capacidade de gerar a si mesmo (autopoiesis), relacionando ideias com a dialética e com a cibernética.

Dentro do ciclo de desenvolvimento do produto, a cadeia CAD/CAM/CNC é a parte que se refere diretamente à manufatura. Mesmo considerando apenas sistemas de manufatura existem vários sistemas de gerenciamento de dados que permitem a integração entre sistemas que gerenciam dados oriundos de processos, recursos e produtos.

Se todos estes sistemas estiverem dentro de um mesmo contenedor, os dados podem ser armazenados em um banco de dados único, tornando as informações necessárias acessíveis a todos os sistemas. Porém, mesmo com todo o esforço dos maiores *players* do mercado em consolidar este modelo, agregando a um banco de dados monolítico todas as informações necessárias, não é esta a realidade encontrada no processo do desenvolvimento do produto atualmente (SCHLEICH *et al.*, 2007). As fases do desenvolvimento do produto são efetivadas, cada uma, por especialistas em uma rede altamente distribuída, estando conseqüentemente distribuídas as informações referentes a cada sistema. Com a atual possibilidade de acesso a todas estas informações, não se pode deixar de lado o conhecimento que está espalhado por várias pessoas, em diversas companhias e em diferentes localidades. Ao envolver estas diferentes pessoas ou organizações, nas quais todas controlam uma parte do processo e nenhuma tem o controle absoluto do todo, um sistema produtivo é visto como um

sistema colaborativo e distribuído, onde cada parceiro implementa o seu próprio processo de negócio (TEÓFILO e SILVA, 2005).

A plataforma de tecnologia de informação que apoia estes processos deve refletir este ambiente distribuído, de tal modo que garanta a informação correta na hora precisa para a tomada de decisão, independente do ator que detenha esta informação. Uma das soluções existentes para ambientes com alto volume de dados e muita automação, na detecção e análise do fluxo de processos é a utilização de máquinas de inferência (THOMASMA e NOLLER, 2008).

A complexidade também é discutida dentro do âmbito exclusivo do produto. Segundo Brown (2010) são cinco as dimensões críticas da complexidade do produto: a complexidade mecânica; a complexidade mecatrônica; o mercado global; a manufatura em si e o ciclo de vida do produto. Todas estas dimensões mostram um cenário de avanço da manufatura para um ambiente cada vez mais distribuído e dinâmico.

2.6 Banco de Dados

Os bancos de dados em tecnologia da informação surgiram na década de sessenta com os modelos hierárquicos e de rede, que realizavam as funções mais simples de organização, manutenção e geração de relatórios. Sendo puramente navegacionais e sem um embasamento teórico robusto, tinham dependência física e lógica dos dados (KHOSHAFIAN, 1994). A evolução destes modelos levou aos bancos de dados relacionais, sendo que na década de oitenta também surgiram, o modelo de dados semântico e o modelo de objetos complexos, que diferentemente do modelo relacional, não chegaram a ser usados comercialmente de modo significativo (CAREY e DEWITT, 1996).

A evolução seguinte dos modelos de banco de dados se deu através da possibilidade tecnológica de aplicação dos conceitos do paradigma de orientação a objetos.

Apesar de ser uma evolução em relação ao banco de dados relacionais, o modelo orientado a objeto não o substituiu, pois não tem algumas funcionalidades importantes do modelo relacional que continuavam necessárias (BOSCARIOLI *et al.*, 2006). A

adoção de funcionalidades relacionais aos bancos de dados orientados a objetos levou ao que se chama de banco de dados pós-relacionais. A Figura 2.10 apresenta um esquema da evolução dos bancos de dados.

Este trabalho utiliza banco de dados distribuídos com características pós-relacionais. Neste tópico não serão discutidos os bancos de dados de rede e hierárquico, por estarem muito distantes da aplicação.

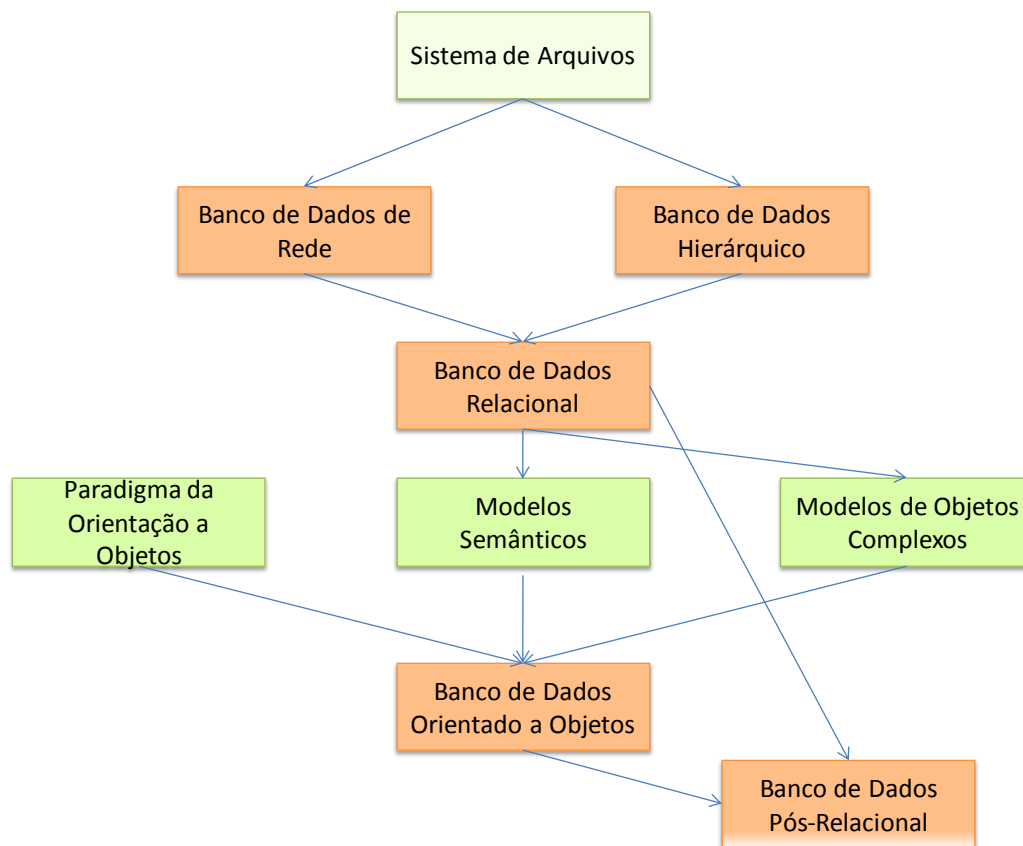


Figura 2.10 Evolução dos Bancos de Dados, adaptado de (KHOSHAFIAN, 1994).

2.6.1 Banco de Dados Relacionais

Bancos de dados relacionais são largamente utilizados devido à facilidade de uso e implementação aliado à independência de dados, podendo ser utilizado com grandes quantidades de dados. Um banco de dados relacional também provê um conjunto de funções para armazenamento e recuperação de dados através de álgebra relacional.

A origem do modelo relacional está ligada ao estudo teórico feito por Codd (1970) e sua grande utilização deve-se a flexibilidade e adequação aos problemas que se encontram na concepção e implementação de uma base de dados.

Estruturalmente, um banco de dados relacional fundamenta-se numa relação, ou tabela, constituída por um ou mais campos que informam o tipo de dados a armazenar. Cada linha da tabela é chamada de tupla ou registro. Não existem, no modelo relacional caminhos pré-definidos para a busca de dados, como nos sistemas hierárquico ou em rede, que antecederam o modelo relacional. No modelo relacional as estruturas de dados são organizadas em relações durante a implementação (FELICIANO, 2009).

Alguns cuidados devem ser tomados na construção de um banco de dados relacional para evitar repetição de informação (redundância) ou a inability de representação de certas informações, isto pode ser feito através da normalização seja usando dependências formais, dependências multivaloradas ou dependências na junção (SILBERSCHATZ *et al.*, 1999). A Figura 2.11 apresenta um exemplo simples de banco de dados relacional.

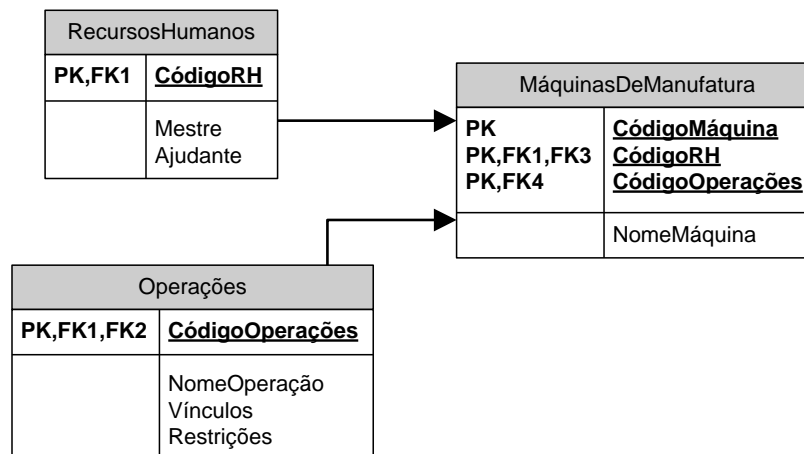


Figura 2.11 Exemplo de Banco de Dados Relacional.

2.6.2 Banco de Dados Orientado a Objetos

Apesar de ser uma tecnologia já estabelecida com várias aplicações em uso e ter uma linguagem de consulta específica (SQL *Structured Query Language*) que atende muito bem os processos de consulta, atualização e gerenciamento de bancos de dados

relacionais, estes nem sempre são úteis para aplicações de engenharia (KHOSHAFIAN, 1994).

Aplicações de engenharia se diferenciam de aplicações comerciais, no que concerne ao uso de sistema de banco de dados, essencialmente pelo fato de que o volume de operações com as informações é superior ao volume de dados armazenados, justamente o oposto das aplicações comerciais (DEBONI e MARTINI, 1997) e (KERN e BØHN, 1997). Banco de dados relacionais que atendem adequadamente às necessidades de aplicações comerciais apresentam alguns problemas nas aplicações de engenharia e até mesmo aplicações financeiras tradicionais têm algumas dificuldades em expressar suas necessidades em um banco de dados relacional (SILBERSCHATZ *et al.*, 1999).

Os bancos de dados orientados a objetos, por outro lado, têm uma tecnologia mais adequada para atingir as necessidades das aplicações de engenharia, com as seguintes características, que os fazem eficazes quando o volume de operações com as informações é muito grande e quando há um contínuo desenvolvimento do modelo (KHOSHAFIAN, 1994):

- Modelagem rica de dados;
- Compartilhamento de objetos;
- Permite a evolução em objetos e classes;
- Apropriados para transações em trabalho de projeto corporativo;
- Armazenamento de objetos distribuídos e independentes da plataforma

2.6.3 Banco de Dados Pós-Relacionais

Segundo Kim (1990), dois domínios em especial levaram ao uso de banco de dados orientados a objetos para superar a deficiência dos bancos de dados relacionais, quais sejam: o desenvolvimento de programação orientada a objetos e as aplicações de engenharia auxiliada por computador, também conhecidas pelo acrônimo CAx. Os

requisitos de combinação de dados destes dois domínios representam uma parte significativa dos requisitos que não são atendidos pelos bancos de dados relacionais.

Banco de dados orientados a objetos atendem aos requisitos de sistemas CAx de modo satisfatório (JAGENBERG *et al.*, 2009), especialmente no que concerne ao encadeamento complexo de objetos e à generalização.

No entanto, mesmo aplicações que requerem um modelo de informação simples podem demandar um grande volume de dados. Estas aplicações, com o incremento do uso de tecnologia da informação, tornaram-se mais e mais extensas, e passaram a ter requisitos não atendidos pelo modelo relacional (FONG, 2006).

Desta demanda surgiu o desenvolvimento em duas soluções: a extensão do modelo relacional para acomodar características do paradigma orientado a objetos, como apresentado por Sujithan (1995) e outros; e a inclusão de uma linguagem de consulta para o modelo orientado a objetos que permitisse consultas como em um modelo relacional (FITSILIS *et al.*, 1994), esta solução combina dados complexos baseados em um modelo relacional aninhado juntamente com os conceitos de orientação a objetos (identidade de objeto e herança) (SILBERSCHATZ *et al.*, 1999).

Um modelo orientado a objetos é uma base mais natural para este desenvolvimento do que a base de um modelo relacional, já que aquele comporta as ideias de tipos de dados genéricos, encadeamento de objetos e também pela capacidade de trabalhar com aplicações de uso intensivo (KIM, W., 1990). Vários estudos foram propostos para integrar ao modelo orientado a objetos as características relacionais, conforme destaca Kozaczynsky e Lilien (1987). Uma solução possível é a integração dos dois modelos como apresentado por Fong (2006), onde é adicionado um índice secundário para cada um dos modelos existentes. Outra possibilidade é a criação de um esquema específico para acomodar ambas as características, como proposto por Fitsilis *et al.* (1994) ou então, segundo Lentner e Kowalsky (2007), de uma linguagem de consulta especialmente construída para este fim.

Do ponto de vista estratégico é recomendado que sistemas de manufatura considerem a adoção de banco de dados pós-relacionais, já que incorporam as características mais importantes dos banco de dados orientados a objetos (herança, encapsulamento, abstração, reuso e troca de mensagens) ao mesmo tempo em que mantém a base formal de um banco de dados relacional (ZHANG, 2001) e (HARDING *et al.*, 2006).

2.7 Sistemas de Gerenciamento de Banco de Dados

O gerenciamento da manufatura por sistemas de informação dedicados pode ser entendido pelo contínuo fluxo de informações que são processadas pelos diversos atores do processo. Estas informações devem estar armazenadas em sistemas de banco de dados, que são propostos para resolver os problemas de integração de dados (LITWIN *et al.*, 1990).

2.7.1 Sistemas Centralizados

Um sistema de gerenciamento de banco de dados centralizado mantém todos os dados relativos à organização dentro de um único esquema interno. Visões, ou esquemas externos são utilizados para permitir o acesso de usuários ou grupos de usuários ao banco de dados central de acordo com a necessidade específica. Linguagens de consulta a dados são utilizadas para a criação destas visões, definindo assim uma imagem de uma quantidade limitada de dados (LITWIN *et al.*, 1990).

Para aplicações específicas que atendam aos requisitos de um usuário ou de um grupo restrito de usuários, são criadas visões parciais do banco de dados centralizado. Essas visões podem ser materializadas ou serem somente abstrações. São materializadas quando são feitas cópias dos dados para o uso em diferentes imagens de dados. Por outro lado, são abstrações quando a visão é produzida pelas transformações das requisições abstratas em requisitos concretos do banco de dados ou a outras visões em níveis inferiores (KEMPER *et al.*, 1990).

O usuário do banco de dados, ou de uma visão, deve saber o modelo de dados empregado e o seu esquema para fazer o acesso direto ao sistema de gerenciamento de banco de dados. Uma alternativa é a construção de aplicativos sobre o sistema para

facilitar o uso. Um nível elevado de coordenação é necessário para a criação de integração de sistemas de banco de dados centralizados, sendo que esta coordenação é muito difícil de ser obtida, mesmo que envolva alguns departamentos de uma mesma corporação (THOMAS, G. *et al.*, 1990).

2.7.2 Sistemas Heterogêneos

Sistemas de banco de dados heterogêneos são aqueles que integram vários sistemas de banco de dados já existentes. Para o sistema de banco de dados heterogêneos, os sistemas agregados são chamados componentes e podem ser, desde sistemas simples até um outro sistema heterogêneo. Os componentes podem ser desenvolvidos e gerenciados de forma totalmente independente.

A heterogeneidade de um sistema pode ser caracterizada, segundo Sheth e Larson (1990), por três eixos ortogonais: autonomia; integração e a heterogeneidade em si. A autonomia se refere à liberdade que cada componente do sistema tem para definir e gerenciar seus dados, sendo que a necessidade de se manter a autonomia ao mesmo tempo em que se precisa compartilhar dados é por muitas vezes conflitante. A integração, objetivo principal, não deve bloquear totalmente a capacidade de manutenção de um componente, ainda segundo Sheth e Larson (1990), a autonomia pode ser classificada em quatro categorias:

- de projeto: compreende a autonomia de cada componente para definir o seu projeto.
- de comunicação: compreende a capacidade de cada componente para determinar o modo de comunicação com outro componente. Um componente com autonomia de comunicação pode decidir quando e como responder a uma requisição de um outro componente.
- de execução: refere-se à possibilidade de um componente executar operações (locais ou externas), com total controle do processo de transação;

- de associação: define se um componente pode, de modo independente, decidir quais informações trocar com outro componente; para quais requisições irá responder e quando inicia ou termina a participação no sistema que o agregou.

A distribuição, enquanto característica de heterogeneidade, está relacionada à localização do componente, um sistema distribuído pode envolver diversas plataformas de hardware, software e de comunicação. Mesmo que em princípio, ou em teoria, a distribuição seja ortogonal à heterogeneidade, deve-se esperar que sistemas tenham maiores discrepâncias quando mais distribuídos forem (FILETO, 2003).

A heterogeneidade em si caracteriza o número de discrepâncias significantes entre os componentes, com relação aos esquemas, modelos de dados, linguagem de consulta, capacidade de gerenciamento de transações, hardware, sistemas operacionais e protocolos de comunicação. Estas discrepâncias podem existir em qualquer nível de abstração (instâncias, esquema ou modelos de dados) (HÄRDER *et al.*, 1999).

2.7.3 Sistemas Distribuídos

Os sistemas distribuídos fazem a integração das informações através da integração de esquemas, criando um esquema único para os bancos de dados distribuídos através do qual todos os usuários acessam os dados integrados. A Figura 2.12 apresenta o funcionamento de um sistema distribuído. O esquema de cada um dos bancos de dados pertencentes ao sistema distribuído é uma visão para o esquema unificado, através do qual as aplicações isoladas obtêm as informações para construção do modelo central (FILETO e MEDEIROS, 2003).

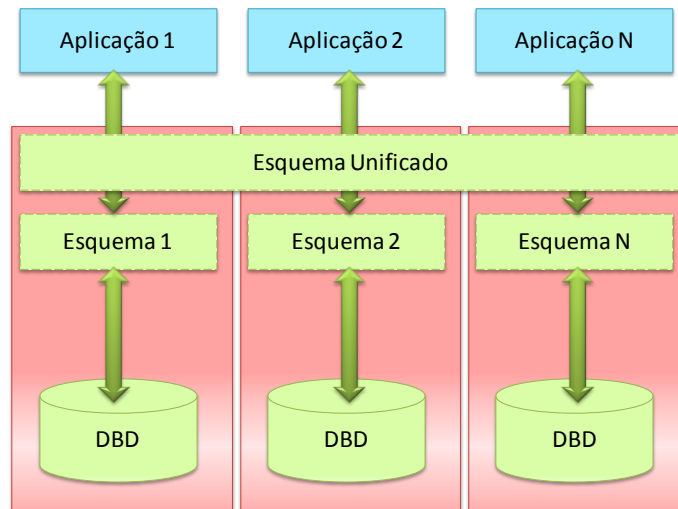


Figura 2.12 Sistema de Banco de Dados Distribuídos (FILETO e MEDEIROS, 2003).

Segundo Litwin *et al.* (1990), um sistema distribuído pode ser considerado como um único banco de dados que está fisicamente distribuído em vários locais. Apesar das aplicações estarem em locais diversos, o sistema distribuído funciona com somente um modelo de dados e uma única linguagem de consulta, com um único esquema.

2.7.4 Sistemas Federativos

Sistemas federativos, em oposição aos sistemas distribuídos, somente têm significado com a exportação de esquemas ou visões, o que deixa para o usuário uma parte do trabalho de integração dos dados. Na Figura 2.13 é apresentado o modelo de funcionamento de um sistema de bancos de dados federativos, no qual cada aplicação utiliza um esquema para a obtenção dos dados e os esquemas para importação e exportação de dados são gerenciados externamente ao sistema. Enquanto que um sistema distribuído é transparente ao usuário no que concerne à integração, o sistema federativo permite um grau de autonomia muito mais elevado para os sistemas componentes, de tal forma que cada aplicação possa ser utilizada sem grandes alterações; para o usuário final o processo é transparente já que as adaptações são feitas internamente. Segundo Fileto e Medeiros (2003) um sistema federativo é um caso especial de sistema distribuído em que é permitido a componentes heterogêneos trabalhar com diferentes modelos de informação, esquemas e visões.

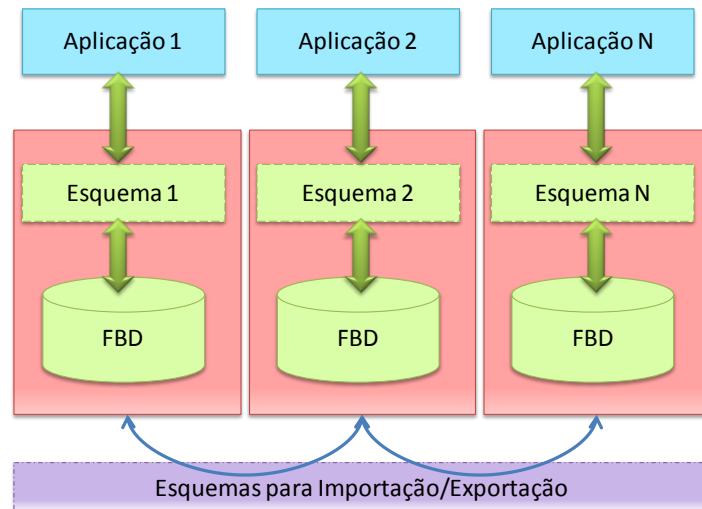


Figura 2.13 Sistemas de Bancos de dados federativos (FILETO e MEDEIROS, 2003).

2.7.5 Integração de Dados

A integração dos dados em um sistema centralizado não é uma questão relevante visto que os dados estão todos centralizados em um único local. Para dados heterogêneos a integração é uma questão importante, já que estes dados apresentam diferenças em relação à representação e interpretação, mesmo que se refiram a mesma realidade (LITWIN *et al.*, 1990). Conflitos de heterogeneidade podem causar incapacidade de obter os dados desejados, o que leva a ausência de informações (do ponto de vista do usuário ao menos) ou a redundância de dados na tentativa de solucionar este problema. Para a integração de dados heterogêneos é necessária uma visão única para um conjunto de dados, de modo a unificar a sintaxe, a estrutura e a semântica, desta forma resolvendo os conflitos existentes. Para tanto há que se prover transformação dos dados na sua fonte para adequação a esta visão única (MOSCH *et al.*, 2010).

O processo de integração de dados heterogêneos passa pela categorização dos dados disponíveis para a construção do modelo central e dos conflitos de heterogeneidade. Então, os conflitos podem ser solucionados em sequência para cada uma das categorias.

Dos sistemas apresentados pode-se verificar que conquanto sistemas distribuídos tenham elevado grau de distribuição e autonomia entre as aplicações, há um baixo grau de flexibilidade e de integração já que cada aplicação trabalha de modo isolado e as

comunicações tem que ser feitas entre cada aplicação e o modelo central. De outra forma o sistema integrado mantém um elevado grau de integração pois as aplicações ficam internalizadas no modelo, isso contudo prejudica a autonomia e a distribuição, porque as aplicações ficam dependentes do modelo central, o que acaba também por prejudicar a flexibilidade do sistema. A solução intermediária é a utilização de um sistema federativo que preserva a autonomia do sistema distribuído e a integração do sistema integrado permitindo uma maior flexibilidade do sistema. Nesse caso o modelo central é construído dinamicamente a cada alteração dos modelos parciais. A Figura 2.14 apresenta esquematicamente a comparação entre os sistemas apresentados (ABELN, 1997; MONTAU, 1995 e PEDERSON, 2005).

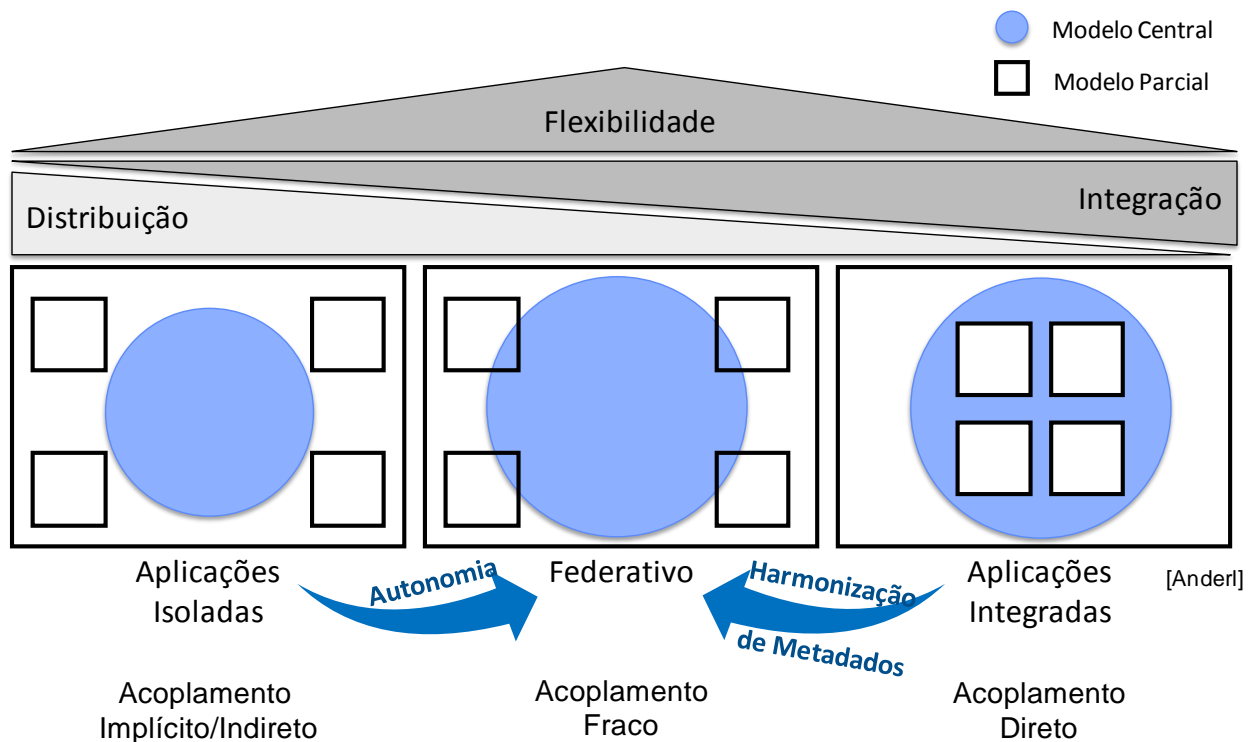


Figura 2.14 Comparação entre sistemas de integração. (ABELN, 1997; MONTAU, 1995; PEDERSON, 2005)

2.8 Considerações Finais do Capítulo

Neste capítulo foi apresentado o sistema produtivo com especial atenção sobre o sistema de manufatura. O modo de compreender o sistema produtivo em três visões é uma solução promissora para o gerenciamento do grande e crescente volume de dados

de um sistema de aparência complexa. Também foram exibidos as características dos dados existentes e os modos de gerenciamento. Percebe-se que o grande volume de dados é um obstáculo à integração dos vários sistemas componentes do sistema produtivo e que essa falta de integração causa falhas e dificuldades tanto no ciclo de vida do produto quanto nos processos produtivos.

O próximo capítulo discute as tecnologias disponíveis para o gerenciamento de dados ao longo do ciclo de vida do produto.

3 Tecnologias de Informação utilizadas durante o ciclo de vida do produto

O ciclo de vida do produto pode ser decomposto em vários sistemas, cada qual com suas aplicações e com suas ferramentas de TI específicas. Cada componente gera, de forma autônoma, os dados referentes às suas aplicações e os gerencia independentemente dos outros componentes, a seguir são apresentados os sistemas estudados neste trabalho e as tecnologias utilizadas para a integração das informações.

3.1 Aplicativos *Add-on* para sistemas CAx

Sistemas comerciais de auxílio computacional para as diversas áreas da engenharia como projeto, manufatura, qualidade, planejamento e outras estão disponíveis e em constante evolução. Estes sistemas são conhecidos pelas siglas compostas das duas letras iniciais CA para o termo em inglês “*Computer Aided*” (auxiliado por computador) e por uma terceira letra indicando a especificidade deste auxílio como D para projeto (*Design*); M para manufatura (*Manufacturing*); E para métodos de engenharia (*Engineering*), entre outros. Para designar o conjunto de sistemas auxiliados por computador convencionou-se usar a sigla CAx, onde o “x” serve para significar qualquer sistema auxiliado por computador.

Sistemas CAx comerciais atendem um grande espectro de necessidades das atividades de engenharia, mesmo assim não são capazes de atender a todas as especificidades de aplicações possíveis. Para o atendimento de necessidades muito específicas alguns sistemas CAx permitem a criação de aplicativos e sua agregação ao sistema. Desta forma se há a necessidade de uma característica ou funcionalidade especial esta pode ser desenvolvida e agregada ao sistema.

3.2 Projeto Auxiliado por Computador

Os sistemas de Projeto Auxiliado por Computador, conhecidos por CAD pela sigla em inglês de *Computer Aided Design*, evoluíram muito rapidamente e conjuntamente com o desenvolvimento da tecnologia da informação. O que nos primórdios era somente uma ferramenta de desenho efetuando as mesmas atividades feitas em uma prancheta, hoje

evoluiu para a possibilidade de representação de sólidos em três dimensões e com uma interface bastante intuitiva (KRAUSE *et al.*, 1993), (MCMAHON, 1998).

Um sistema CAD atual é capaz de armazenar dados não somente geométricos, mas também informações de dimensionamento e tolerâncias geométricas, informações tecnológicas como: material; processos de usinagem, tratamento térmico entre outros.

De maneira clássica o desenvolvimento de um produto pode ser entendido como um processo iterativo, com repetição das atividades de síntese, análise e otimização e avaliação, como descrito em 1977 por Shigley (1977):

Apropriação da necessidade: Recolhimento ou percepção de uma necessidade de mercado que possa ser atendida por um produto ou serviço. Etapa desenvolvida normalmente pelos setores que estão em contato com o mercado, no cenário proposto esta necessidade é apropriada pelo contato com o cliente que insere no sistema as dimensões do cilindro desejado.

Definição do problema: Transformação da necessidade apropriada na etapa anterior em dados tangíveis. A necessidade é abstrata, o problema é concreto e pode ser expresso em um documento que contenha as informações para elaboração do produto. Considerado o cenário de avaliação e teste esta etapa é feita em conjunto com a anterior, com os dados do cilindro sendo preenchidos diretamente no sistema.

Síntese: Nesta etapa a necessidade, consolidada em um problema definido, é sintetizada em uma solução inicial. Devido ao elevado grau de automação proposto no cenário deste trabalho, esta etapa é feita automaticamente com base no conhecimento de engenharia inserido no sistema e uma primeira versão do cilindro é apresentada sem interferência humana.

Análise e Otimização: Em conjunto com a etapa anterior, esta etapa é iterativa, já que a solução inicial proposta anteriormente é analisada e sugestões de melhorias podem ser inseridas no produto em desenvolvimento.

Avaliação: Estando maduro o suficiente, o produto desenvolvido é submetido à avaliação, que pode ser interna a quem desenvolve o produto ou externa, envolvendo o cliente final. Esta etapa pode ser incluída no ciclo iterativo com as duas etapas anteriores.

Apresentação: O produto avaliado é considerado maduro o suficiente e uma apresentação formal é disponibilizada na forma de um documento geral e também de protótipo através de um sistema de prototipagem rápida ou de realidade virtual.

A Figura 3.1 apresenta o ciclo de desenvolvimento do produto clássico segundo Shigley (1977).

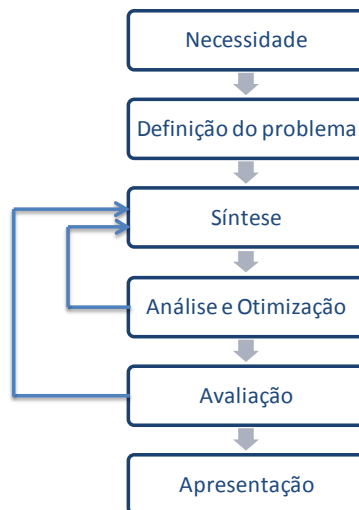


Figura 3.1 Ciclo de Projeto Clássico, adaptado de (SHIGLEY, 1977).

Na década seguinte Groover e Zimmers (1983), com a possibilidade do uso de ferramentas computacionais, relacionam quatro das etapas propostas no ciclo de projeto clássico que podem ser elaboradas diretamente no sistema CAD, que são:

- Modelagem geométrica;
- Análise de engenharia;
- Revisão do projeto e avaliação;
- Geração automática dos desenhos.

A modelagem geométrica resume a etapa de síntese quando uma proposta inicial é elaborada, o resultado desta etapa, em um sistema CAD é um modelo geométrico em três dimensões. Para a etapa de análise e otimização está relacionada a etapa de análise de engenharia que pode ser feita em um sistema de engenharia auxiliada por computador, ou CAE pela sigla em inglês para *Computer Aided Engineering*. A avaliação do produto durante o desenvolvimento pode ser feita diretamente no sistema CAD através da modificação do projeto seja pela análise de engenharia seja para atender a requisitos do cliente. Por fim, a apresentação do produto se dá automaticamente através da geração do modelo em realidade virtual ou pela transmissão dos dados do sistema CAD para a construção do protótipo. A Figura 3.2 apresenta as relações entre o sistema clássico e com o uso de ferramentas computacionais.

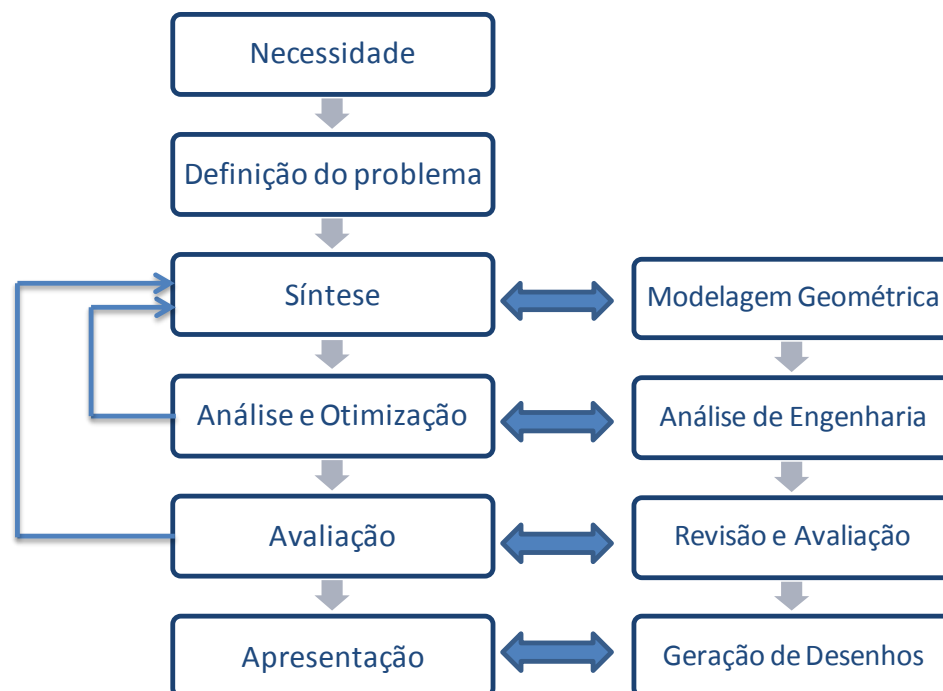


Figura 3.2 Comparação entre sistema clássico e computacional, adaptado de (GROOVER e ZIMERS, 1983).

Atualmente a utilização de ferramentas computacionais no desenvolvimento do produto é intensiva e a abordagem requer uma efetiva colaboração interdisciplinar para o desenvolvimento de produtos multidisciplinares, isso só é possível através da integração de ferramentas computacionais específicas para modelagem, análise e

simulação entre outras, conforme descrito por Anderl *et al.* (2010) que sugere uma estrutura em V para esta integração, conforme apresentado na Figura 3.3.

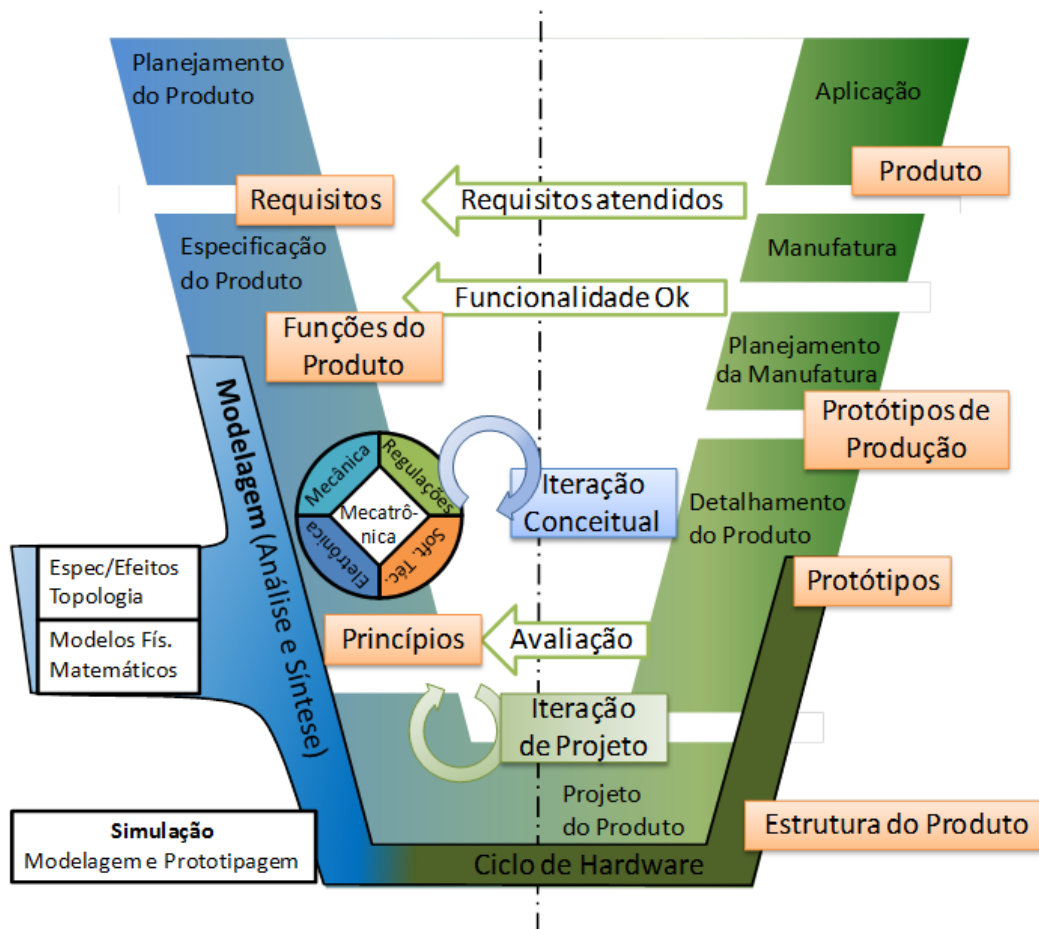


Figura 3.3 Modelo em V para o desenvolvimento multidisciplinar do produto, adaptado de (ANDERL *et al.*, 2010).

O processo de desenvolvimento do produto com ferramentas computacionais pode iniciar já na apropriação da necessidade através de ferramentas de relacionamento com o cliente, ou CRM pela sigla em inglês para *Customer Relationship Management*, onde são coletados dados de reclamações, sugestões, pesquisa, etc. (FELICIANO, 2009). A definição do problema também pode ser feita através de um sistema de engenharia baseada no conhecimento (KBE - *Knowledge Based Engineering*) (PRIETO JUNIOR, 2002). Estas duas etapas estão mais distantes do escopo deste trabalho e não serão tratadas aqui, as demais são descritas a seguir.

A modelagem geométrica é a etapa na qual um modelo geométrico do produto é construído em três dimensões e com informações tecnológicas agregadas. Os sistemas

CAD modernos têm capacidade para criação deste modelo de modo intuitivo, a agregação do sistema de engenharia baseada no conhecimento advindo da etapa anterior permite a criação automática deste modelo (WESTKÄMPER e BRIEL, 2001).

A análise de engenharia considera o modelo geométrico criado na etapa anterior e verifica se o modelo atende aos requisitos estabelecidos. A integração entre os dados do modelo CAD e do modelo para CAE pode ser feita das seguintes formas: o uso de um modelo comum; de um modelo neutro ou de tradutores entre os modelos (GUJARATHI e MA, 2011). Nesta etapa já se considera a iteratividade, pois os resultados da análise (etapa posterior) são considerados para melhorias no modelo geométrico (etapa anterior).

Com a aprovação do modelo pela análise de engenharia há a necessidade de revisão e avaliação seja interna ou externa, para tal podem ser construídos modelos em sistemas de foto realismo para considerações de ordem estética, ou protótipos com o uso de máquinas de prototipagem rápida para avaliações de ergonomia e funcionalidade (HALTTUNEN e TUIKKA, 2000).

O produto já maduro, aprovado pelas etapas anteriores e apresentado formalmente com o uso das ferramentas existentes e pode ser transferido para o processo produtivo, novamente questões relativas à interoperabilidade e transferência de dados se apresentam e precisam ser solucionadas. Cabe ressaltar que com o uso dos conceitos de engenharia simultânea, considerações sobre a maturidade do produto para início das etapas seguintes não necessariamente compreende a aprovação definida, as etapas podem e devem se sobrepor (CHANDRA e KAMRANI, 2003); (NI *et al.*, 2010) e (GUJARATHI e MA, 2011).

3.3 Manufatura Auxiliada por Computador

O início dos sistemas de manufatura auxiliada por computador (CAM – *Computer Aided Manufacturing*), segundo Rembold *et al.* (1993) se dá na década de 50 com o desenvolvimento de máquinas numericamente controladas. Cartões perfurados continham a informação das coordenadas e de alguns poucos comandos como

velocidade de avanço e rotação do eixo, que eram interpretadas por comandos eletromecânicos para o funcionamento das máquinas-ferramenta.

O sistema CAM, em sua tradução literal: manufatura auxiliada por computador, pode ser entendido como a preparação da manufatura feita com o auxílio de computador, representando as tecnologias utilizadas no chão de fábrica, não se restringindo somente a automação da manufatura, mas também compreendendo o comando numérico programável, o controle lógico programável, coletores de dados, tomada de decisão e plano operacional entre outros (KOCHAN, 1984).

Sistemas CAD/CAM comerciais reforçam a ideia restritiva de CAM como sinônimo de programação NC, ao vincular a sigla CAM ao módulo de programação do comando numérico (RIBEIRO e ROZENFELD, 1999).

A saída do pós-processador em um sistema CAM é um arquivo que contém as coordenadas e as trajetórias a serem seguidas pela ferramenta para produção do produto, além dos comandos de velocidade de avanço, de corte e outros comandos auxiliares.

Ainda segundo Kochan (1984), o desenvolvimento dos sistemas CAD evidenciou a necessidade de integração com o sistema CAM para a automatização do processo de manufatura, porém a deficiência de uma interface amigável e intuitiva e a dificuldade, para os projetistas, em trabalhar com sistemas operacionais fez com que, durante uma década o uso dos sistemas CAM ainda não fosse significativo.

Com o desenvolvimento da tecnologia de informação provendo sistemas operacionais mais amigáveis, maior capacidade de processamento e memória, o uso de sistemas CAD e CAM se ampliou, assim como a integração dos sistemas. Some-se a isso o fato de que os sistemas de controle das máquinas também tiveram uma rápida evolução com o uso de motores lineares, *transcoders* com maior precisão e comandos numéricos muito mais potentes, com capacidade de trabalhar com interpolações circulares e NURBS (HELLENO, 2008).

A integração entre sistemas de desenvolvimento do produto, especialmente no que concerne ao par CAD/CAM tem sido objeto de vários estudos e são muitas as propostas, inclusive de integração através da Web (ALVARES e FERREIRA, 2003); (CHAO e WANG, 2001) e (BENAVENTE, 2007).

3.4 SOA - Arquitetura Orientada a Serviços

O ciclo de vida do produto é composto de atividades cada vez mais colaborativas e distribuídas, desenvolvidas em unidades de produção flexíveis e adaptáveis, e levada a cabo em processos cada vez mais inter-relacionados e independentes de tempo e de espaço, nos quais são aplicados os conceitos de engenharia simultânea. Estas características possibilitam produtos complexos e inovadores disponíveis ao mercado em tempo cada vez mais curto. O cenário no qual se insere o ciclo de vida do produto tende a ser entendido como um cenário de complexidade, principalmente ao se levar em conta o tipo de produção, sistemas ou aplicativos envolvidos, cada qual com suas características próprias. Além destes fatores, a presença de mais de uma fábrica, no mesmo cenário, operando um sistema diferente de software, só aumenta esta possibilidade de complexidade (MASSON, 2006).

Para se obter uma plataforma mais flexível, capaz de se adaptar rapidamente às necessidades de mercado, respondendo a pressões, como as de lançamento de novos produtos no mercado, é imprescindível o desenvolvimento de ferramentas de TI capazes de trabalhar com elevado volume de dados em um ambiente descentralizado (MING *et al.*, 2008).

Os dados que vêm de várias e independentes ferramentas de TI são, por sua própria natureza, heterogêneos, uma vez que cada aplicação isolada tem autonomia para gerar e gerenciar os dados. A integração dos dados heterogêneos requer uma resolução de conflitos de heterogeneidade e a transformação de documentos e fontes de dados em um conceito integrado (LUBELL *et al.*, 2004). Os sistemas de gestão de dados para domínios específicos são bastante maduros, mas não a colaboração entre eles. Os demais domínios têm de colaborar, os demais problemas em integração de dados surgem, os quais devem ser abordados.

Devido à necessidade interação de entre sistemas, a manufatura compartilha informações entre várias aplicações, tanto as de manufatura, quanto as de negócio, dentro do conceito de corporação estendida, pois contempla a cadeia de suprimento e os desenvolvedores independentes. A troca de dados em si já é um problema de difícil resolução, compartilhar informações é um desafio ainda maior. Segundo Abbas (2008) o compartilhamento das informações para a manufatura depende dos seguintes aspectos: semântica, normas, globalização e pessoas, sendo que a construção de uma semântica é a base para a utilização dos serviços Web.

Como solução para este ambiente heterogêneo, utiliza-se a Arquitetura Orientada a Serviços – SOA (*Services Oriented Architecture*), que torna possível a migração de um sistema monolítico para um sistema distribuído.

Uma arquitetura de software é um conjunto de definições que descrevem componentes de software e determina a funcionalidade do sistema que contém estes componentes. São definidas as estruturas, as restrições e as características dos componentes e de seus vínculos, segundo a definição de Krafizg *et al.* (2005). A escolha, ou desenvolvimento, de uma arquitetura de software adequada aos requisitos do projeto é um dos pontos-chave para o sucesso, pois deve contemplar a criação de novas funcionalidades e também a possibilidade de alteração das funcionalidades existentes e também a adequação a sistemas legados e sistemas futuros (GARLAN, MONROE & WILE, 1997). Dentro deste escopo a arquitetura orientada a serviços lida com os processos distribuídos em sistemas heterogêneos; facilitando a interação entre os provedores de serviços e os consumidores de serviços. Esta arquitetura possibilita que os agentes localizem e utilizem as capacidades distribuídas de que precisam, mesmo que estejam sob controle de diferentes proprietários (BOOTH, 2004).

Devido a estas características, SOA é uma abordagem promissora dentro do ambiente de TI. Pode ser descrita com as seguintes características (ABRAMOVICI e BELLALOUNA, 2008):

- SOA é um processo controlado e orientado, no qual a integração de processos é o principal alvo.

- SOA é independente. Nenhuma tecnologia específica ou para o ambiente de TI é necessária para sua implementação, deste modo pode-se afirmar que SOA é extensível.
- SOA leva a um aumento da conectividade em ambientes heterogêneos (processos, dados e aplicações). Assim, a interligação dos vários sistemas de TI heterogêneos existentes serão melhorados.

Por manter camadas sem acoplamento, os serviços podem ser disponibilizados e acessados de forma independente, o que vai ao encontro do objetivo de um sistema que trabalho de modo distribuído, preservando a autonomia tanto do fornecedor quando do consumidor dos serviços disponíveis.

Segundo Krafzig *et al.* (2005) a arquitetura orientada a serviços contém os seguintes elementos: aplicação; serviço; repositório de serviços; canal de serviços; contrato; implementação; interface; domínio de negócio e dados, como apresentado na Figura 3.4

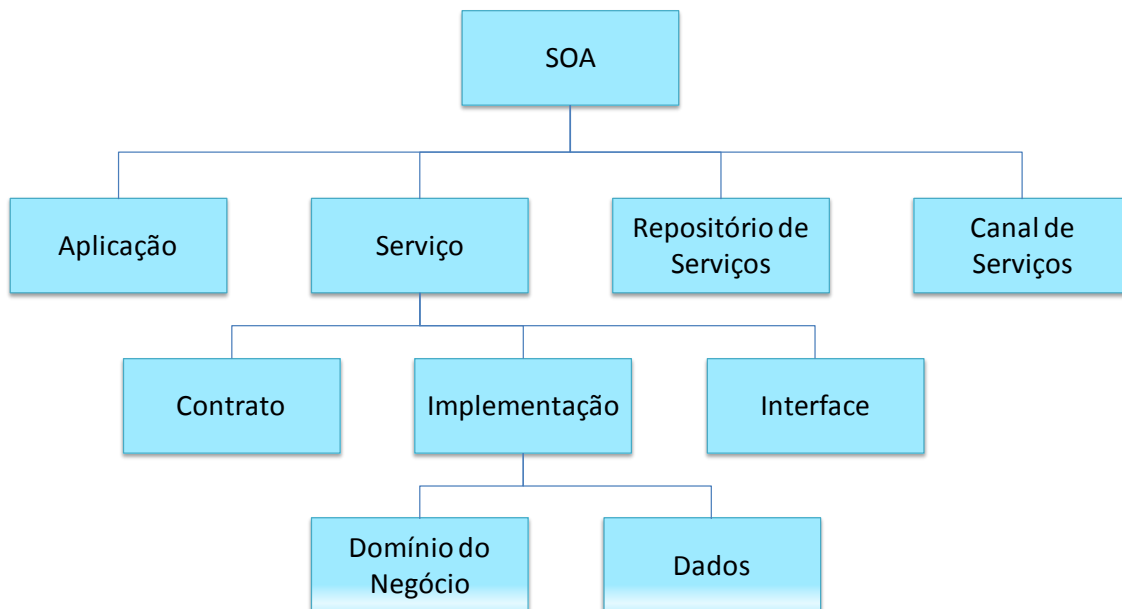


Figura 3.4 Elementos de SOA, adaptado de (KRAFZIG *et al.*, 2005).

Na camada superior, se encontram a aplicação, o repositório de serviços, o canal de serviços e o serviço em si. A aplicação não difere das aplicações finais de outras arquiteturas em camadas, no entanto é esperado que as aplicações em SOA tenham

maior proximidade com os serviços. O repositório de serviços é o conjunto de serviços disponíveis para a aplicação, da mesma forma o canal de serviços apresenta o modo como os serviços são acessados ou consumidos. Para o serviço em si será feita uma descrição mais detalhada a seguir.

Um serviço é um componente de software que encapsula as funcionalidades em um conceito de alto nível, consiste de várias partes como pode ser visto na Figura 3.5 (KRAFZIG *et al.*, 2005).

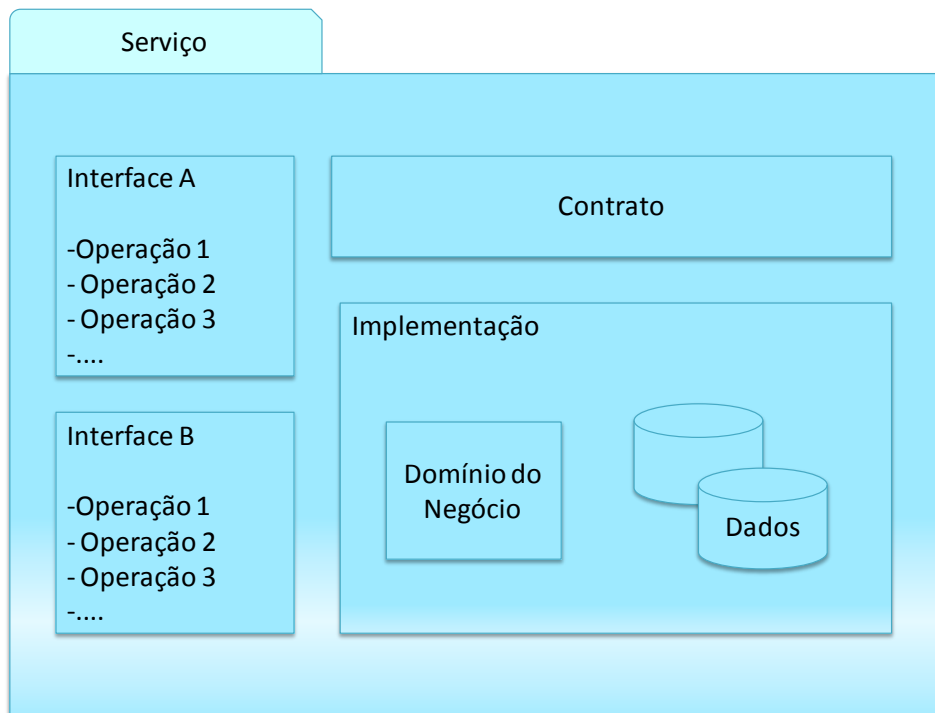


Figura 3.5 Componentes de um serviço, adaptado de (KRAFZIG *et al.*, 2005).

Contrato

O contrato existente em um serviço dá a especificação do serviço, definindo o propósito, a funcionalidade, as restrições e o uso. Não há uma padronização para definição deste contrato, porém há muitos benefícios em uma formalização deste contrato que é normalmente feito em IDL (*Interface Description Language*) ou WSDL (*Web Service Description Language*), dentre os benefícios está a de prover maior abstração e independência em relação à tecnologia utilizada, inclusive sobre a linguagem de programação escolhida. O contrato pode definir a semântica utilizada e

os parâmetros para o que serviço aceite a requisição e responda como esperado (KRAFZIG *et al.*, 2005).

Interface

A interface apresenta aos clientes do serviço a conexão para o consumo através de uma rede. A descrição da interface é parte do contrato e a implementação física é feita através de conexões entre o provedor e o requisitante (BOOTH, 2004).

Implementação

É a implementação física que permite acesso ao domínio do negócio e aos dados dando realização ao que está no contrato, a implementação é composta de códigos, configuração de dados e dados em si (BOOTH, 2004).

Domínio de Negócio

O domínio do negócio está encapsulado no serviço e é parte da implementação sendo disponibilizado pelas interfaces (BOOTH, 2004).

Dados

Os serviços podem, ou não, incluir dados em seu encapsulamento. Quando isso ocorre o serviço é centrado em dados, que pode ser útil em alguns casos, mas não quando se trata de dados distribuídos (BOOTH, 2004).

A arquitetura orientada a serviço, dentro do campo de engenharia de software, é uma continuação do desenvolvimento de técnicas de computação distribuída.

3.5 Serviços Web

Segundo Booth (2004), serviços Web são definidos como um sistema de software que possibilita a interação máquina a máquina em uma rede e tem uma interface descrita em um formato passível de ser interpretado automaticamente, especificamente WSDL (*Web Service Description Language*). Outros sistemas podem interagir com um serviço Web por meio de mensagens SOAP (*Single Object Access Protocol*). Normalmente,

utilizam o protocolo HTTP (*Hypertext Transfer Protocol*) com dados em XML (*eXtensible Markup Language*) em conjunto com outras normas relacionadas.

De modo concreto, a implementação de um serviço Web é feita por meio de trocas de mensagens entre um provedor do serviço e um requisitante solicitador do serviço. Estas mensagens podem ser desde dados simples até objetos complexos. Um requisitante é uma entidade que deseja fazer uso da funcionalidade disponibilizada pelo provedor do serviço, para tanto envia uma mensagem ao agente e recebe outra mensagem como resposta. Como mensagem entende-se desde um dado específico até um objeto complexo (BOYD *et al.*, 2008).

Para que esta troca de mensagens aconteça de modo pleno, tanto o requisitante quanto o provedor devem primeiro acordar sobre a semântica e o mecanismo de troca de mensagens, estabelecendo um acordo, feito com base na descrição do serviço que determina o formato da mensagem, o tipo de dados, protocolos de comunicação e mecanismos de serialização, também fica estabelecido o local na rede onde o serviço estará disponível. Na descrição do serviço também está estabelecido em que local da rede o serviço pode ser acessado (BOOTH, 2004).

Dentro do contexto de um serviço Web, a semântica compreende a expectativa sobre o comportamento do serviço, em particular na resposta de mensagens enviadas. Para um serviço Web isolado a questão semântica não é tão importante, porém em uma arquitetura baseada em serviços para um sistema distribuído, a semântica do serviço Web é essencial para a adequação do pedido à resposta correta. Enquanto a descrição do serviço representa um contrato entre partes, a semântica representa o contrato que governa o sentido e propósito possibilitando a automação do uso dos serviços (LITWIN *et al.*, 1990).

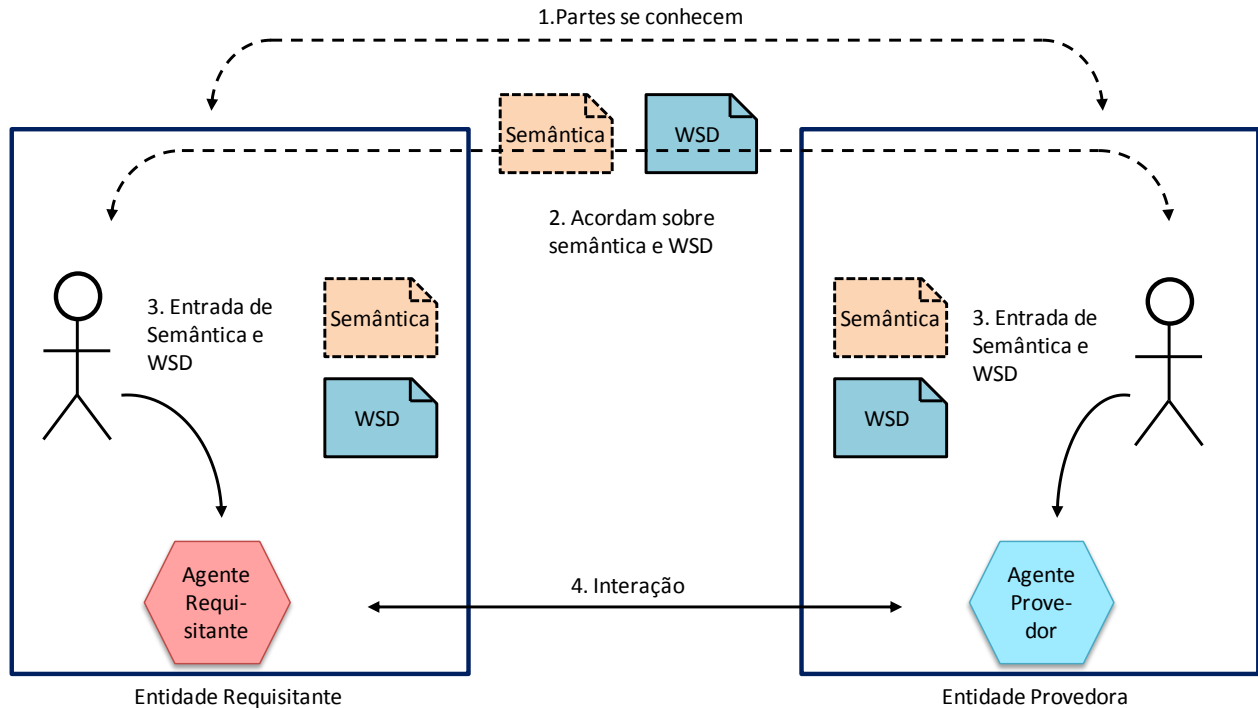


Figura 3.6 Processo geral de funcionamento de serviço Web, adaptado de (BOOTH, 2004)

De maneira geral o uso de um serviço Web ocorre do modo apresentado na Figura 3.6, seguindo os passos, que podem ou não ser automatizados:

- o requisitante e o provedor se conhecem;
- o requisitante e o provedor entram em acordo sobre a semântica que irá apoiar a interação;
- a descrição do serviço e a semântica são executadas;
- os agentes do requisitante e do provedor trocam mensagens, enviando pedidos, processando-os e recebendo respostas.

A interação entre os agentes mostrada na Figura 3.6, item 4 acontece através de troca de mensagens, que obedecem a dois padrões distintos (JOSUTTIS, 2008), o primeiro no qual para cada mensagem de requisição é enviada uma resposta e o segundo em que é possível combinar mensagens de requisição e de resposta, de tal forma que o consumidor do serviço não fica bloqueado durante o tempo entre requisição e resposta.

As mensagens podem ser escritas em XML, sendo este o padrão mais utilizado para a comunicação entre agentes já que, pelo fato de possibilitar várias alternativas de agregar funcionalidades, sendo capaz de descrever não somente os dados, mas também metadados e instruções para integração determinou um grande avanço para a integração de diferentes bancos de dados (FONG, 2006).

XML é uma linguagem de marcação desenvolvida pelo consórcio responsável pela Web, W3C - *World Wide Web Consortium*, para solucionar limitações da linguagem de marcação de hipertexto - HTML (*Hypertext Markup Language*). Uma vez que a linguagem HTML se apresenta como um formato fixo, a falta de mecanismo de flexibilidade motivou a procura por outras linguagens de marcação. Diferentemente da HTML, o XML permite aos autores a definição dos seus próprios elementos para a representação de esquemas de banco de dados ou hierarquias orientadas a objeto.

Para o acesso às mensagens é utilizado o protocolo SOAP, que é um protocolo simples para a troca de informações em um ambiente distribuído e descentralizado que habilita a comunicação entre dois processos, desconsiderando o hardware e a plataforma em que estão rodando (GUDGIN *et al.*, 2006). Com a utilização da tecnologia XML, o SOAP define uma estrutura extensível de troca de mensagens, padronizando seu formato e estrutura (PIRNAU, 2009).

Podem ser citados como objetivos do SOAP a simplicidade e a extensibilidade. Para isso, as características encontradas em sistemas distribuídos são omitidas, tais como questões relacionadas à confiabilidade, segurança, correlação e testes nos diferentes padrões de troca de mensagens, entre outras (PIRNAU, 2009).

3.6 WSDL

A linguagem para descrição de serviços Web (*WSDL para a sigla em inglês de Web Services Description Language*) define um modelo para que serviços Web possam ser descritos de modo a separar a descrição do serviço em si, de modo abstrato é um modelo que informa como e onde o serviço pode ser consumido e dá o modo como as

mensagens podem ser trocadas, questões relativas à segurança, estilo das operações e outras extensões (CHINNICI *et al.*, 2007).

O serviço Web, é descrito em dois estágios sendo um concreto e outro abstrato, para isso é utilizada uma linguagem de descrição do serviço Web - WSDL. As mensagens trocadas entre o requisitante e o provedor do serviço são definidas em modo abstrato através dos termos nos quais essas mensagens são descritas independentemente da conexão feita para a comunicação. O estágio concreto define a maneira como o serviço é implementado e como é feito o transporte das mensagens e também o acesso para o serviço em questão (CHINNICI *et al.*, 2007).

Possíveis requisitantes de um serviço Web podem, pela sua descrição, encontrar os meios de comunicação com o serviço sem que seja necessário o conhecimento do funcionamento interno deste. Todas as interações possíveis com um serviço Web estão descritas na sua especificação mesmo que nem todas sejam necessariamente utilizadas pelo requisitante (LEE, H.-J. *et al.*, 2009).

3.7 Considerações Finais do Capítulo

As tecnologias disponíveis para o gerenciamento das informações durante o ciclo de vida do produto permitem o uso de bancos de dados federativos dentro de uma arquitetura orientada a serviços, com isso é possível disponibilizar os dados da cadeia CAD/CAM/CNC em arquivos na Web e recuperar estes dados com o uso de serviços Web. Uma solução baseada nessas tecnologias permite a persistência das características de colaboração e distribuição de um sistema produtivo.

No capítulo seguinte é definido o objetivo desta tese e sua contribuição delineando o foco de atuação dentro do sistema produtivo.

4 Contribuição do Trabalho

Este trabalho pretende demonstrar que o ciclo de desenvolvimento do produto pode ser organizado dentro de uma arquitetura orientada a serviços em um modelo dividido em três domínios representados por três eixos ortogonais entre si, a saber: o eixo dos recursos; o eixo do produto e o eixo dos processos. Desta forma a manufatura, atividade colaborativa e distribuída, a despeito de sua aparente complexidade, pode ser reduzida a problemas simples que permitem o gerenciamento de um grande volume de informações. Esta proposta lida com a grande variedade de informações por intermédio de um gerenciamento federativo dos dados, utilizando as tecnologias de serviços Web e troca de mensagens envelopadas em protocolos de acesso a simples objetos.

4.1 Objetivos

O objetivo desta tese é propor um modelo de organização de um sistema de manufatura estabelecido sobre relações causais lineares e que esta organização seja gerenciada em uma arquitetura orientada a serviços, especialmente no que concerne à integração do desenvolvimento do produto.

Considerando que um sistema de manufatura lida com um volume de informações heterogêneas, e que estas informações estão distribuídas em diversos ambientes, é feita a proposta de uma divisão do sistema de manufatura em três domínios e com isso o objetivo pode ser desmembrado em duas etapas consecutivas: propor o modelo do sistema de manufatura; e estabelecer este modelo em uma arquitetura orientada a serviços.

Para atingir este objetivo é necessário o cumprimento dos seguintes requisitos:

- Desenvolvimento de aplicativos para cada um dos domínios da manufatura;
- Desenvolvimento de serviços Web para cada um dos domínios de manufatura;
- Desenvolvimento de um aplicativo para agregação dos dados dos modelos parciais para o gerenciamento do modelo central.

4.2 Originalidade

A organização de um sistema de manufatura com vistas à integração dos seus constituintes é um caminho para atingir a possibilidade de ganhos em produtividade e flexibilidade, sendo objetivo de várias pesquisas. Destas pesquisas notam-se duas frentes principais, uma que busca a integração plena de todos os constituintes do sistema de manufatura (sistemas integrados), mas que perde em autonomia e outra, que compreendendo a natureza distribuída desse sistema, busca soluções distribuídas com foco nas propostas advindas da teoria da complexidade, esta apresentando deficiências na integração (sistemas distribuídos). A originalidade deste trabalho está na proposta de um modelo intermediário construído sobre uma arquitetura orientada a serviços, mantendo tanto a autonomia quanto a integração de forma parcial, e tendo ganhos na flexibilidade do sistema.

Ao se modelar um sistema de manufatura dentro do contexto de serviços, todas as atividades passam a ser vistas como serviços e os atores do sistema como fornecedores ou clientes destes serviços. Desta forma um sistema com muitos componentes e, portanto, uma grande quantidade de informações, pode ser administrado sem que se perca a celeridade do processo e nem a flexibilidade, características essenciais para um sistema de manufatura.

4.3 Método de Pesquisa

O problema prático que motiva a realização deste trabalho é a dificuldade em organizar um sistema produtivo que utilize os benefícios atuais da tecnologia da informação, sendo simples o suficiente para sua implantação e entendimento e que também seja capaz de lidar com um grande volume de dados.

Sistemas comerciais oferecem soluções centralizadas que garantem a integração dos componentes, mas obrigam o uso de uma plataforma única que limita a flexibilidade do sistema. Pesquisas estão sendo feitas com soluções baseadas na teoria da complexidade e apresentam dificuldades na integração de todos os seus componentes. Enquanto que este trabalho propõe uma solução intermediária entre a integração plena

e a autonomia plena, apresentando ganhos na flexibilidade do sistema e na possibilidade de trabalhar também com sistemas legados.

Foram escolhidas ferramentas de desenvolvimento de uso corrente, compatíveis com softwares comerciais de uso em larga escala, especialmente no que concerne a sistemas CAx.

4.4 Contextualização

Os desenvolvimentos de produtos e de processos estão inseridos dentro das atividades do sistema produtivo e são atividades básicas da indústria de transformação. As atividades do desenvolvimento de produto podem ser realizadas de modo mais racionalizado ou de modo mais informal. Produtos simples podem ser desenvolvidos de maneira mais informal, o oposto exige procedimentos determinados, executados por profissionais especializados dentro de grupos organizados (HANSEN e KRISTENSEN, 2011).

Para a produção em escala comercial de um produto são necessários vários recursos e uma oportunidade de mercado. Neste sentido, a produção é consequência de um processo que transforma dados sobre o mercado em oportunidade e possibilidades técnicas em recursos.

As informações são criadas, compartilhadas e transferidas fazendo uso das tecnologias que servem de suporte ao processo de desenvolvimento, caracterizando-o, desta forma, como um processo abstrato. O processo de desenvolvimento, por esta razão pode ser caracterizado como baseado no fluxo dessas informações e traz como consequência o fato de poder ser facilmente deslocado espacialmente, permitindo um trabalho distribuído (ZIMMERMANN, 2005).

Durante o desenvolvimento do produto são incorridos, relativamente, poucos custos, entretanto é nessa fase que é comprometida a maior parcela dos custos, como pode ser visto na Figura 4.1, podendo chegar até a 85% do custo do produto final (ROZENFELD *et al.*, 2006). No caso do setor automotivo este valor fica em torno de 80% (EIGNER, 2008).

No início do desenvolvimento do produto as soluções construtivas e as especificações do produto são definidas, assim como as tecnologias e os materiais que serão utilizados, bem como os processos de fabricação. Mesmo que soluções alternativas sejam consideradas no decorrer do processo as soluções centrais são determinadas nas fases iniciais. Estas definições são feitas com base em atividades de pesquisa de mercado para prospecção e detecção de necessidades e desejos dos clientes, definindo: o nicho de mercado; o público-alvo; o grau de qualidade; e em certa maneira até o preço de venda. As etapas iniciais, posto que determinem em grande parte o custo de produção, ocorrem no momento em que há pouco conhecimento sobre o produto.

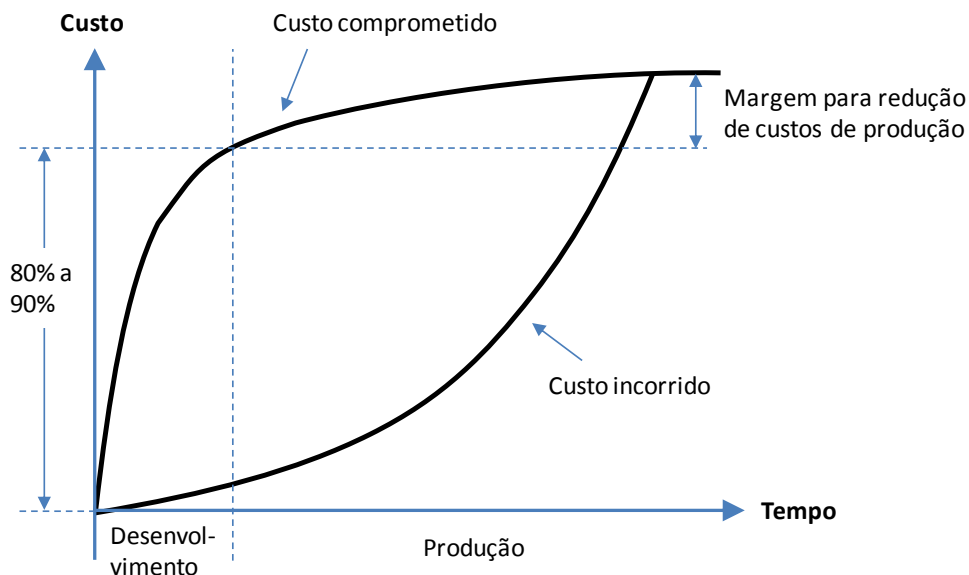


Figura 4.1 Curva de comprometimento do custo do produto (ROZENFELD et al., 2006).

Durante a produção a maior parte dos custos do produto já está definida, sendo diminuída a possibilidade de redução.

A divisão do custo do produto considerada a contabilidade de custos, como apresentado na Figura 4.2 mostra a diferença entre o custo de cada setor e sua atividade, em comparação com o quanto este setor impacta no custo do produto. Enquanto o setor de projeto, com suas atividades, representa apenas 5% do custo do produto, esta atividade impacta na determinação de 70% do custo total do produto. Como visto nos parágrafos anteriores esta atividade ocorre justamente nas fases

iniciais em que se tem pouco conhecimento do produto, e é nesse momento em que se tomam as decisões que irão impactar mais fortemente no custo do produto.

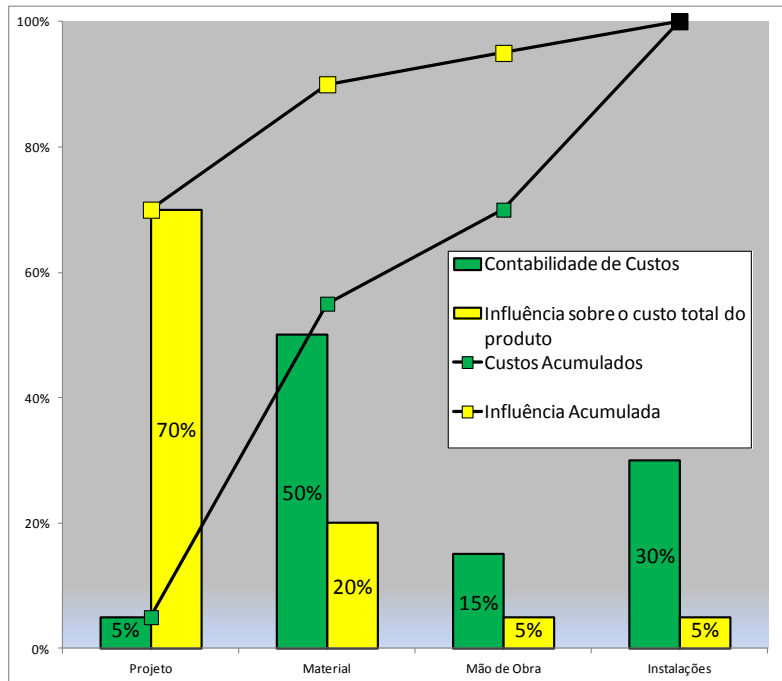


Figura 4.2 Relação entre custo provocado e custo do componente, adaptado de (SMITH e REINERTSEN, 1998).

Por influir na maior determinação do custo do produto o projeto impacta diretamente o desempenho da organização e é natural que se busquem alternativas para redução deste custo sendo isso possível por meio da redução do custo provocado e da redução do custo componente em si.

A redução do custo final do produto pode ser efetivada pela economia de escala e de escopo, especialmente nas indústrias de transformação nas quais o custo da unidade produtiva decresce exponencialmente à razão da quantidade produzida (DIAS, 2003).

Para que se obtenham ganhos com a economia de escala pode haver a tendência de se trabalhar dentro de uma racionalidade rígida, sem espaço para a flexibilidade, que leva a uma redução na diversidade dos produtos, maior padronização e aumento do ciclo de vida do produto. Levado a um limite, esta estratégia faz com haja a redução do impacto do custo devido ao desenvolvimento do produto e também seja reduzida a

necessidade de inovação, enfim, menor flexibilidade e menos inovação, o que é totalmente o oposto do que se exige atualmente (SZWARCFITER e DALCOL).

A redução de escopo também pode ser realizada na etapa de desenvolvimento do produto visto que o conhecimento utilizado nesta atividade não está ligado a um único produto, o conhecimento envolvido no projeto de um motor pode ser utilizado para desenvolver vários motores a serem utilizados em diversos produtos (DIAS, 2003).

Dentro desta perspectiva é natural que a estrutura empresarial se encaminhe para uma rígida organização funcional e especialização, mormente para produtos com vários componentes que atuam de maneira integrada e a especialização é uma maneira de sedimentar o conhecimento e de o ampliar sobre as tecnologias utilizadas.

Se a atividade de desenvolvimento do produto impacta diretamente o custo provocado (ou comprometido), o tempo despendido neste desenvolvimento tem implicações mais profundas, isto porque a duração desta etapa no ciclo de desenvolvimento do produto não impacta somente os custos diretos e de mão de obra.

Mais importante que os custos diretos e de mão de obra, o tempo de desenvolvimento do produto acarreta um risco comercial já que a necessidade do consumidor é dinâmica. Quanto maior o tempo entre a captação da percepção do desejo do consumidor e o lançamento do produto ao mercado maior o risco deste produto não atender às necessidades identificadas, supondo que corretamente captadas (ZARIFIAN, 2001).

Porém, além dos custos diretos e do risco comercial é ainda mais importante o custo de oportunidade. Segundo Midler (2005) *“tempo de projeto não é um dado da natureza, mas uma variável estratégica”*, porque permite o lançamento do produto ao mercado antes da concorrência, obtendo um ganho de mercado que só pode acontecer no momento do lançamento. Claro que, neste sentido, a pressão sobre o tempo de desenvolvimento está diretamente ligada ao nível concorrencial em que se opera, mesmo que não seja possível lançar ao mercado um produto inovador, interessa reduzir o tempo para reagir ao produto lançado pela concorrência.

Os conceitos aqui apresentados se mostram antagônicos, de um lado a busca por reduzir o custo e o tempo de lançamento do produto ao mercado, o que leva a uma racionalidade sistemática no desenvolvimento do produto, que por sua vez, afeta negativamente a flexibilidade e inovação. De outro lado existe a busca por produtos personalizados e inovadores, o que exige flexibilidade e inovação, requisitos não atendidos através da racionalidade sistemática de desenvolvimento do produto. O desafio, típico de um sistema flexível de manufatura, está em atender a estas duas exigências opostas.

Na tentativa de superar este desafio existem várias propostas de organização do sistema de manufatura, com a ideia central de disponibilizar o maior conjunto de informações a todos componentes do sistema.

Uma forma de se conseguir isso é pela integração total de todas as tecnologias envolvidas, com o conjunto pleno das informações armazenadas em um sistema centralizado, esta solução ganha em integração, porém perde na autonomia de cada componente; também há perdas na flexibilidade já que todas as modificações devem ser feitas considerando o conjunto do sistema.

Outra possibilidade é manter cada componente de forma independente e que as comunicações sejam feitas de modo externo ao sistema completo. Diferentemente da solução anterior, esta mantém a plena autonomia dos sistemas, mas há dificuldades na integração de todos os dados. A flexibilidade também é prejudicada já que as modificações de um componente tem que considerar as interfaces de integração com o sistema completo.

A solução proposta por este trabalho mantém parcialmente a autonomia e a integração ao estruturar o sistema de manufatura com banco de dados distribuídos, o que garante o máximo de flexibilidade. Isto é possível por meio de uma divisão do sistema em três domínios e por usar como tecnologia de suporte uma arquitetura orientada a serviços. Mesmo o trabalho de Barton *et al.* (2001) que questiona avaliação do projeto no custo do produto, recomenda uma abordagem que envolva mais atores e, portanto agrega mais disponibilização de informação para a melhoria do projeto do produto.

5 Desenvolvimento do modelo

A aplicação aqui demonstrada é baseada em uma aplicação para produção de um cilindro pneumático como ocorre na *Prozesslernfabrik* do *Center für industrielle Produktivität* pertencente à *Technische Universität Darmstadt* que é parceira do projeto *FEDMAN*, que por sua vez integra ao programa Bragecrim (*Brazilian-German Collaborative Research Initiative on Manufacturing Technology*), como apresentado em Anderl *et al.* (2011). Esta escolha se deveu ao fato deste produto permitir as considerações de processos, recursos e de produto, necessárias à proposta apresentada.

A *Prozesslernfabrik* é uma iniciativa do *Institut für Produktionsmanagement, Technologie und Werkzeugmaschinen* (<http://www.prozesslernfabrik.de/>) com o objetivo de criar um ambiente real de produção para aplicação educacional.

O modelo de desenvolvimento do produto, neste cenário, é um modelo baseado em processos dentro de uma abordagem de desenvolvimento integrado do produto. Segundo Rozenfeld *et al.* (2006), uma das características propostas pelas abordagens mais recentes é o conceito de gerenciamento do ciclo de vida do produto de modo a ampliar o escopo do processo e também permitir a integração entre projetos.

No modelo proposto são definidos processos contidos no macro processo desenvolvimento do produto. Cada um dos processos tem, para o seu funcionamento, sistemas de tecnologia de informação dedicados que podem estar distribuídos fisicamente. As informações geradas por cada processo ficam armazenadas em bancos de dados distribuídos. A integração do processo de desenvolvimento passa pela integração destas informações, como mostrado na Figura 5.1, com uma visão reduzida do ciclo de vida do produto como proposto por Eigner e Stelzer (2009) e discutido em 2.3.1. Cada ferramenta específica gera os dados autonomamente que são acessados através de uma camada federativa externa ao sistema.

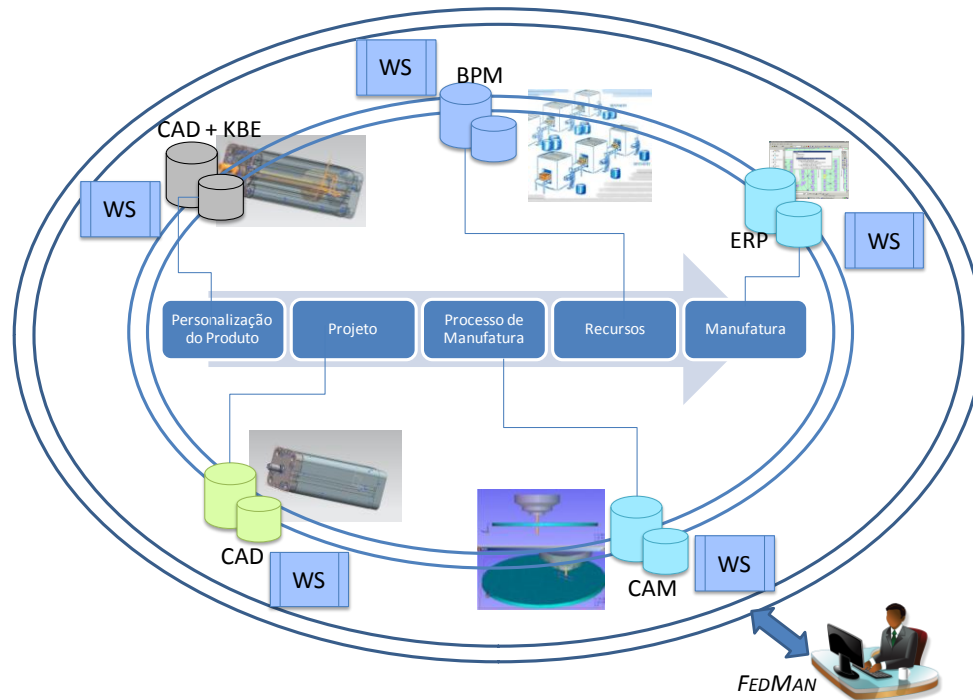


Figura 5.1 Cenário proposto.

O fluxo de informações neste cenário se inicia com a consolidação do conceito do produto construído em um sistema CAD parametrizado com a inserção do conhecimento utilizando as técnicas de engenharia baseada no conhecimento. Nesta base é inserida a informação do mercado, advinda de um pedido com as especificações técnicas do solicitante. O produto obtido nesta etapa do desenvolvimento, já consideradas as suas versões e variantes, é trabalhado para a definição dos processos de usinagem com a utilização de um sistema CAM. A partir deste ponto são definidos os recursos necessários para a produção e montagem do produto. Estes recursos, armazenados em um sistema BPM são então alocados para a produção. Nesse trabalho foram considerados os dados relativos ao produto em si, utilizados no sistema CAD; os dados de processo, considerados no sistema CAM e os dados de recursos que ficam alocados também no sistema CAM. A cada etapa o responsável pelo desenvolvimento deve enviar os respectivos dados a partir do momento em que estejam maduros o suficiente para a utilização nas outras etapas, fazendo uso dos conceitos de engenharia simultânea.

Com estas informações consolidadas é possível vincular: os dados do produto (o cilindro com as especificações do cliente feitas no sistema CAD); os dados de processo

(os processos de usinagem definidos no sistema CAM); e os dados dos recursos para produção (os equipamentos necessários para produção do cilindro considerados pelos sistemas BPM e ERP), ver Figura 5.2.

Para que todo este processo seja possível em um ambiente altamente distribuído, todos os dados devem estar disponíveis para todos os atores a todo o tempo, portanto foi utilizado o conceito de banco de dados distribuídos, permitindo a heterogeneidade e preservando a autonomia de cada um dos sistemas. O processo de diálogo entre os atores deste processo e dos sistemas por eles utilizados foi feito com o uso de serviços Web e utilizando sistemas neutros.

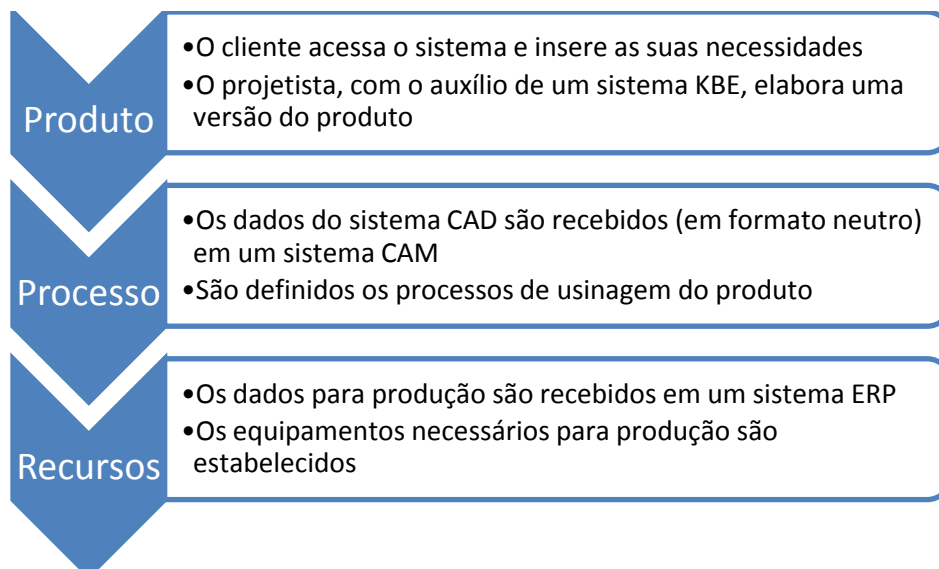


Figura 5.2 Processo simplificado de desenvolvimento do produto

Com a integração dos dados de Produto x Processos x Recursos é possível determinar a viabilidade e uma estimativa de custo e tempo de produção. Os dados também servem para a simulação da produção em um sistema de fábrica digital.

A integração deste processo geral de desenvolvimento do produto a um sistema de gerenciamento do ciclo de vida do produto consolida o controle sobre o sistema de maneira geral.

Desde que os serviços Web disponibilizam o acesso aos bancos de dados distribuídos há a necessidade de coletar os dados disponibilizados e congregá-los de modo a se

fazer a interação entre dados de produto, processos e recursos, fechando o ciclo de desenvolvimento do produto.

Um aplicativo possível para o recolhimento destes dados é do tipo navegador que acesse os serviços disponibilizados e faça vínculos entre eles, também é desejável que este aplicativo possa atuar nos bancos de dados distribuídos, porém para que isso ocorra com a preservação da autonomia - característica intrínseca à heterogeneidade - o aplicativo deve fazer esta atuação por meio dos serviços e não diretamente.

A implementação dos aplicativos e dos serviços sustentados pela base teórica aqui apresentada é mostrada no capítulo seguinte.

6 Desenvolvimento do Modelo

No escopo deste trabalho serão considerados aplicativos para integração das informações de um sistema produtivo. Sendo esse considerado como o conjunto de meios e processos utilizados para a transformação de matéria prima em um produto acabado, incluindo os processos intermediários. Um sistema assim pode abarcar um grande volume de dados, já que cada um dos processos internos gera e administra dados de maneira autônoma. O grande volume de dados leva a uma aparente complexidade do sistema, como demonstram os estudos apresentados em 2.5.

Somente a grande variedade e diversidade de informações existentes não implicam, necessariamente, em considerar um sistema como complexo. Em um sistema complexo é preciso que se apresente um estado no qual os objetos estejam ligados em função de um padrão dinâmico e que não se apresentem os pressupostos de causalidade; linearidade do tempo; ou de ordem. Assim, os problemas gerados pelo gerenciamento das informações da manufatura e do ciclo de vida do produto, mesmo que geradas de diferentes fontes, em diferentes formatos, em grande quantidade e em relações n-dimensionais, podem ser reduzidos a problemas simples. Os dados de manufatura podem ser decompostos, analisados e sintetizados mantendo uma ordem, o que nada mais é que o pressuposto pelo método científico.

Baseado em um modelo teórico de bancos de dados federativos foram desenvolvidos aplicativos vinculados a sistemas comerciais atuais e também serviços Web para os três domínios propostos. Para a demonstração do gerenciamento federativo e da integração das informações foi desenvolvido um aplicativo.

Para cada um dos domínios aqui estudados, produto, processos e recursos, foram criados serviços Web. A agregação destes serviços e a integração dos domínios podem ser demonstradas pelo aplicativo especialmente desenvolvido.

Os aplicativos para os sistemas CAD e CAM foram desenvolvidos utilizando-se a biblioteca de funções NXOpen, disponibilizada pela Siemens PLM para o acesso às funções dos sistemas auxiliados por computador deste fabricante.

Os serviços Web foram desenvolvidos utilizando a linguagem ASP.NET e as ferramentas da plataforma dotNet®, dentro do paradigma da orientação a objetos. Os serviços foram disponibilizados na Web através de um servidor de aplicativos fornecido pela Microsoft.

A implementação do protótipo para integração das visões de produto, processos e recursos através de serviços Web se deu através do desenvolvimento em camadas. Na camada mais interna foram desenvolvidos aplicativos dentro de sistemas CAD e CAM para obtenção de dados de produto e processo, respectivamente, e disponibilização destes dados na Web. Na camada de serviços foram desenvolvidos serviços Web para leitura e análise dos dados disponíveis na Web para responder às requisições do aplicativo que faz a papel de integrar todos os dados de um sistema de manufatura. Por fim, também foi desenvolvido um aplicativo capaz de requisitar aos serviços Web os dados de produto, processos e recursos (SCHÜTZER *et al.*, 2011a) (SCHÜTZER *et al.*, 2011b). Um esquema dos aplicativos internos, dos serviços Web e do aplicativo de integração estão apresentados na Figura 6.1

Todo o desenvolvimento foi feito com o uso da plataforma dotNET®, dentro do paradigma de orientação a objetos. Neste ambiente de desenvolvimento foram criados tanto os serviços Web, como os aplicativos internos aos sistemas CAD e CAM, bem como o aplicativo de integração. Os serviços foram desenvolvidos em ASP.NET e os aplicativos em VB.NET.

Para o protótipo não foi utilizado nenhum sistema de gerenciamento de banco de dados. Os dados foram descritos em um arquivo físico, utilizando a linguagem XML e modelados de acordo com a norma ISO 10303 - STEP, em consonância com a parte 28.

Os aplicativos e serviços Web desenvolvidos foram:

- Aplicativo para o NX CAD, disponibilização do conjunto montado;
- Aplicativo para o NX CAM, disponibilização das operações e ferramentas utilizadas;

- Serviço Web para análise e recuperação de dados do produto;
- Serviço Web para análise e recuperação de dados de processo;
- Serviço Web para análise e recuperação de dados de recursos;
- Aplicativo para integração dos dados;
- Aplicativo para o NX CAD, disponibilização de uma curva B-Spline;
- Aplicativo para o NX CAM, disponibilização de um arquivo CLData;
- Serviço Web para estimativa do tempo de usinagem;
- Serviço Web para definição de pontos para inspeção em superfícies esculpidas e entrega ao CAI.

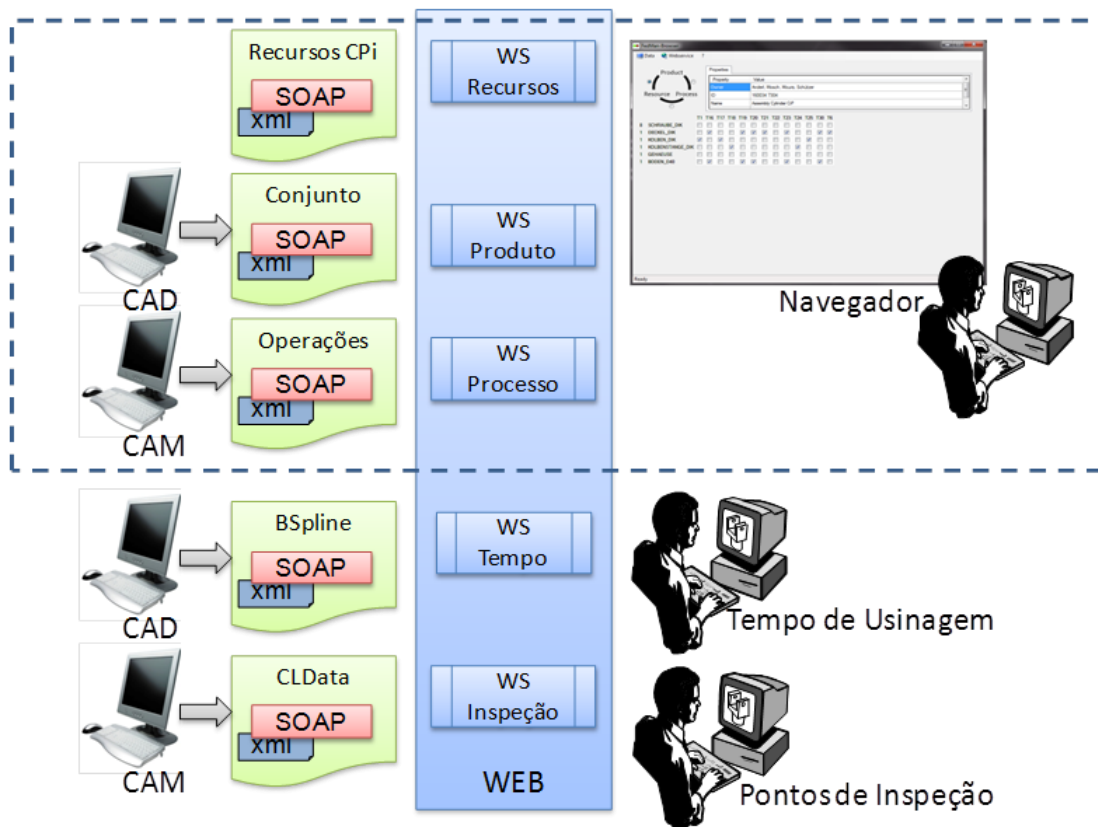


Figura 6.1 Esquema de funcionamento dos aplicativos e serviços desenvolvidos.

Os dois últimos aplicativos e serviços Web listados, estimativa de tempo de usinagem (SCHÜTZER *et al.*, 2012) e de pontos de medição em superfícies esculpidas, foram desenvolvidos de modo a responder a resultados diretos e, por essa razão, não estão vinculados ao aplicativo de integração.

6.1 Aplicativos Internos a Sistemas CAx

Sistemas de apoio baseados em computadores estão disponíveis para diversas áreas de atuação da engenharia, o desenvolvimento destes sistemas segue o ritmo do desenvolvimento geral da tecnologia da informação, especialmente para sistemas de apoio à manufatura e ao projeto são lançadas novas versões anualmente, tal velocidade de renovação faz com que seja comum a prática do consumidor final não atualizar todas as versões, esperando duas ou mais atualizações para implantação.

Estas atualizações atendem às demandas dos consumidores, que, no entanto, podem ter necessidades específicas não atendidas, para que seja possível um atendimento ainda mais especializado, os sistemas computacionais permitem que sejam desenvolvidos aplicativos que trabalhem em conjunto ou internamente ao sistema original conhecidos como aplicativos “*Add-on*”.

Do sistema original é possível iniciar a execução do aplicativo que pode ter acesso às funções internas do sistema, utilizando a mesma interface com o usuário de modo a tornar o uso do aplicativo bastante amigável. Para a demonstração dos conceitos propostos nesta tese foram desenvolvidos aplicativos para o gerenciamento de versões do produto e de manufatura.

Como o sistema utilizado para a demonstração tem os módulos de CAD e CAM integrados foi desenvolvida uma interface comum para os dois sistemas, que incluiu também as funcionalidades de estimativa do tempo de manufatura e de determinação dos pontos de medição. Na barra de títulos do programa foi incluída a funcionalidade do *FEDMAN*, que abre a possibilidade de gerenciamento de versões e conseqüentemente abre uma caixa de diálogo para as funções descritas, como pode ser visto na Figura 6.2

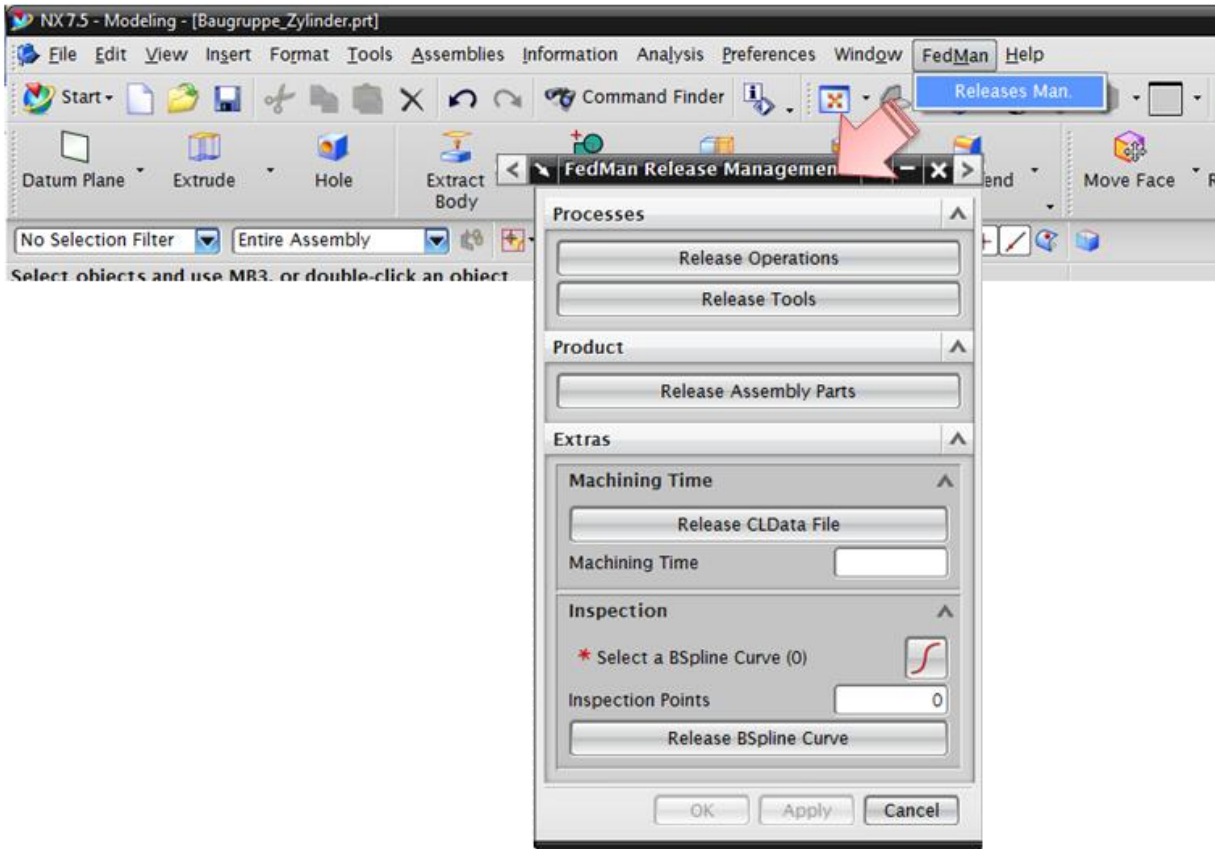


Figura 6.2 Caixa de diálogo do aplicativo.

Esta caixa de diálogo está dividida em três blocos: Processos (*Processes*); Produto (*Product*) e Extras.

Para a funcionalidade de produto é possível interagir com o serviço Web que disponibiliza o conjunto montado para um banco de dados federativo, incluindo o material de cada componente, sua localização, a quantidade de vezes que o componente aparece no conjunto e o material, o arquivo como um todo preserva a estrutura hierárquica do conjunto montado, tanto para o produto principal quanto para o produto variante.

Para a funcionalidade de processos é possível interagir com um serviço Web e disponibilizar as informações das ferramentas necessárias para o processo de manufatura e também disponibilizar os processos de usinagem para cada um dos componentes.

No subconjunto “Extras” estão os dois aplicativos: um que calcula pontos para o sistema de inspeção; outro que estima o tempo de manufatura.

O aplicativo de cálculo de pontos para inspeção está interno ao sistema CAD, inicialmente são disponibilizados para a Web os dados de um curva B-Spline, em seguida, este aplicativo requisita de um serviço Web o cálculo dos pontos sugeridos para inspeção.

Para a estimativa do tempo de usinagem o aplicativo envia para a Web o arquivo CLData (*Cutter Location Data*) contendo as informações sobre o caminho da ferramenta no processo de usinagem, ao mesmo tempo este aplicativo interage com um serviço Web que faz o cálculo estimado do tempo de usinagem.

Os dois serviços Web acessados através do sistema CAD/CAM também podem ser acessados externamente.

As funcionalidades de produto e de recursos serão apresentadas a seguir.

6.1.1 Aplicativo para sistema CAD

O aplicativo para o sistema CAD que fornece informação sobre o produto é acessível através do botão “*Release Assembly Parts*”, inserido no grupo “Product”. Este botão aciona um método para recolher no produto CAD os seus componentes, incluindo - neste estágio de desenvolvimento - a posição, o material e a quantidade de vezes que o componente aparece no conjunto montado. Estes são então compilados em um arquivo XML e enviados para um banco de dados distribuído, sendo esta etapa feita através de um serviço Web.

O sistema CAD mantém a estrutura do produto que tem um papel importante na integração dos sistemas, pois define a relação física entre módulos e componentes que constituem o produto (SCHUH *et al.*, 2003). Os vínculos entre os componentes do produto estão armazenados nesta estrutura, que pode conter conjuntos e subconjuntos de modo encadeado, a estrutura pode ser apresentada na forma de árvore ou lista indentada (LAURINDO e MESQUITA, 2000). No exemplo utilizado o eixo é composto

do eixo em si, do êmbolo e de fixadores. A Figura 6.3 apresenta a informação do produto em forma de árvore. A mesma informação é apresentada na Figura 6.4, com o desenho do produto à direita e à esquerda a lista indentada dos componentes do produto, O encadeamento dos componentes pode ter vários níveis, de tal forma que a obtenção dos dados dos componentes deve ir até o nível mais interno dos componentes para obter as informações relativas, sem que se percam as informações da estrutura do produto.

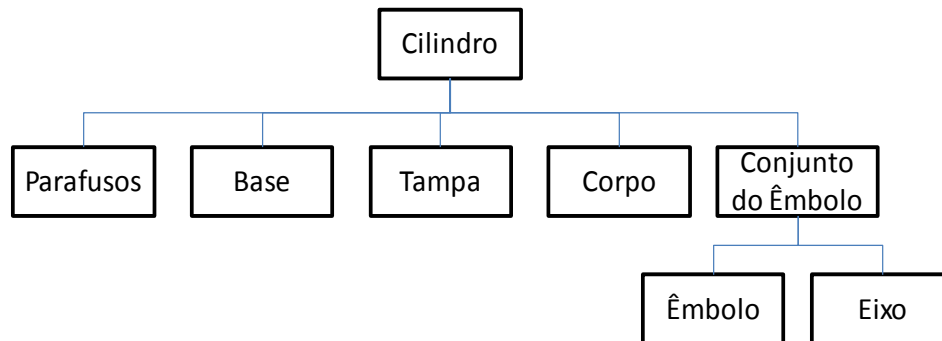


Figura 6.3 Árvore de construção.

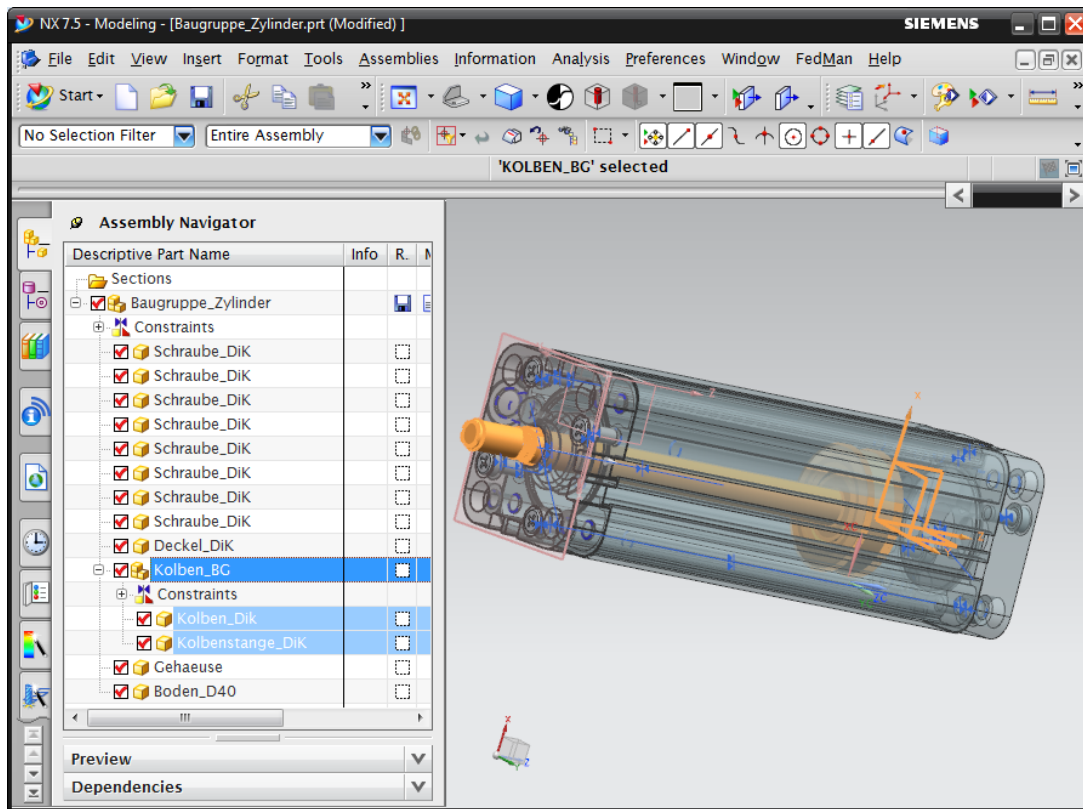


Figura 6.4 Conjunto montado e árvore de construção (subconjunto do eixo com eixo e êmbolo).

Para a garantia de que todos os níveis da árvore de construção sejam lidos, e que seja mantida a informação da estrutura do produto, foi elaborado um método recursivo, que obtém os dados dos componentes-filhos até o seu nível mais interno, como apresentado no estrutograma da Figura 6.5, enquanto houver componentes-filhos o método busca em um nível inferior, ao alcançar o nível mais interior são coletados os dados dos componentes que são inseridos em um objeto XML, o conjunto destes objetos é transferido a um arquivo XML através de métodos próprios da plataforma dotNet e um arquivo XML é criado e transferido para um banco de dados distribuído.

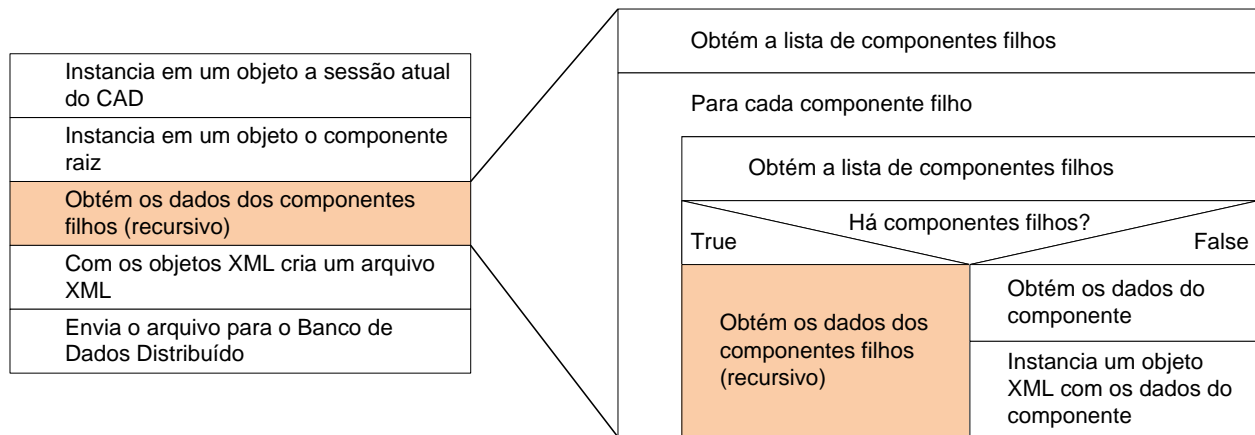


Figura 6.5 Estrutograma de leitura dos dados dos componentes

Os objetos XML são colocados em uma estrutura de dados do tipo lista e o arquivo gerado é baseado nestes objetos, ao final tem-se um arquivo XML estruturado com os componentes e seus atributos, como apresentado na Figura 6.6.

No arquivo mostrado na Figura 6.6 estão mostrados os valores para cada um dos componentes iniciados pela etiqueta “<Part>”, que contém os atributos: número; quantidade; tipo, localização e material.

Cada campo do arquivo é iniciado por uma etiqueta do tipo “<etiqueta>” e finalizado por outra etiqueta do tipo “</etiqueta>”, por ser extensível podem ser criadas tantas etiquetas para quantos campos foram necessários.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
-<Assembly Name="">
  -<PartsList>
    -<Part Number="0">
      <name>SCHRAUBE_DIK</name>
      <quantity>8</quantity>
      <type>NXOpen.Assemblies.Component</type>
      <locationX>2,09881368775334</locationX>
      <material>AISI_Steel_1005</material>
    </Part>
    -<Group Number="0">
      <name>KOLBEN_BG</name>
      <quantity>1</quantity>
      <type>NXOpen.Assemblies.Component</type>
      <locationX>2,09881368775334</locationX>
      <material>AISI_Steel_1005</material>
      -<Part Number="2">
        <name>KOLBEN_DIK</name>
        (...)
      </Part>
      -<Part Number="3">
        <name>KOLBENSTANGE_DIK</name>
        (...)
      </Part>
    </Group>
```

Figura 6.6 Excerto de arquivo XML com dados dos produto.

Após a transferência do arquivo para o banco de dados é informado ao usuário o resultado da operação como pode ser visto na Figura 6.7. Como o arquivo usa a linguagem de marcação extensível (XML), outros atributos podem ser incluídos sem prejuízo do formato atual, o que, em parte, garante a utilização mesmo com o uso de sistema legados, já que informações obsoletas podem ser incluídas sem alteração da estrutura do arquivo.

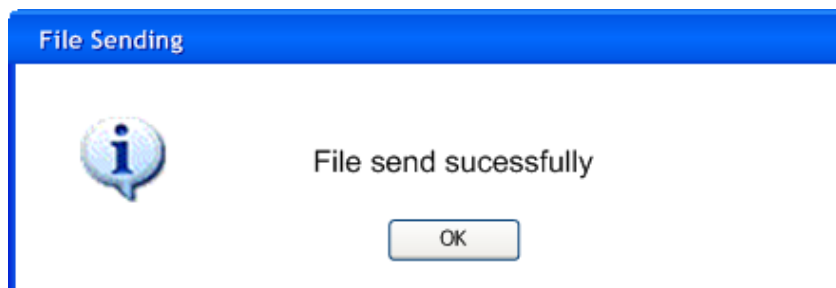


Figura 6.7 Caixa de diálogo com informação do envio do arquivo XML.

6.1.2 Aplicativo para sistema CAM - Operações

As informações do sistema de manufatura auxiliada por computador encontram-se dentro do escopo de informações relativas a processos, aqui considerados como processos de fabricação, como: furação; rosqueamento; faceamento; etc. Também podem ser obtidas informações relativas a recursos, já que o sistema CAM contém os equipamentos e as ferramentas utilizadas no processo de manufatura.

Da mesma forma que o aplicativo interno do sistema CAD, este aplicativo é acessível através do botão “*Release Operations*” dentro do grupo “*Resources*”. Ao acionar este botão é feita uma varredura de todos os processos de usinagem previstos para esta peça. As operações são então compiladas em um arquivo XML que por sua vez é enviado para um banco de dados distribuído. A etapa de transmissão para o banco de dados é feita por meio de um serviço Web.

A programação de usinagem é montada em um sistema hierárquico como pode ser visto na Figura 6.8.

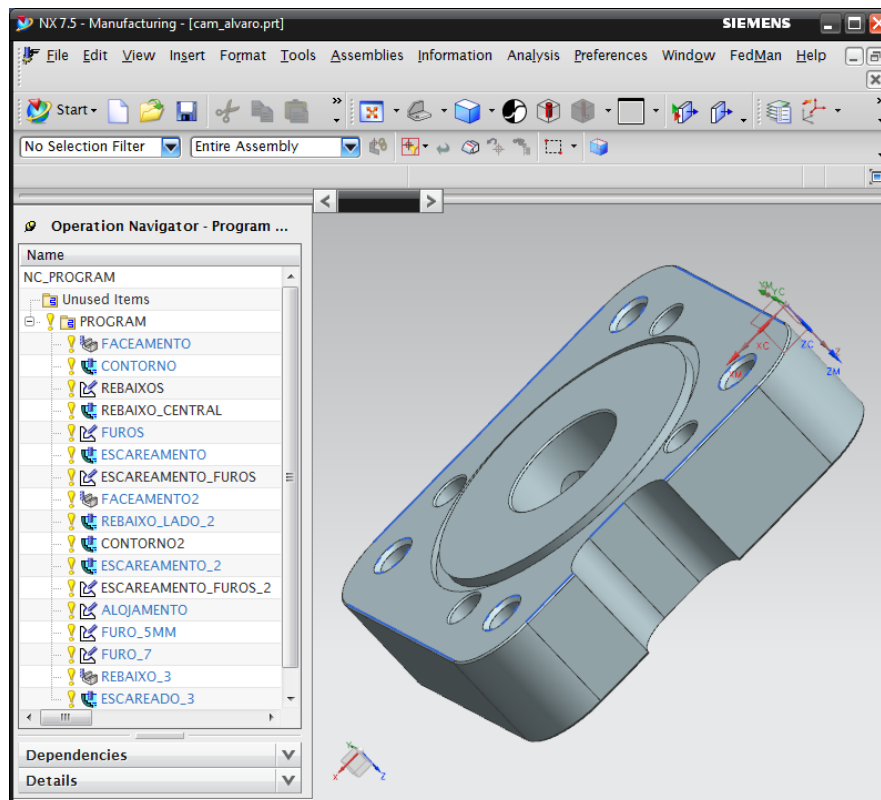


Figura 6.8 Sistema CAM com navegador de operações.

No exemplo apresentado a usinagem está dividida em três partes: a face superior; a face inferior e a face lateral. As operações do lado superior, mostrado na Figura 6.9 (a) e indicado como “REFERENCIA_1”, são o faceamento; o contorno da peça; o rebaixo externo; o rebaixo circular central; os furos e seus escareamentos. A face inferior indicada na Figura 6.9 (c) e sob o grupo “REFERENCIA_2”, contém um faceamento, o rebaixo lateral, o contorno, os escareamentos dos furos e o alojamento do anel de vedação feito com uma ferramenta especial. Por fim, a face lateral que é perpendicular à face superior, está indicada como “REFERENCIA_3” e pode ser vista na Figura 6.9 (b). Nesta face são usinados: o furo interno, o furo externo com rebaixo e o escareamento.

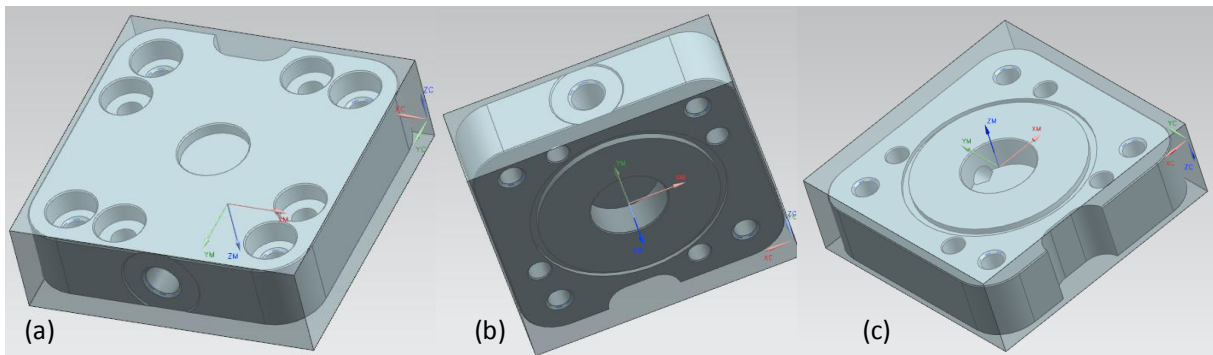


Figura 6.9. Superfícies a serem usinadas na peça exemplo.

Cada operação está vinculada a uma ferramenta específica, o sistema CAM fornece as informações relativas à operação sendo que uma das opções pode abrir a janela de diálogo que mostra os dados relativos à ferramenta, estas duas janelas estão apresentadas na Figura 6.10 para a operação de faceamento, com destaque especial para os dados que foram escolhidos para compor o arquivo a ser transferido para o banco de dados distribuído.

O sistema CAM armazena as operações dentro de um objeto do tipo coleção de objetos, que são as informações de cada operação e suas características. O aplicativo desenvolvido obtém o acesso a esse objeto e busca as informações fazendo uma varredura por todas as operações nele contidas, de acordo com o estrutograma apresentado na Figura 6.11.

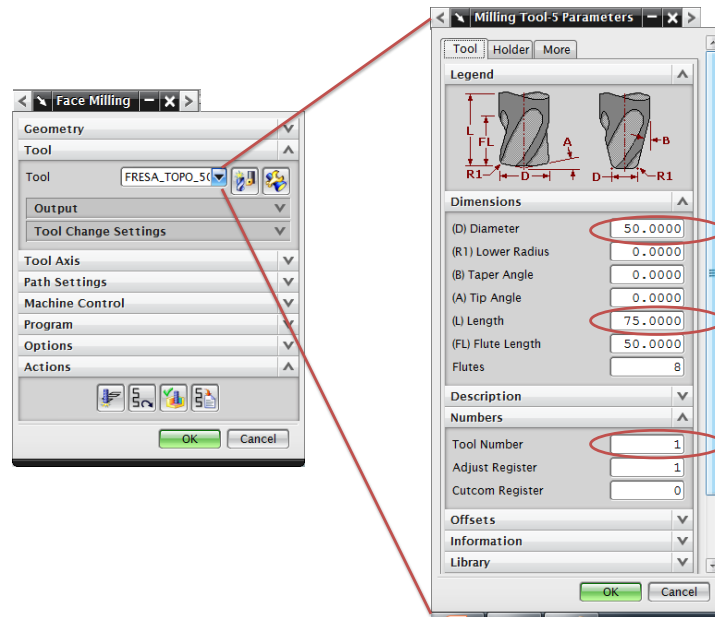


Figura 6.10 Dados da operação de faceamento e a ferramenta associada.

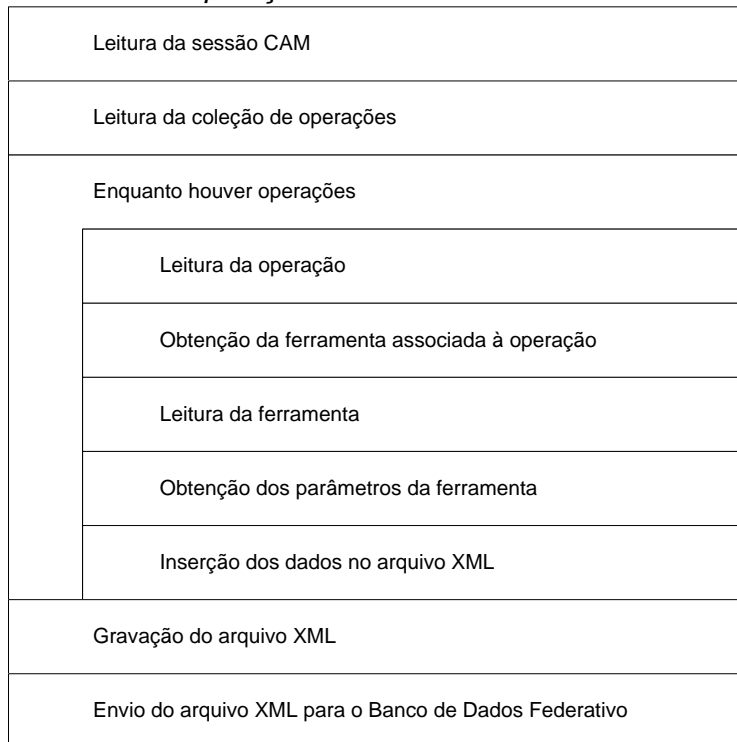


Figura 6.11 Estrutograma de leitura dos dados das operações de manufatura

O arquivo XML gerado e enviado a um banco de dados federativo tem uma estrutura do tipo lista contendo os atributos sobre a operação e sobre a ferramenta utilizada, indicando o diâmetro e o comprimento. Um trecho do arquivo está apresentado na Figura 6.12. Por ser feito em uma linguagem extensível, demais operações podem ser inseridas sem prejuízo do funcionamento do sistema, no exemplo acima o número da

ferramenta é o número que o sistema CAM utiliza para identificar a ferramenta, sendo único para cada ferramenta.

Depois de transferido para o banco de dados federativo é informado ao usuário da sessão CAM o sucesso da operação, da mesma forma como é feito para a informação do resultado da operação CAD.

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <Product Name="">
- <OperationList>
- <Operation Name="FACEAMENTO" Type="FACE_MILLING">
<diameter>50</diameter>
<number>1</number>
<length>75</length>
</Operation>
- <Operation Name="CONTORNO" Type="CAVITY_MILL">
<diameter>10</diameter>
<number>2</number>
<length>75</length>
</Operation>
- <Operation Name="REBAIXOS" Type="COUNTERBORING">
<diameter>10</diameter>
<number>2</number>
<length>75</length>
</Operation>
```

Figura 6.12 Excerto de arquivo XML com dados das operações de manufatura

6.1.3 Aplicativo para Sistema CAM - Ferramentas

As ferramentas associadas às operações são consideradas recursos de manufatura e são definidas no sistema CAM, para que estas informações possam ser distribuídas para outras aplicações foi também desenvolvido um aplicativo no sistema CAM para obtenção das informações das ferramentas e sua correta disponibilização.

Neste caso o aplicativo interno ao sistema CAM é acionado através do botão “Release Tools”, também dentro do grupo “Resources”. O aplicativo então faz uma varredura das operações e, para cada operação, a leitura da ferramenta associada. Obtida a ferramenta é feita a leitura dos parâmetros - para este caso: nome; diâmetro; tipo; sub-tipo e número. As informações obtidas são compiladas em um arquivo XML e enviadas para o banco de dados distribuído. Da mesma forma que para os aplicativos anteriores, o envio dos dados também é feito por um serviço Web.

Os dados relativos às ferramentas podem ser obtidos no sistema CAM pela caixa de diálogo apresentada na Figura 6.13.

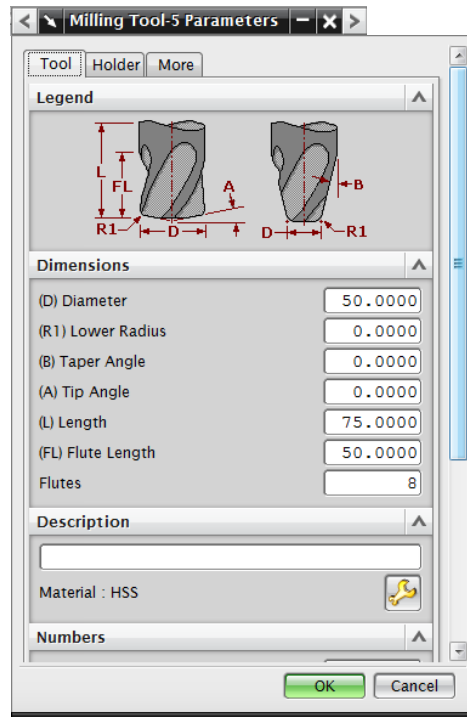


Figura 6.13. Caixa de diálogo do sistema CAM com dados da ferramenta.

O sistema CAM mantém as ferramentas em um objeto do tipo coleção de ferramentas, onde cada ferramenta é um objeto desta coleção, sendo as ferramentas definidas por uma classe com tipo e sub-tipo, sendo 9 os tipos e 26 os sub-tipos. O aplicativo faz a varredura de toda a coleção de ferramentas e agrega os atributos específicos.

Para cada ferramenta encontrada é criado um objeto com os atributos designados e esse objeto é inserido em uma coleção de objetos. Como é possível que uma mesma ferramenta seja utilizada em mais de uma operação é feita uma verificação inicial da existência da ferramenta na coleção para evitar duplicidade.

A coleção de ferramentas é transformada em um arquivo XML que por sua vez é transferido para o banco de dados distribuído, o estrutograma apresentado na Figura 6.14 mostra a sequência de funcionamento do aplicativo.

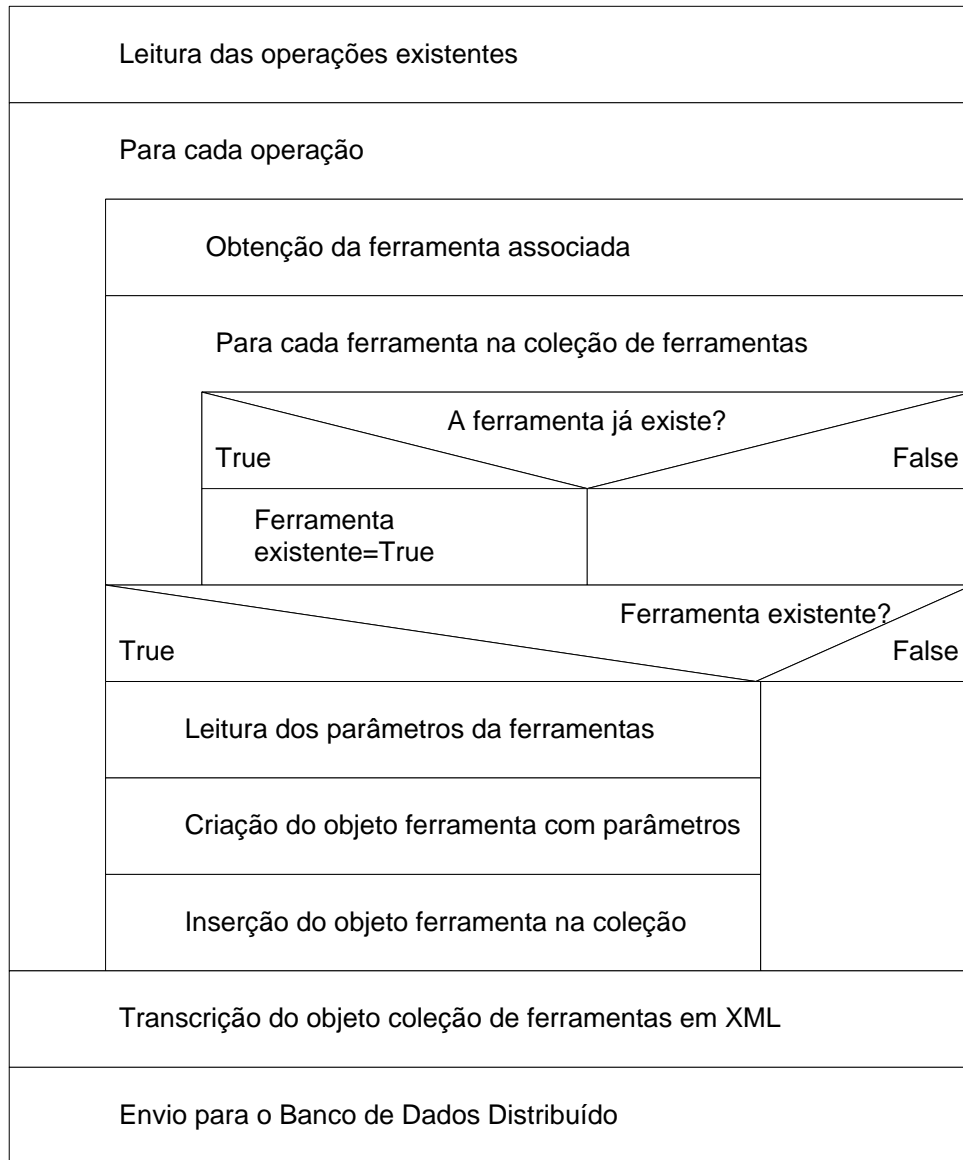


Figura 6.14 Estrutograma do aplicativo de leitura de ferramentas

O arquivo XML disponibilizado na Web contém as informações das ferramentas, um excerto do arquivo pode ser visto na Figura 6.15. Neste exemplo são apresentadas duas ferramentas, seus tipos e sub-tipos e também os diâmetros e comprimentos. Igualmente aos arquivos anteriores é possível estender as informações sem prejuízo do funcionamento do sistema.

```
<?xml version="1.0"?>
-<tools_collection>
  -<tool>
    <tool_name> FRESA_TOPO_50MM </tool_name>
    <tool_type> Mill </tool_type>
    <tool_subtype> Mill 5 </tool_subtype>
    <diameter> 50 </diameter>
    <tool_number> 1 </tool_number>
  </tool>
  -<tool>
    <tool_name> FRESA_TOPO_10MM </tool_name>
    <tool_type> Mill </tool_type>
    <tool_subtype> Mill 5 </tool_subtype>
    <diameter> 10 </diameter>
    <tool_number> 2 </tool_number>
  </tool>
  ...
```

Figura 6.15 Excerto de arquivo XML com dados de ferramentas

6.2 Serviços Web

Os serviços Web elaborados para demonstração do conceito do uso de um sistema federativo no desenvolvimento do produto estão hospedados na Web. Para o desenvolvimento deste trabalho foi decidido utilizar um domínio específico e uma hospedagem contratada para que os testes com liberdade de uso não contaminassem o ambiente universitário.

A hospedagem dos serviços fica localizada em um ambiente contendo um servidor Web. Por ser utilizada a plataforma dotNet® como ambiente de desenvolvimento, o servidor Web deve ser compatível, estando, portanto instalado com o servidor de aplicativos IIS (*Internet Information Services*).

A estrutura dos arquivos e pastas mínimos para o funcionamento de um serviço Web dentro da plataforma dotNet®, representada na Figura 6.16, contém um arquivo que apresenta as funções existentes nos serviços e outro arquivo com as configurações para a execução destes serviços, uma pasta para conter a biblioteca de vínculos dinâmicos e, opcionalmente, uma pasta para conter os dados.

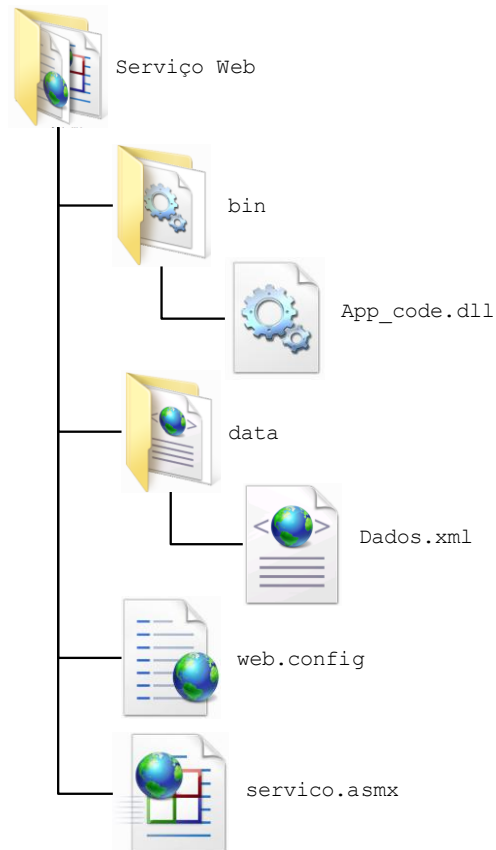


Figura 6.16 Estrutura de pastas de um serviço Web.

O controle do comportamento fica em um arquivo “web.config”. As definições encontradas nesse arquivo são sobrecarregadas sobre as funções definidas originalmente pelo servidor de aplicativos. Para cada serviço Web é necessário que exista um arquivo de configurações Web definindo as funções a serem executadas.

O arquivo que contém as funções é do tipo *Active Server Methods* (asmx) e apresenta somente um vínculo para a biblioteca de vínculos dinâmicos e ao ser solicitado gera um arquivo em hipertexto (ver Figura 6.17) que apresenta a um navegador Web (Internet Explorer, Chrome, Mozilla, Opera, Safari) a lista de funções e também a descrição do serviço na linguagem WSDL.

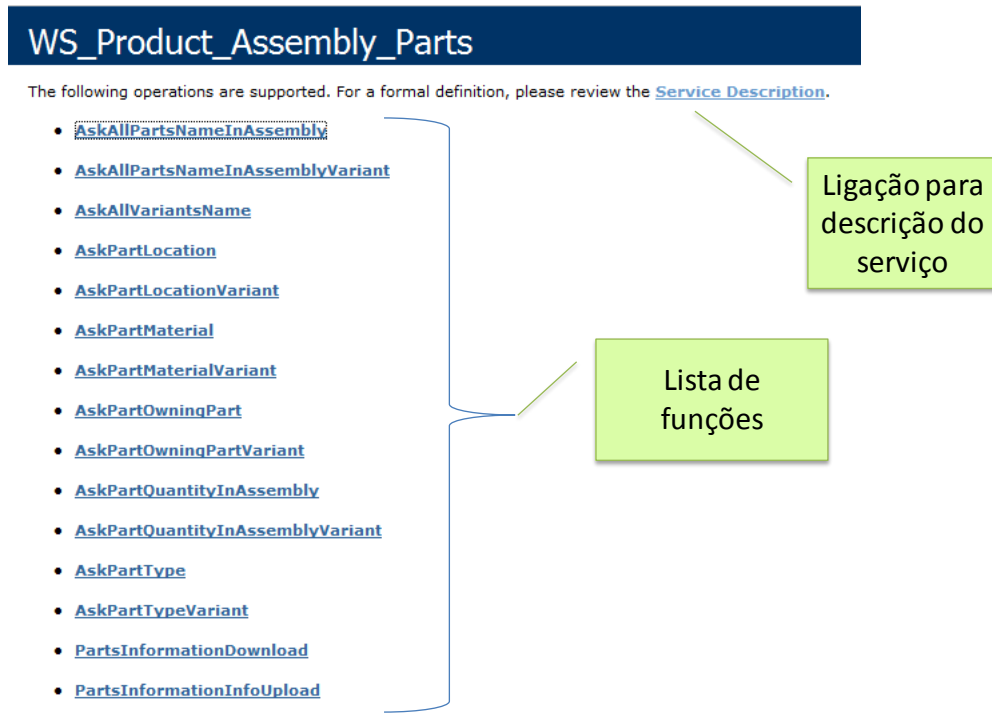


Figura 6.17 Apresentação de um serviço em um navegador.

Pelo arquivo apresentado é possível acessar a descrição do serviço através da ligação apropriada, com isso será apresentado um arquivo em XML, também possível de ser visualizado em um navegador comum, como pode ser visto na Figura 6.18. Neste arquivo estão todos os métodos de acesso e as respostas esperadas às funções do serviço Web.

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://bragecrim.net/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s1="http://bragecrim.net/AbstractTypes"
  xmlns:tns="http://bragecrim.net/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
- <wsdl:types>
  - <s:schema targetNamespace="http://bragecrim.net/" elementFormDefault="qualified">
    + <s:element name="PartsInformationInfoUpload">
    + <s:element name="PartsInformationInfoUploadResponse">
    + <s:element name="PartsInformationDownload">
    + <s:element name="PartsInformationDownloadResponse">
    + <s:element name="AskAllPartsNameInAssembly">
    - <s:element name="AskAllPartsNameInAssemblyResponse">
      - <s:complexType>
        - <s:sequence>
          <s:element name="AskAllPartsNameInAssemblyResult" type="tns:ArrayOfString" maxOccurs="1"
            minOccurs="0"/>
        </s:sequence>
      </s:complexType>
    </s:element>
    + <s:complexType name="ArrayOfString">
    + <s:element name="AskAllPartsNameInAssemblyVariant">
    + <s:element name="AskAllPartsNameInAssemblyVariantResponse">
    + <s:element name="AskPartQuantityInAssembly">
    + <s:element name="AskPartQuantityInAssemblyResponse">
    + <s:element name="AskPartQuantityInAssemblyVariant">
```

Figura 6.18 Trecho do arquivo WSDL

Ao navegar para as funções é fornecida uma página em HTML que apresenta como a comunicação é feita entre o requisitante e o provedor, mostrando os protocolos de acesso a objetos simples. Além disso, também há a opção de se testar o funcionamento da função diretamente na página apresentada, isso se dá através de um botão que requisita o serviço e retorna o arquivo XML com as respostas obtidas. A Figura 6.19 apresenta a função que retorna o nome dos componentes de um produto. O objeto de retorno dessa função é uma lista em XML, com o nome das partes cujo tipo da variável é uma cadeia de caracteres (*string*).

The screenshot shows a web page titled "WS_Product_Assembly_Parts". It includes a link for a complete list of operations and a section for testing the "AskAllPartsNameInAssembly" operation. A green callout box points to an "Invoke" button, labeling it as "Botão para testar a função". Below the button, there are two SOAP 1.1 examples. The first is a request, and the second is a response. Blue brackets on the right side of the page label these as "Objeto com a requisição" and "Objeto com a resposta".

WS_Product_Assembly_Parts

Click [here](#) for a complete list of operations.

AskAllPartsNameInAssembly

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be repl

```
POST /Product_Assembly_Parts/WS_Product_Assembly_Parts.asmx HTTP/1.1
Host: www.a3moura.eti.br
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://bragecrim.net/AskAllPartsNameInAssembly"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml
  <soap:Body>
    <AskAllPartsNameInAssembly xmlns="http://bragecrim.net/" />
  </soap:Body>
</soap:Envelope>
```

Objeto com a requisição

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml
  <soap:Body>
    <AskAllPartsNameInAssemblyResponse xmlns="http://bragecrim.net/">
      <AskAllPartsNameInAssemblyResult>
        <string>string</string>
        <string>string</string>
      </AskAllPartsNameInAssemblyResult>
    </AskAllPartsNameInAssemblyResponse>
  </soap:Body>
</soap:Envelope>
```

Objeto com a resposta

Figura 6.19 Informações sobre acesso e requisição do serviço.

Ao acionar o botão de acesso a função é retornado um arquivo XML com o nome das partes componentes do conjunto montado, previamente disponibilizado pelo sistema CAD, como pode ser visto na Figura 6.20.

```
<?xml version="1.0" encoding="UTF-8"?>
-<ArrayOfString xmlns="http://bragecrim.net/">
  <string>SCHRAUBE_DIK</string>
  <string>DECKEL_DIK</string>
  <string>KOLBEN_DIK</string>
  <string>KOLBENSTANGE_DIK</string>
  <string>GEHAEUSE</string>
  <string>BODEN_D40</string>
</ArrayOfString>
```

Figura 6.20 Exemplo de uma resposta a um serviço Web.

Os arquivos que contém os dados do produto estão armazenados em um local acessível através da Internet e no formato XML. Para a leitura deste arquivo é lançado mão de um conjunto de objetos disponíveis na plataforma dotNet pertencente à classe *Object*, parte da estrutura desta classe é apresentada na Figura 6.21 e os objetos de interesse para esta aplicação estão em destaque.

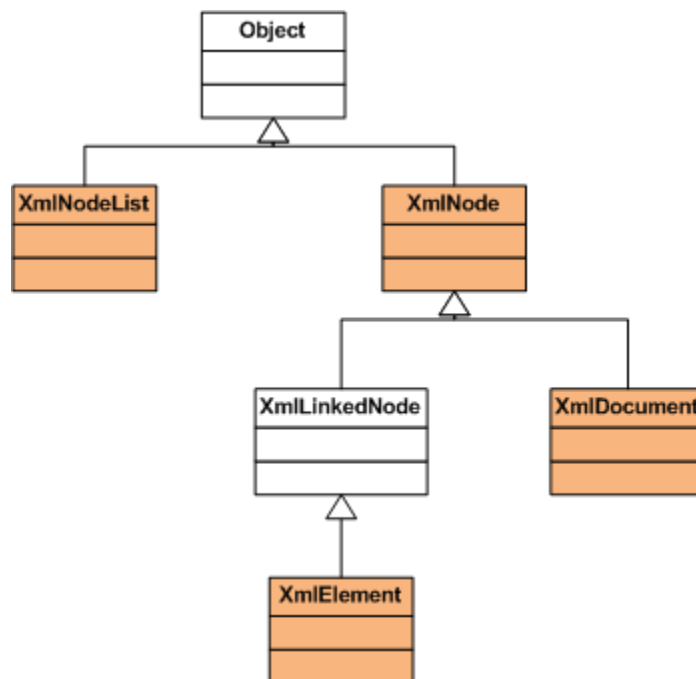


Figura 6.21 Excerto da estrutura de classe *Object*

XmlNode

A classe [XmlNode](#) representa um único nó no documento XML, possui vinte e três atributos e vinte e sete métodos, através dos quais é possível navegar e editar o nó de

um documento XML. Dentre estes métodos é utilizado, nesta aplicação, o que retorna um item especificado por um nome.

XmlNodeList

A classe [XmlNodeList](#) representa uma coleção ordenada de nós em um arquivo XML, há somente um método construtor que cria uma instância desta classe sem vínculo a nenhum documento. Existem dois atributos, um com o número de nós existentes na lista e outro que recupera um nó especificado por um índice. Oito são os métodos desta classe que permitem navegar pela lista de nós.

XmlDocument

Um documento XML é representado pela classe [XmlDocument](#) que tem três métodos construtores, um para instanciação simples, sem parâmetros, outro para instanciação com uma especificação de implementação e ainda outro para criação do objeto baseado em uma tabela. Os atributos são trinta e os métodos sessenta e quatro, considerando os métodos herdados das classes superiores. Para esta classe também estão presentes quatro controles de eventos de modo a poder verificar alterações, inserções ou remoções de nós. Nesta aplicação é utilizado somente o método “[load](#)” que carrega o arquivo XML para o objeto criado.

XmlElement

O objeto [XmlElement](#) é o que permite o acesso à informação inserida dentro de um nó, em um arquivo XML. Dos quarenta e cinco métodos existentes para este objeto, são usados apenas dois nesta aplicação, um para obtenção do atributo do elemento ([GetAttributeNode](#)), e outro para leitura do valor deste atributo ([InnerText](#)).

6.2.1 Análise e Recuperação de Dados do Produto

Considerando que os dados do produto estão na Web em um arquivo físico no formato XML, cabe ao serviço acessar este arquivo e retornar os dados solicitados.

Os dados de produto disponibilizados através do sistema CAD ficam armazenados na Web e o serviço oferece treze funções para recuperação dos dados de produto, sendo que seis são duplicadas para o produto principal e para o produto variante. As informações a serem obtidas são:

- Nomes dos componentes ([AskAllPartsName](#));
- Localização do componente ([AskPartLocation](#));
- Material do componente ([AskPartMaterial](#));
- A quem pertence o componente ([AskPartOwningPart](#));
- A quantidade de vezes que o componente ocorre no conjunto montado ([AskPartQuantityInAssembly](#));
- O tipo do componente ([AskPartType](#));

Com estas funções é possível acessar as informações do produto necessárias para a integração dos domínios de produto, processos e recursos. Por exemplo, para obter a informação do material de um componente existente em um conjunto montado usa-se a seguinte sequência de operações:

- Obter o nome dos componentes ([AskAllPartsNameInAssembly](#));
- Com o nome do componente, obter o material ([AskPartMaterial](#))

A Figura 6.22 mostra a sequência para obtenção do material de um componente usando um navegador comum. Na primeira parte é feito o chamado à função para obtenção dos nomes dos componentes existentes no conjunto montado ([AskAllPartsNameIn_Assembly](#)) com o acionar do botão “Invoke”, para esta função não é necessário nenhum parâmetro a ser passado.

A resposta é um arquivo XML com os nomes dos componentes, com os nomes dos componentes, no exemplo apresentado é escolhido o componente “DECKEL_DIK”. A seguir é chamada à função para obtenção do material do componente

([AskPartMaterial](#)), que exige no nome do componente como parâmetro a ser passado para a função, neste caso se escolheu o componente “DECKEL_DIK”. A resposta a esta função é também um arquivo XML que contém o material do componente, no caso em tela a resposta é “Aluminum_2014”.

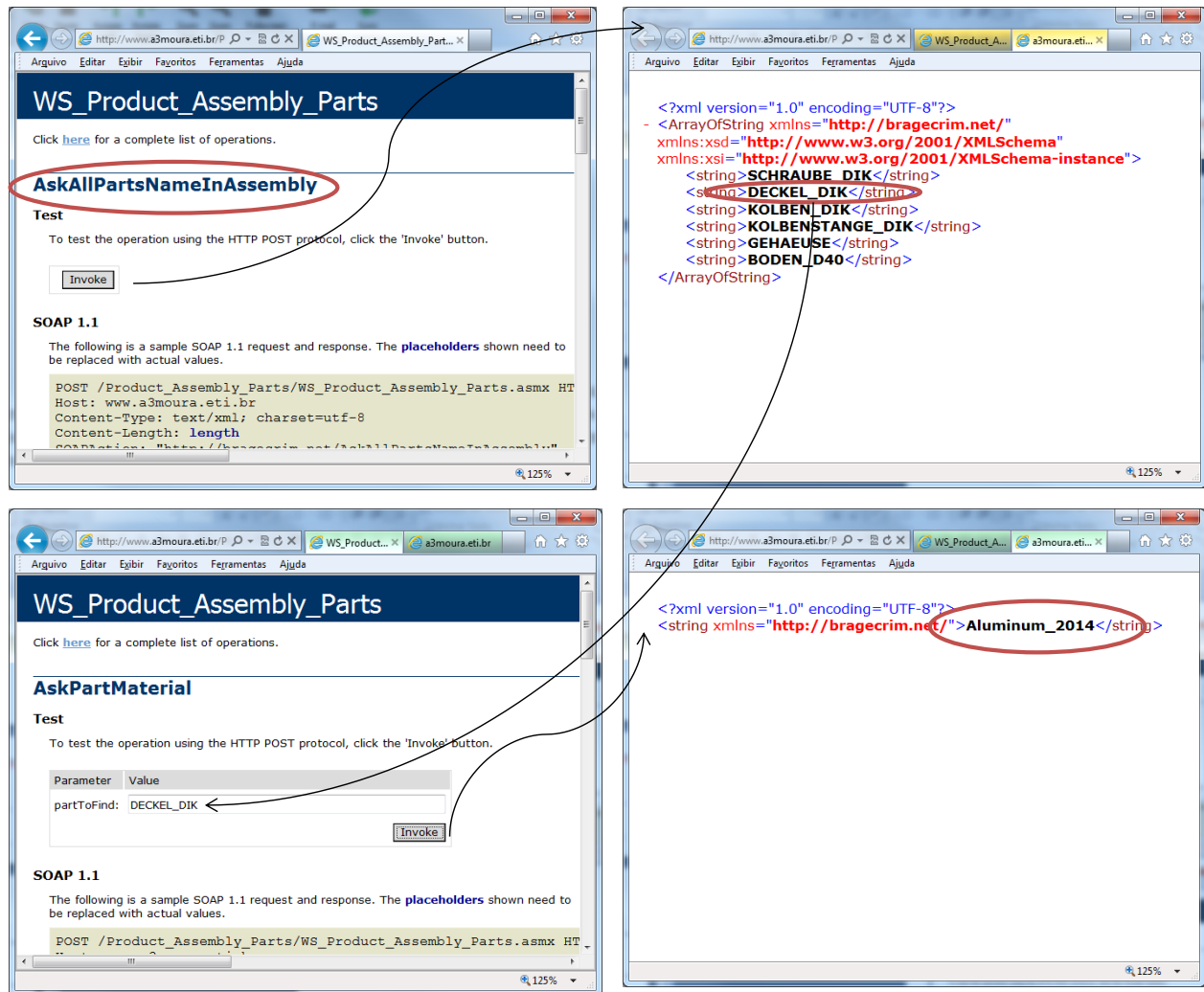


Figura 6.22 Sequência para obter o material de um componente.

Cabe lembrar que toda a comunicação com o serviço Web é feita através de objetos encapsulados pelo protocolo SOAP, tanto o nome do componente do qual se quer saber o material, quanto a resposta do material do qual é feito o componente. No entanto, toda esta comunicação é transparente ao usuário do serviço.

A sequência e as telas apresentadas são somente exemplos e servem para teste e validação dos serviços, o uso real do serviço é feito através de um navegador capaz de

tratar estes dados adequadamente. Além disso, as outras informações podem ser obtidas a partir do nome do componente, como: localização; material e componente-pai. O conjunto destas informações possibilita a reconstrução da estrutura do produto.

6.2.2 Análise e Recuperação de Dados de Processo

Os dados de processo ficam armazenados em um arquivo XML na Web e a recuperação das informações nele contidas é feita através de um serviço Web, com seis funções específicas, a saber:

- Nome de todas as operações ([AskAllOperationsNames](#));
- Nome das operações de um tipo previamente especificado ([AskAllOperationsOfType](#));
- Nome dos tipos existentes ([AskAllTypesOfOperations](#));
- Nome das ferramentas ([AskAllOperationTool](#));
- Tipo de uma operação previamente especificada ([AskOperationType](#));
- Nome da ferramenta vinculada a uma operação previamente especificada ([AskOperationType](#)).

Com estas funções é possível acessar todo o conjunto de informações do arquivo de processos. Por exemplo, para saber o tipo de uma operação definida para um produto é preciso primeiro obter a lista com todas as operações existentes, isso é feito através da função “[AskAllOperationsNames](#)”, que retornará um arquivo XML com a lista de operações. Obtida esta informação é possível, com o uso da função “[AskOperationType](#)” obter o tipo da operação, desde que especificado o nome desta operação.

A Figura 6.23 mostra a sequência para obtenção do tipo de uma operação usando um navegador comum. Inicialmente é acionada a função “[AskAllOperationsNames](#)” com o botão “[Invoke](#)”, para esta função não é necessário nenhum parâmetro. A resposta é um

arquivo XML com todas as operações, no caso é escolhida a operação “S1_PLANEN_T1”, que é passado como parâmetro para a função “AskOperationType”. Esta função, por sua vez, retorna a resposta “FACE_MILLING”.

The image shows two screenshots of a web application interface for 'WS_Process_Mach_Operation'.

The top screenshot shows the 'AskAllOperationsNames' function. The 'Test' section has an 'Invoke' button. Below it, the SOAP 1.1 request and response are shown. The response is an XML array of strings containing various operation names like S1_PLANEN_T1, S1_AUSSENK_T6, etc.

The bottom screenshot shows the 'AskOperationType' function. The 'Test' section has a table with 'Parameter' and 'Value' columns. The 'operationName' parameter is set to 'S1_PLANEN_T1'. An 'Invoke' button is present. Below it, the SOAP 1.1 request and response are shown. The response is an XML string containing the value 'FACE_MILLING'.

Figura 6.23. Sequência para obter o tipo de uma operação.

Toda a comunicação é feita com as informações descritas em XML e envelopadas em um protocolo de objeto simples (SOAP).

6.2.3 Análise e Recuperação de Dados de Recursos

Os recursos aqui considerados são as ferramentas utilizadas nos processos de usinagem. As informações das ferramentas foram obtidas diretamente do sistema CAM, por um aplicativo específico como apresentado em 6.1.3, e transferidas para a Web em um arquivo em linguagem XML.

As funções para a recuperação dos dados das ferramentas são três:

- Tipo e sub-tipo da ferramenta ([AskToolTypeAndSubType](#)).
- Diâmetro da ferramenta ([AskToolDiameter](#));

- Número da ferramenta ([AskToolNumber](#));

Para estas três funções é necessário informar o nome da ferramenta da qual deseja obter a informação.

Estas funções podem ser verificadas em qualquer navegador com acesso à Internet, sendo inclusive disponibilizadas as informações de acesso ao serviço Web.

Como exemplo, a Figura 6.24 apresenta uma sequência de telas com a obtenção do tipo e sub-tipo da ferramenta

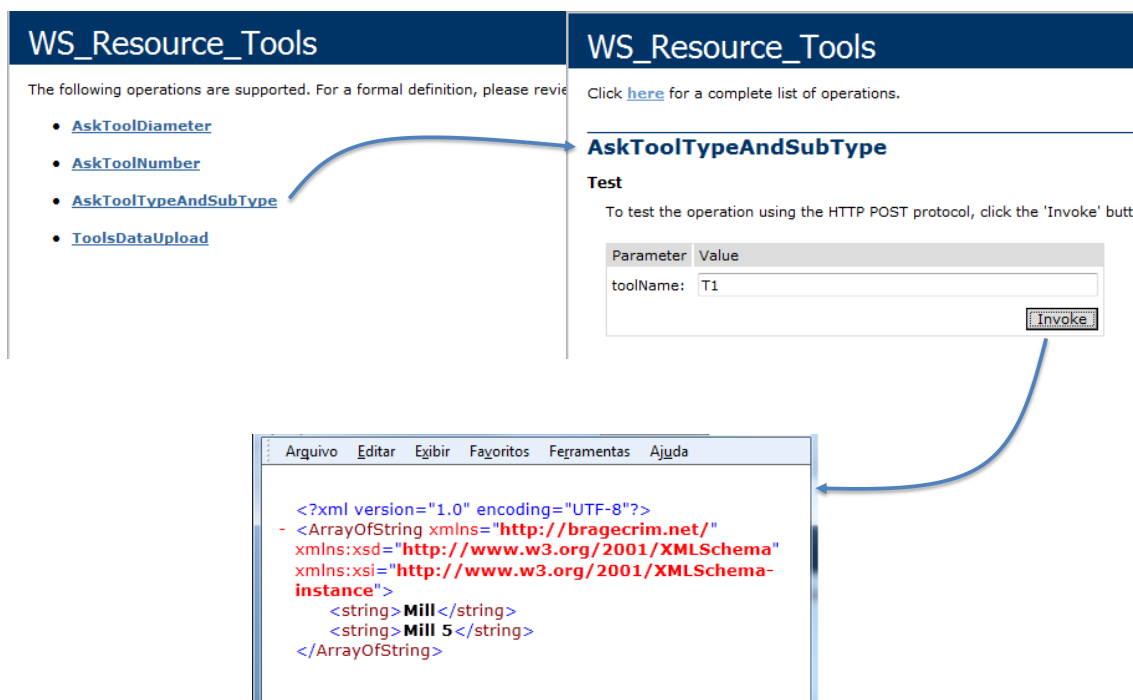


Figura 6.24 Telas com a sequência de obtenção de tipo e subtipo de uma ferramenta.

6.3 Aplicativo para Integração de Dados

A integração dos dados de produto, processos e recursos é feita através de um aplicativo que acessa os serviços Web e possibilita a criação de vínculos entre os três domínios. Esse aplicativo foi desenvolvido em VB.Net, dentro do plataforma dotNet, da mesma maneira que os demais componentes de software.

O aplicativo apresenta no canto superior à esquerda três botões do tipo rádio para indicar o relacionamento entre os três domínios compreendidos. Uma vez escolhidos os

dois domínios é possível requisitar os dados dos bancos de dados federativos por meio do botão “*Request Web Service*”. A Figura 6.25 apresenta, na primeira coluna, o resultado dos dados relativos aos componentes do cilindro pneumático projetado no sistema CAD, informando a quantidade de vezes que o componente aparece no conjunto e o nome do componente, como apresentado na seção 6.1.1; na primeira linha são apresentadas as ferramentas listadas no arquivo CAM como demonstrado na seção 6.1.3.

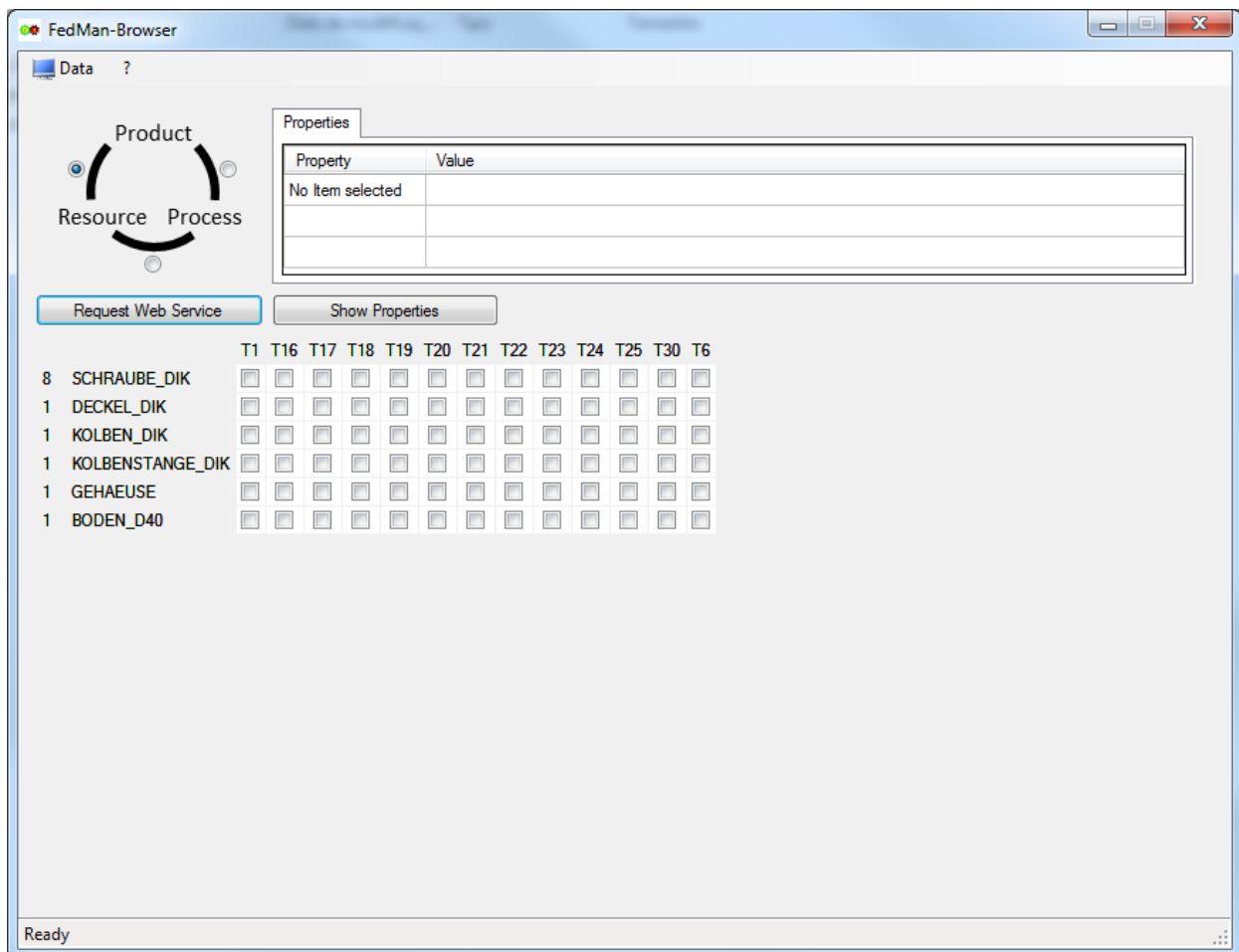


Figura 6.25 Tela do aplicativo com dados de produto e recursos.

A opção de visualizar os domínios de recursos e processos é possível pelo mesmo procedimento, neste caso as ferramentas são apresentadas da mesma forma que a anterior e os processos de usinagem são listados na primeira coluna, uma tela de exemplo é mostrada na Figura 6.26.

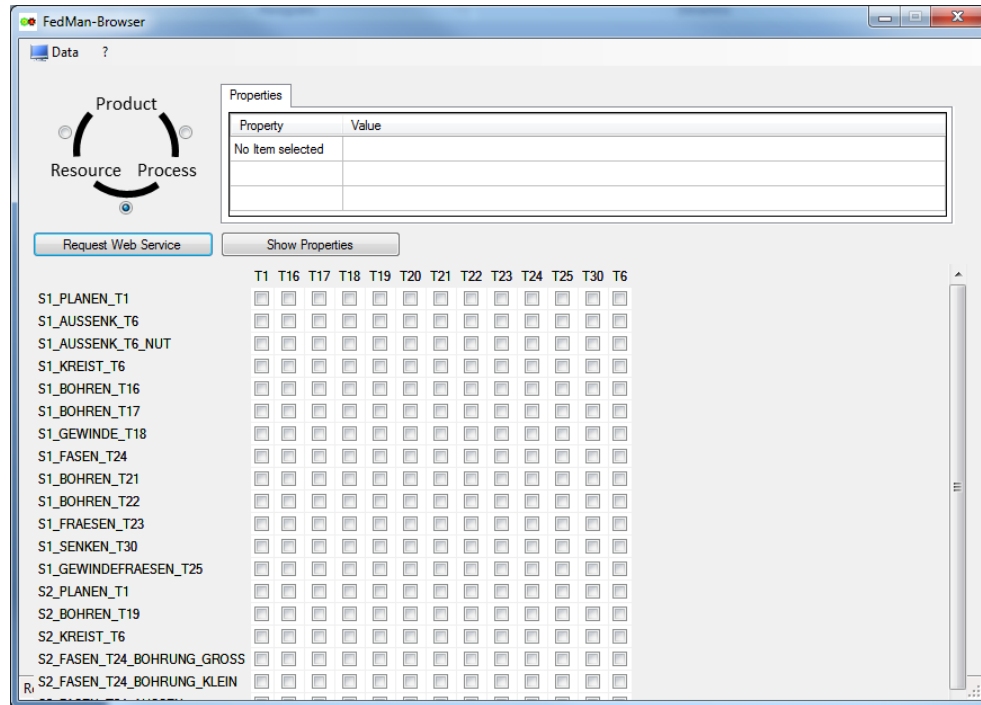


Figura 6.26 Tela do aplicativo com dados de processos e recursos.

Por fim, é possível visualizar dados de processos e produto utilizando o mesmo procedimento. Além disso, as características de cada um dos componentes podem ser visualizadas com a opção de clicar sobre o componente e acionar o botão “Show Properties”. A Figura 6.27 apresenta a tela de processos e produtos e também as características do parafuso (*Schraube_DiK*) escolhido.

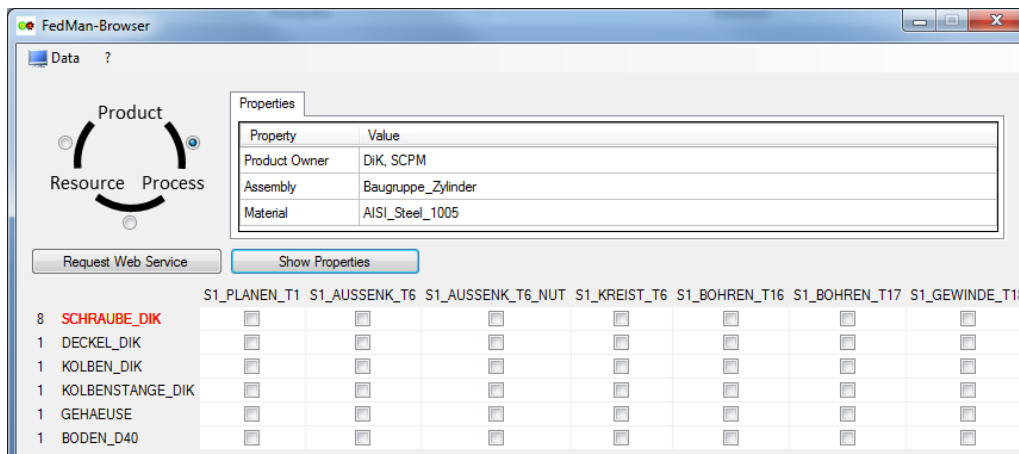


Figura 6.27 Tela do aplicativo com dados de produto e processo e também as características do componente escolhido.

A qualquer momento o projetista pode vincular componentes do produto a processos ou a recursos e também recursos a processos. Estes vínculos são possíveis pelos botões

de opções apresentados nas matrizes. Os vínculos podem ser arquivados e recuperados posteriormente

6.4 Serviços Independentes

Dois serviços, descritos a seguir, foram desenvolvidos como parte deste trabalho, porém sem integrar o aplicativo de integração. São serviços que funcionam dentro da proposta deste trabalho e ajudam a demonstrar a possibilidade de colaboração e distribuição com o uso de serviços Web.

O primeiro serviço faz uma estimativa do tempo de manufatura e o segundo sugere pontos de inspeção para uma superfície livre, de modo integrado a um sistema de medir por coordenadas.

6.4.1 Estimativa do tempo de manufatura

A determinação do tempo de manufatura é um fator chave no processo decisório sobre os parâmetros de corte e estratégias de manufatura. Muitas vezes não se encontram sistemas capazes de estimar o tempo de manufatura com exatidão suficiente devido à falta de conhecimento sobre os valores mínimos das velocidades de avanço reais, que diferem das velocidades de avanço programadas pelos projetistas devido às limitações do equipamento ou das ferramentas. Várias publicações sobre este assunto propõem diferentes métodos para a estimativa do tempo de manufatura, dentre elas: (JONG *et al.*, 2009); (SO *et al.*, 2007); (MAROPOULOS, 2000); (LAI-YUEN e LEE, 2002); (MONARO e HELLENO, 2010) e (HEO *et al.*, 2006).

Para esse serviço será utilizado o método proposto por Heo *et al.* (2006) de modo a demonstrar a integração entre processos e recursos em um ambiente federativo com o uso de serviços Web. Nesse método a distância para aceleração e desaceleração são consideradas como mostra a Figura 6.28. O cálculo das distâncias considera parâmetros intrínsecos da máquina, que podem ser obtidos com as informações do fabricante ou com experimentos práticos.

Considerar a obtenção de um tempo de manufatura estimado como um serviço implica em que o usuário desse serviço envie uma solicitação para este cálculo entregando como informação inicial o arquivo CLData (*Cutter Location Data File*) que é gerado por um sistema CAM e contém os pontos e as trajetórias da usinagem, enviado o arquivo o usuário recebe como resposta o tempo estimado.

Método de cálculo do tempo de manufatura

(HEO *et al.*, 2006) propõem classificar os blocos de programação numérica em quatro casos, como apresentado na Figura 6.28. No primeiro caso (a) o bloco NC tem tamanho suficiente para que haja aceleração, estabilização e desaceleração, portanto o tempo de manufatura pode ser calculado com base na equação 1.

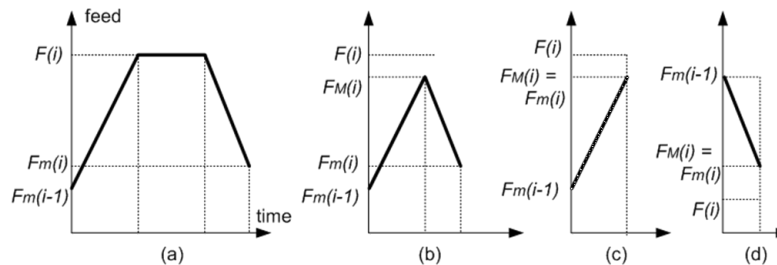


Figura 6.28 Quatro comportamentos para o avanço (HEO *et al.*, 2006).

$$t(i) = \frac{F(i) - F_m(i-1)}{acc} + \frac{F(i) - F_m(i)}{acc} + \frac{\Delta_i - \Delta_{ai} - \Delta_{di}}{F(i)}$$

Equação 1

sendo:

$t(i)$ = tempo de manufatura para o bloco [min]

$F(i)$ = velocidade de avanço considerada para o bloco [mm/min]

$F_m(i-1)$ = velocidade de avanço inicial para o i -ésimo bloco [mm/min] considerado igual a velocidade de avanço final do bloco anterior

$F_m(i)$ = velocidade de avanço final do i -ésimo bloco [mm/min]

acc = aceleração [mm/min²]

Δ_i = comprimento do i-ésimo bloco [mm]

Δ_{ai} = espaço de aceleração do i-ésimo bloco [mm]

Δ_{di} = espaço de desaceleração do i-ésimo bloco [mm]

Porém, se o bloco NC não é suficientemente longo para que haja aceleração e desaceleração a velocidade de avanço considerada no projeto não será alcançada e uma velocidade máxima menor deve ser considerada. O bloco NC nessa condição tem somente duas seções, uma para a aceleração e outra para desaceleração, como apresentado na Figura 6.28 (b). A tempo de manufatura nesse caso é calculado pela equação 2.

$$t(i) = \frac{2F(i) - F_m(i-1) - F_m(i)}{acc}$$

Equação 2

sendo:

$t(i)$ = tempo de manufatura do i-ésimo bloco [min]

$F(i)$ = velocidade de avanço do i-ésimo bloco [mm/min]

$F_m(i-1)$ = velocidade de avanço inicial do i-ésimo bloco [mm/min] considerado igual à velocidade inicial do bloco anterior

$F_m(i)$ = velocidade de avanço final do i-ésimo bloco [mm/min]

acc = aceleração [mm/min²]

Alguns blocos podem ser tão pequenos que não seja possível completar a aceleração Figura 6.28 (c) ou a desaceleração Figura 6.28 (d). Nesses casos uma velocidade de avanço final diferente deve ser calculada de acordo com as equações 3 e 4.

$$t(i) = \frac{F_m(i) - F_m(i-1)}{acc}$$

Equação 3

$$t(i) = \frac{F_m(i-1) - F_m(i)}{acc}$$

Equação 4

sendo:

$t(i)$ = tempo de manufatura para o i -ésimo bloco [min]

$F_m(i-1)$ = velocidade de avanço inicial para o i -ésimo bloco [mm/min] considerado igual a velocidade de avanço final do bloco anterior

$F_m(i)$ = velocidade de avanço final do i -ésimo bloco [mm/min]

acc = aceleração [mm/min²]

Serviço Web para cálculo do tempo de manufatura

Para o cálculo estimado do tempo de manufatura pelo serviço Web duas informações são necessárias: o programa NC no formato CLData e a velocidade de avanço programada, determinada pelas características do conjunto envolvido, como condições de corte, ferramenta, peça e máquina-ferramenta. Este serviço está baseado nos experimentos realizados por Monaro e Helleno (2010), e portanto, é utilizada a aceleração da máquina-ferramenta obtida experimentalmente.

Por trabalhar com a troca de mensagens, o serviço Web solicita do usuário o arquivo CLData e responde com o tempo estimado de manufatura, sendo que a informação sobre a velocidade de avanço é extraída do próprio arquivo CLData. A mensagem enviada ao serviço Web é a localização física do arquivo CLData que é transferido para um banco de dados na Web.

O envio do arquivo CLData e o retorno com o tempo de manufatura estimado foram incluídos no próprio sistema CAM aproveitando a caixa de diálogo com as demais funções, dentro do grupo “Extras” como pode ser visto na Figura 6.29. Ao clicar o botão

“Release CLData File” é acionada a função de leitura do arquivo CLData definido pelo sistema CAM, em seguida este arquivo é enviado para a Web e é automaticamente acionada a função de estimativa do tempo de usinagem. O valor retornado é apresentado na própria caixa de diálogo. As demais funções do serviço Web como quantidade de blocos e tipos de blocos estão acessíveis diretamente no serviço Web.

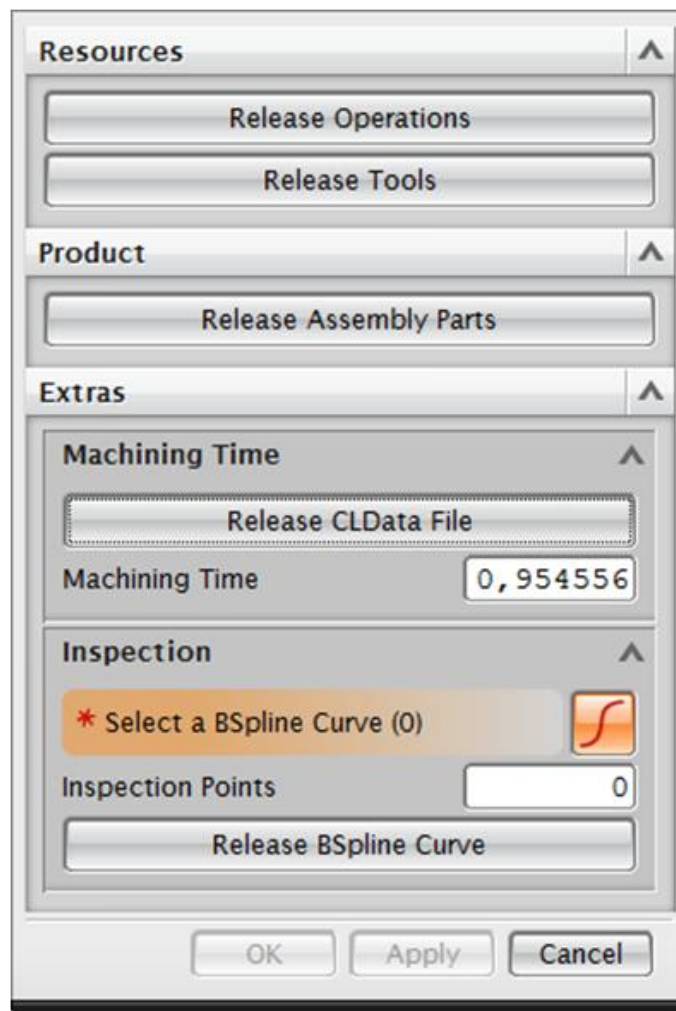


Figura 6.29 Caixa de diálogo com a opção de tempo de manufatura

De posse do arquivo o serviço efetua os cálculos de acordo com o estrutograma apresentado na Figura 6.30 e conforme o método descrito pode informar quantos blocos há e de quais tipos; a quantidade total de blocos; a velocidade de avanço e o tempo estimado para a manufatura.

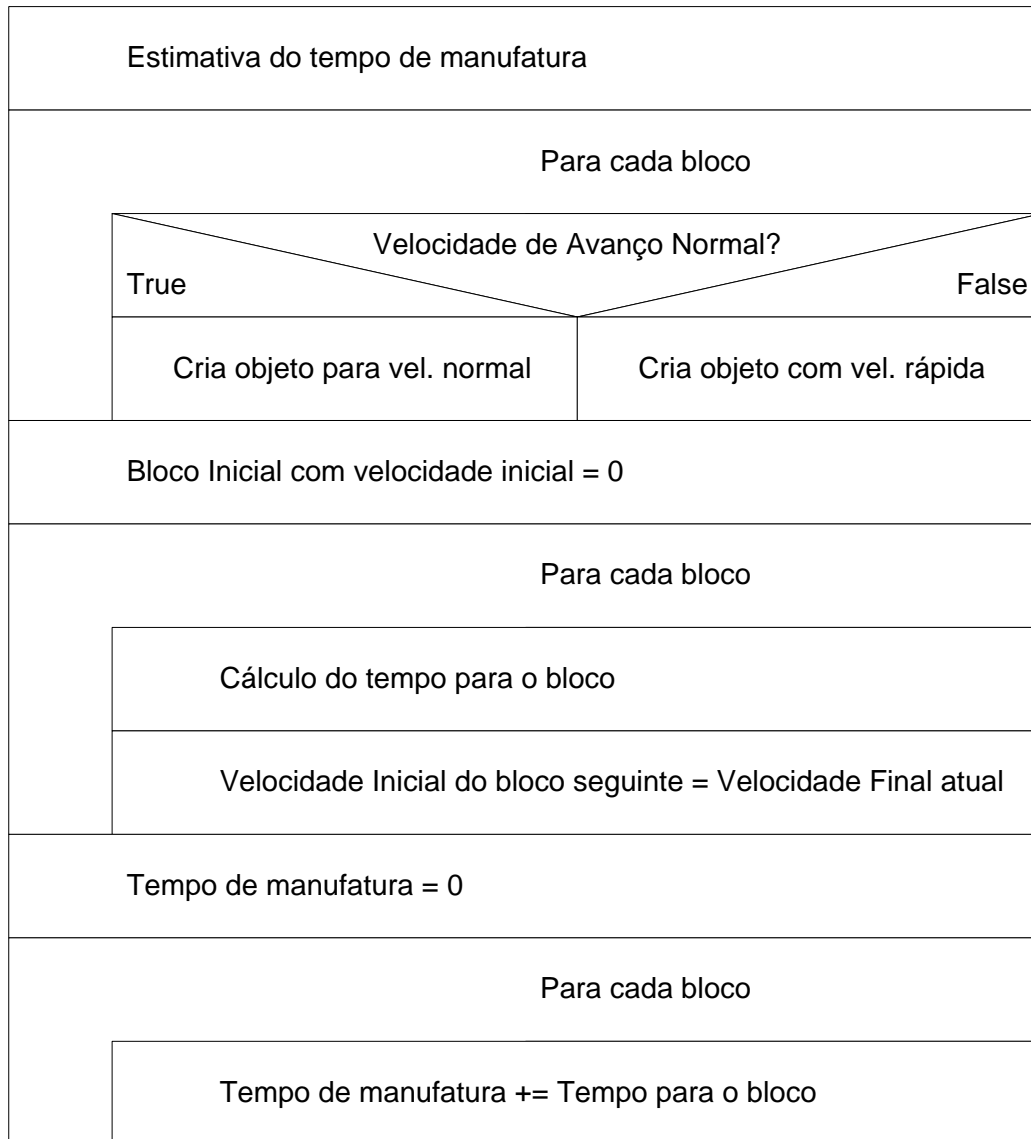


Figura 6.30 Estrutograma para estimativa do tempo de manufatura.

O serviço Web separa cada bloco de acordo com a classificação proposta por Heo *et al.* (2006), e para cada caso determina o tempo de manufatura. Em seguida é feita a somatória dos tempos dos blocos para obter o tempo de manufatura total.

A Figura 6.30 apresenta o cálculo para o tempo de manufatura de um bloco, inicialmente é calculada a velocidade de avanço final do bloco, enquanto que a velocidade de avanço inicial é considerada como sendo a velocidade de avanço final do bloco anterior. Em princípio a velocidade máxima de avanço é considerada como sendo a velocidade de avanço programada. Com estes parâmetros definidos é calculado o espaço necessário para acelerar e para desacelerar. Se o tamanho do bloco é

suficiente para comportar estes espaços, significa que o bloco é do tipo (a) e contém três setores, sendo então calculado o tempo de manufatura como visto anteriormente. No entanto, se o bloco não é grande o suficiente um novo cálculo da velocidade de avanço é feito com base na equação 5.

$$F_M(i) = \sqrt{acc \cdot \Delta_i + \frac{F_m^2(i-1) + F_m^2(i)}{2}} \quad \text{Equação 5}$$

sendo:

$F_M(i)$ = velocidade de avanço máxima para o i-ésimo bloco [mm/min]

$F_m(i-1)$ = velocidade de avanço inicial para o i-ésimo bloco [mm/min] considerado igual a velocidade de avanço final do bloco anterior

$F_m(i)$ = velocidade de avanço final do i-ésimo bloco [mm/min]

acc = aceleração [mm/min²]

Δ_i = comprimento do i-ésimo bloco [mm]

Entretanto, alguns blocos não são suficientes para acelerar e desacelerar. Quando isso ocorre o espaço de aceleração ou desaceleração é igual a zero e o bloco ou é do tipo (c) - somente aceleração; ou do tipo (d) - somente desaceleração.

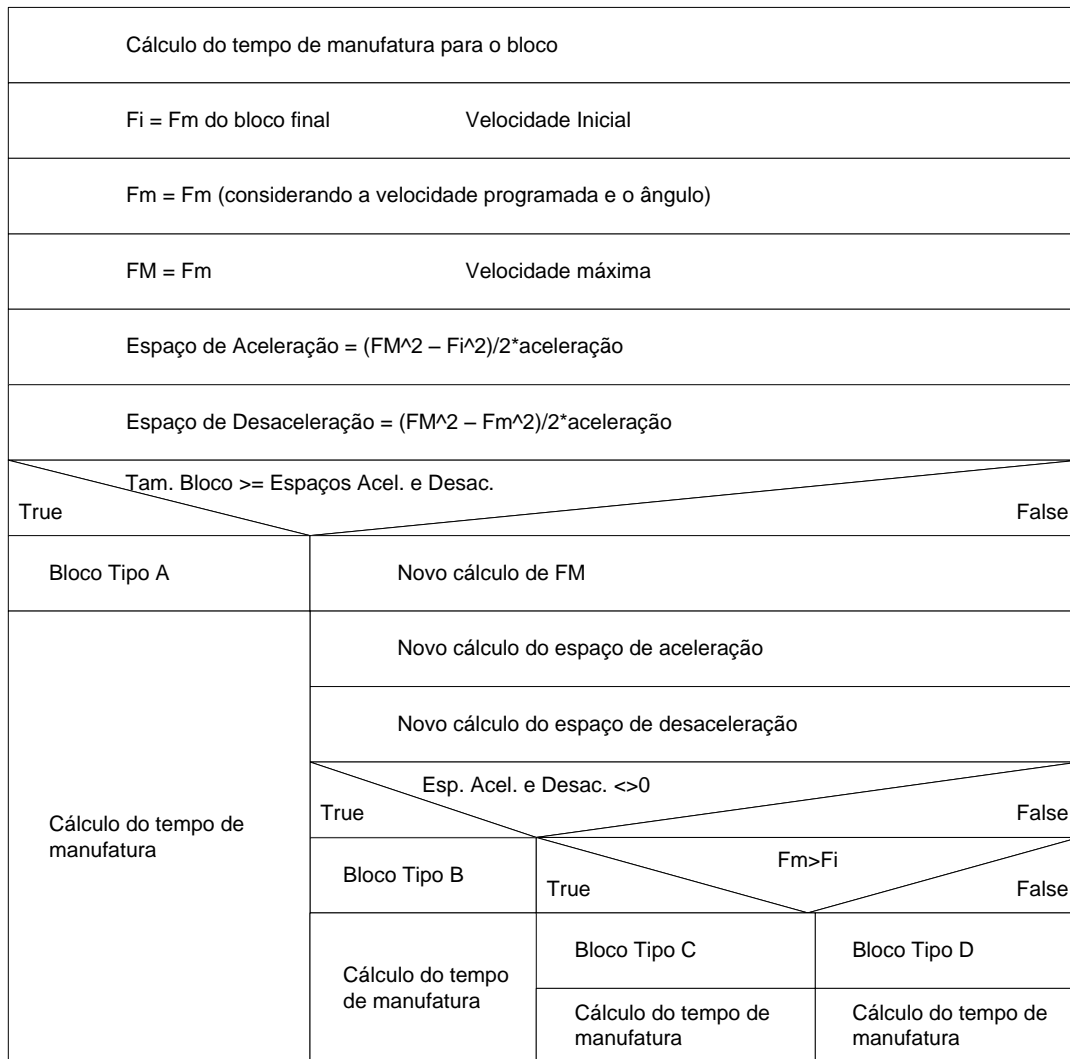


Figura 6.31 Estrutograma para o cálculo do tempo de manufatura para um bloco.

O tempo de manufatura aqui apresentado é feito com base experimental e é válido para máquinas com as mesmas características da utilizada nos testes efetuados, especialmente no que concerne a aceleração e desaceleração.

6.4.2 Determinação de pontos para inspeção de superfícies esculpidas

A determinação de pontos para inspeção de superfícies esculpidas é feito por meio de um serviço Web que recebe uma requisição e devolve um arquivo pronto para utilização em uma máquina de medir por coordenadas (MMC). Na requisição do serviço são enviados os dados de uma curva sobre uma superfície. Toda a troca de dados é feita por mensagens em XML envelopadas em protocolo SOAP.

O modelo de integração entre neste caso é entre os sistemas CAD e CAI. Inicialmente o usuário do sistema CAD escolhe uma curva B-Spline sobre a superfície a ser medida e envia para a Web o arquivo em XML dessa curva. Este envio deve corresponder à decisão do projetista de que a superfície está com nível de maturidade suficiente para ser trabalhada por outras aplicações, de acordo com os preceitos da engenharia simultânea. O sistema CAI, por sua vez, envia uma requisição ao serviço Web através de uma mensagem em XML envelopada com o protocolo SOAP. O serviço Web recebe a mensagem, acessa o arquivo da curva B-Spline, analisa os dados contidos, extrai as informações necessárias, faz o cálculo dos pontos a serem medidos, gera um arquivo final, envelopa e o envia de volta para o sistema CAI.

Na Figura 6.32 é apresentado o fluxo de informações na integração CAD/CAI. O usuário do sistema CAD, de posse do modelo geométrico, escolhe uma seção da peça que deseja inspecionar e como resultado recebe uma curva B-Spline. Esta curva complexa é então enviada para o serviço Web juntamente com a escolha de uma estratégia de medição, escolhida entre as disponibilizadas por este serviço.

O aplicativo que oferece este serviço calcula os pontos de medição e o vetor que indica a aproximação do apalpador da MMC de acordo com a estratégia escolhida.

O sistema CAI agrega o modelo geométrico do sistema CAD aos arquivos de pontos, gerando uma solicitação de medição. A medição feita com base nesta solicitação gera o relatório de inspeção. Pelo fato do serviço Web estar em uma camada diferente dos sistemas CAD e CAI há a possibilidade de atualização constante e de total independência no desenvolvimento.

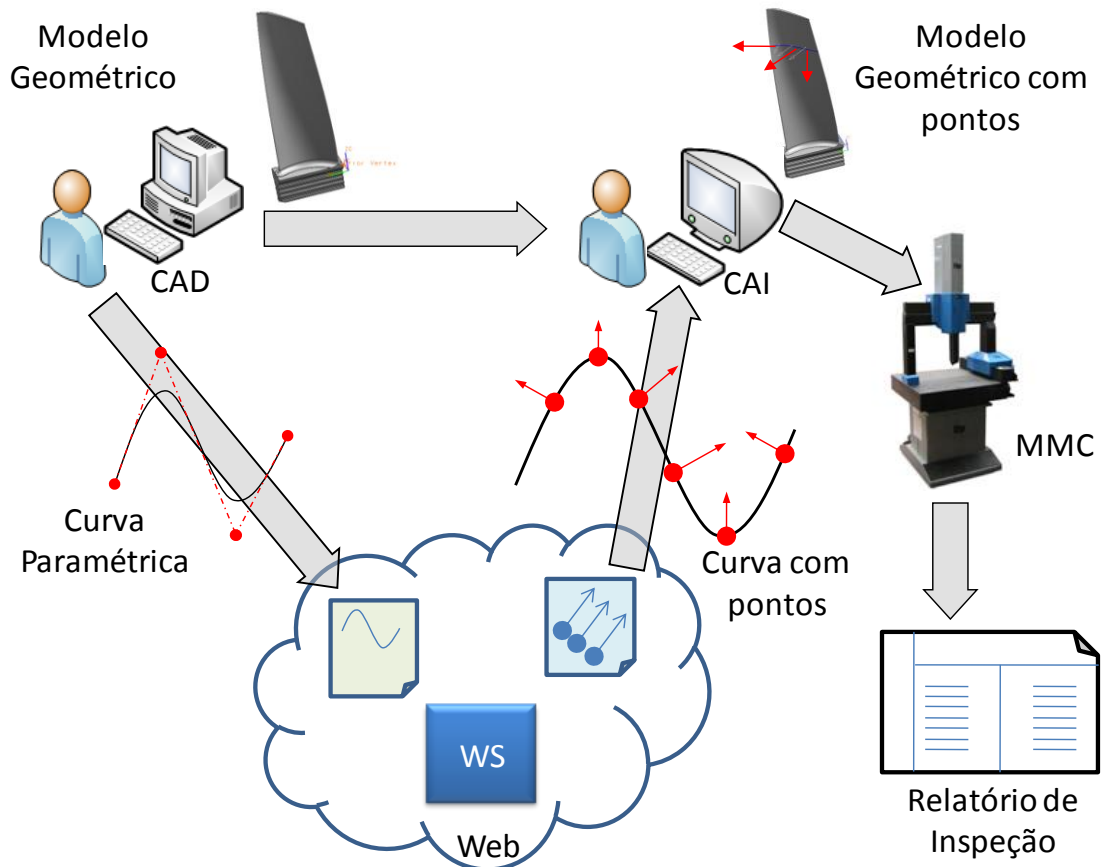


Figura 6.32 Fluxo de Informação na Integração CAD/CAI com serviços Serviços Web

Envio da B-Spline ao Serviço Web pelo cliente CAD

Uma superfície esculpida é construída tendo por base curvas Splines, como mostra a Figura 6.33, sendo estas, uma das mais fundamentais formas para curvas paramétricas (Farin, 1984).

A inspeção de superfícies esculpidas por máquinas de medir por coordenadas exige que seja enviado para o sistema CAI um conjunto de pontos a serem medidos. Estes pontos pertencem a uma curva na superfície a ser medida.

O uso de *B-Splines* (abreviaturas para *Basis Splines*) como uma ferramenta de representação geométrica remonta a 1960 no trabalho de “de Boor” no laboratório de pesquisas da General Motors, o cálculo recursivo de curvas B-Spline é hoje conhecido como algoritmo de “de Boor” (FARIN, 1984).

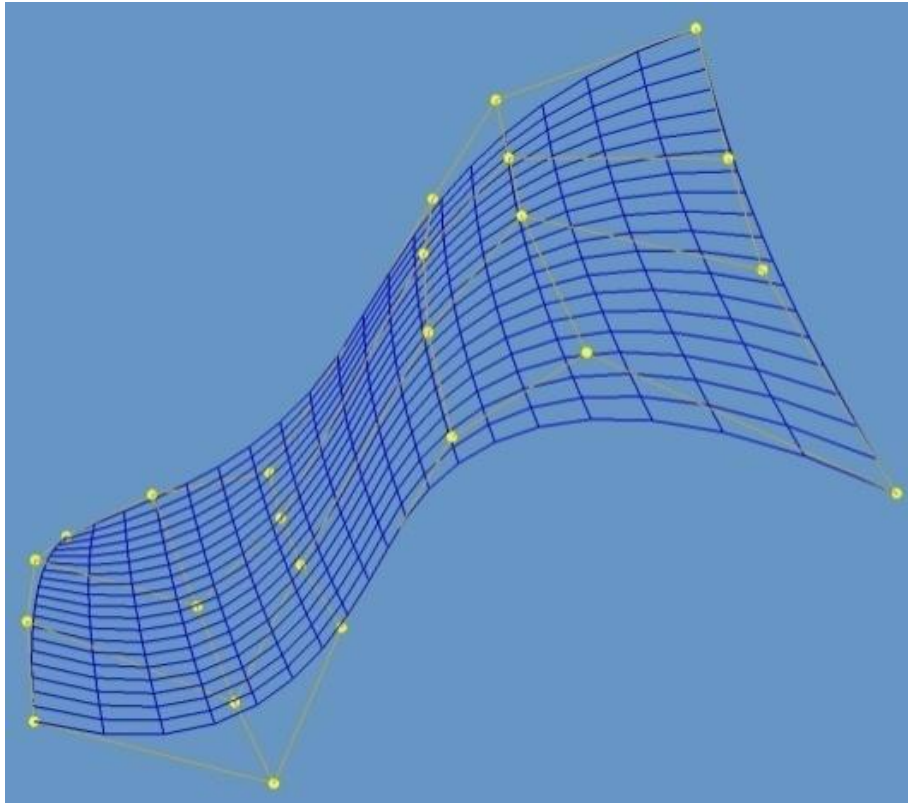


Figura 6.33 Superfície esculpida construída com curvas *B-Splines*

Para a construção de uma curva *B-Spline* necessita-se de um conjunto de pontos (que formam o polígono de controle) e um vetor de nós com ao menos um coeficiente para cada ponto de controle.

Os vértices do polígono de controle e o vetor de nós são as informações necessárias e suficientes para a definição de uma curva *B-Spline*. No entanto, para uma maior amplitude de troca de dados, foi escolhido representar a curva de acordo com as prescrições das normas ISO 10303, protocolo de aplicação AP 214, atendendo aos requisitos da Parte 28 que trata da geração de arquivos de troca de dados em formato XML.

O diagrama em Express-G que representa o modelo da curva de acordo com as normas STEP 10303 - (ISO/IS 10303-238, 2006) é apresentado na Figura 6.34.

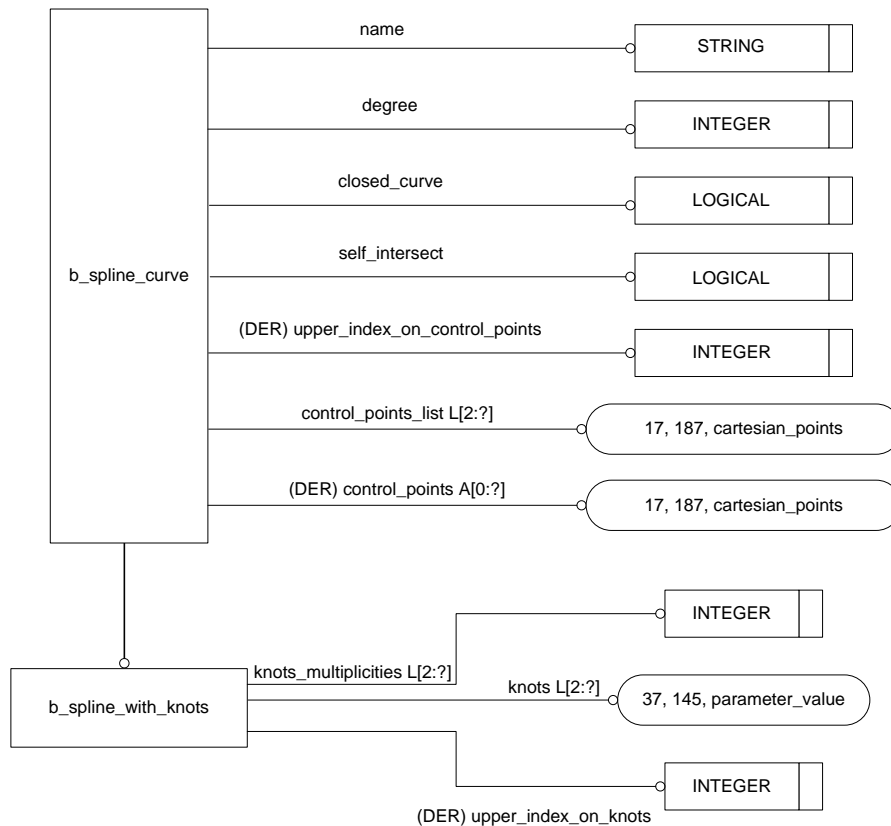


Figura 6.34 Diagrama em Express-G com modelo de uma curva *B-Spline*. (ISO/IS 10303-238, 2006).

De acordo com a norma, uma curva *B-Spline* tem como atributos os valores dos nós e das multiplicidades dos nós, sendo obtido com estes dados o tamanho do vetor de nós. Além destes atributos também são herdados os seguintes atributos:

- nome da curva;
- grau da curva;
- pontos do polígono de controle;
- forma da curva;
- se é fechada ou aberta;
- se tem laços ou não.

Com estes atributos é possível ainda obter a quantidade de pontos do polígono de controle.

Deste modelo foi gerada uma descrição específica para a construção de uma classe com os atributos necessários para o envio das informações sobre uma curva *B-Spline*. Os dados da curva, obtidos do sistema CAD são inseridos no objeto curva *B-Spline* instanciado de acordo com a classe criada conforme acima. Os dados são então transcritos em linguagem XML e enviados para o local onde está também hospedado o Serviço Web.

A Figura 6.35 mostra um trecho de um arquivo em XML com os dados de uma curva *B-Spline*.

```
<?xml version="1.0"?>
-<b_spline_curve_with_knots xmlns=
  "http://bragecrim.net/b_spline_curve_with_knots.xsd">
  -<iso_10303_28_header>
  <time_stamp>18.08.2010 08:43:44</time_stamp>
  <organization>Unimep</organization>
  <documentation>Bragecrim</documentation>
  </iso_10303_28_header>
  -<knots_multiplicities>
    <list_of_integer> 6 </list_of_integer>
    <list_of_integer> 0 </list_of_integer>
  </knots_multiplicities>
  -<knots>
    <list_of_parameter_value> 0 </list_of_parameter_value>
  </knots>
  -<control_points_list>
    <coordinate_X> 0 </coordinate_X>
    <coordinate_Y> 50,2 </coordinate_Y>
    <coordinate_Z> 0,25 </coordinate_Z>
    <coordinate_X> 2,37592790692057 </coordinate_X>
    <coordinate_Y> 55,45 </coordinate_Y>
  </control_points_list>
  <closed_curve>false</closed_curve>
  <self_intersect>false</self_intersect>
  <degree> 5 </degree>
</b_spline_curve_with_knots>
```

Figura 6.35 Excerto de um arquivo XML com dados de uma curva *B-Spline*

Este arquivo em XML, com as informações da curva *B-Spline* é armazenado em servidor na Web, cujo acesso é franqueado ao serviço responsável pela definição dos pontos de medição.

Processamento pelo Serviço Web

Com as informações da curva armazenadas no mesmo local onde está disponível o serviço Web é feito o cálculo dos pontos a serem medidos e dos vetores de orientação e é gerado um arquivo no formato adequado à pronta utilização pela máquina de medir por coordenadas (MMC).

Com a informação da quantidade de pontos a serem gerados para medição, são calculados pontos para os parâmetros divididos igualmente ao longo da curva.

Determinado um parâmetro (p), são inicialmente calculadas as funções de suavização de primeira ordem para todos os nós, conforme algoritmo apresentado na Figura 6.36.

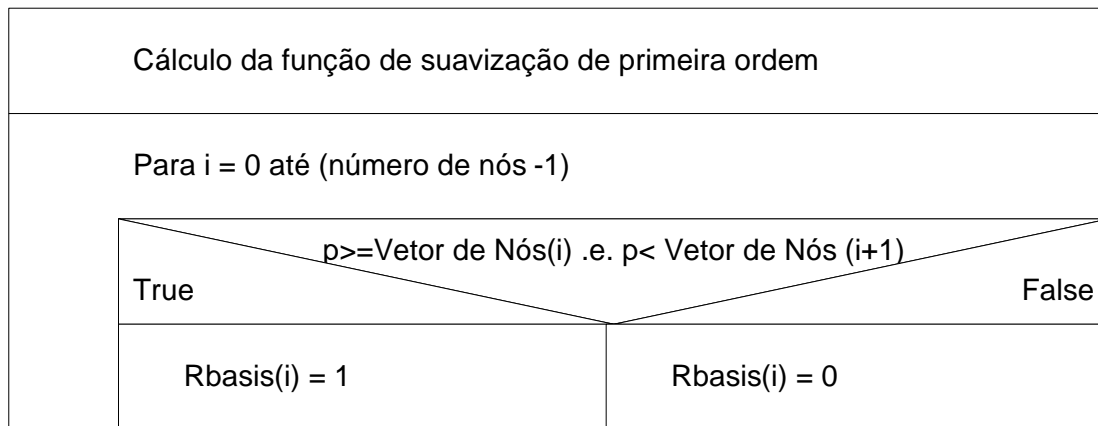


Figura 6.36 Estrutograma para cálculo da função de suavização de primeira ordem

Em seguida são calculadas as funções de suavização de segunda ordem e superiores, conforme algoritmo apresentado na Figura 6.37. Pode ser observado que a divisão do cálculo em dois termos deve-se à facilidade de compreensão e à possibilidade de otimização do algoritmo, eliminando-se a necessidade do cálculo de um dos termos se a função de suavização de ordem anterior for igual a zero. Esta otimização não é apresentada na figura.

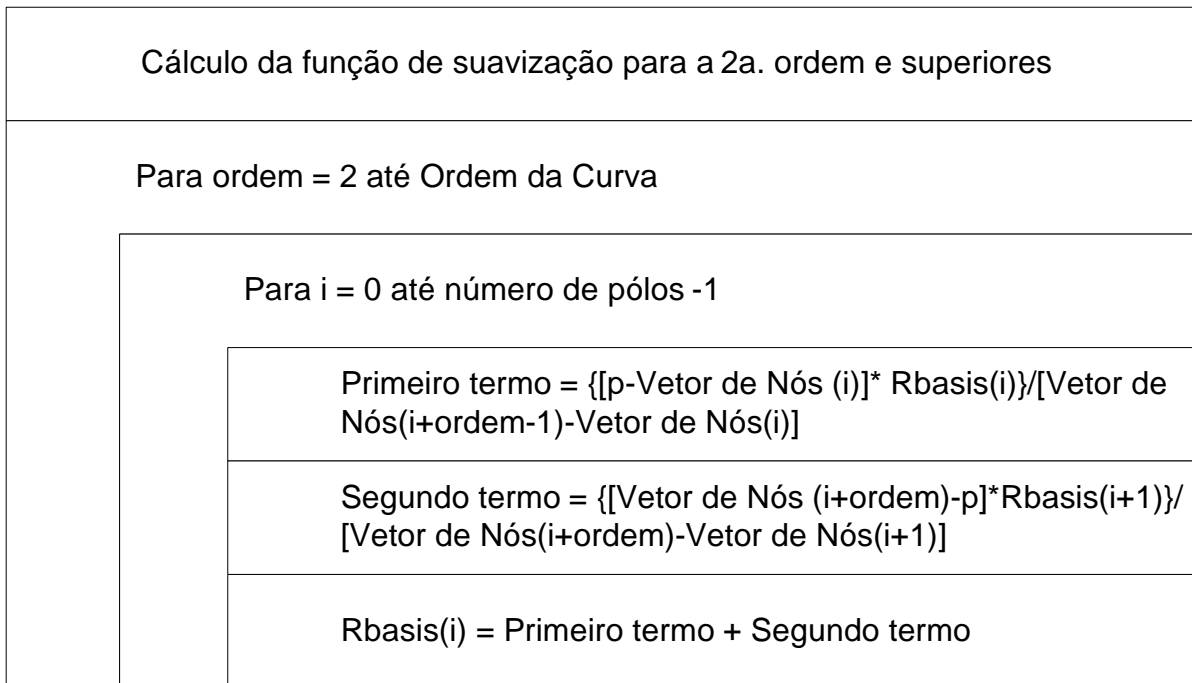


Figura 6.37 Estrutograma para cálculo das funções de suavização de segunda ordem e superiores.

De posse das funções de suavização é então calculado o ponto a ser medido de acordo com o parâmetro escolhido, isto é feito com a ponderação das funções de suavização pelos pontos do polígono de controle conforme apresentado na Figura 6.38, sobre esse ponto é calculado diretamente o vetor normal para indicar o posicionamento do apalpador da máquina de medir por coordenadas.

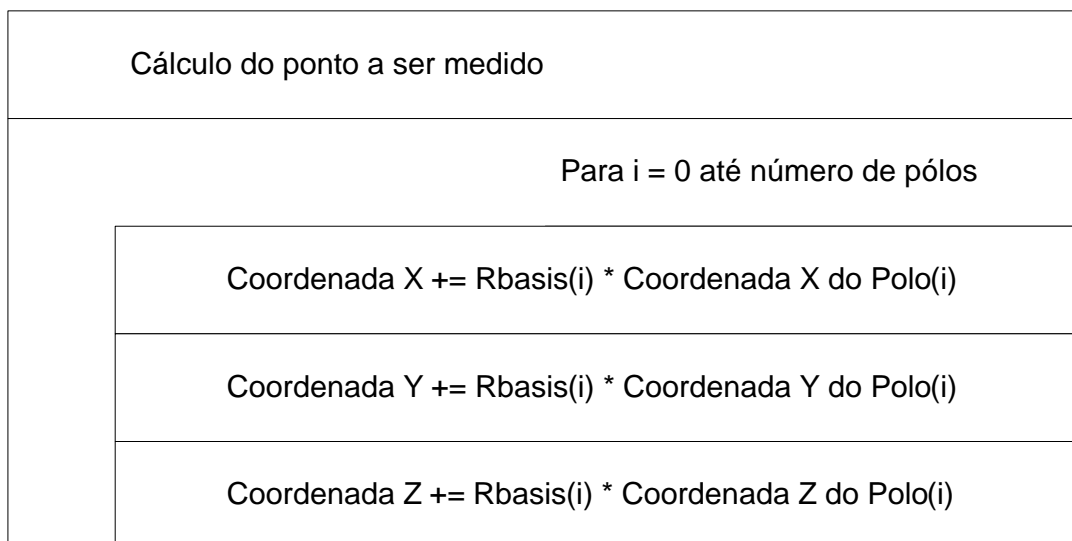


Figura 6.38 Estrutograma para cálculo do ponto a ser medido

Envio do arquivo de medição ao cliente CAI pelo Serviço Web

O sistema de medição por coordenadas dessa integração utiliza um arquivo texto para as informações dos pontos a serem medidos. Este arquivo tem uma sintaxe própria que deve ser respeitada, em que cada um dos pontos é considerado uma *feature* de medição e são dadas instruções para a medição em coordenadas cartesianas e também para o fim da medição.

O formato reconhecido pelo sistema CAI deve ter uma disposição onde, exista um nome referente a cada ponto gerado, seguido pela designação de seu modelo, ou seja, uma *feature* de formato ponto, indicado pela palavra-chave “F(PM###)”. Seguindo o seu conteúdo, existe a informação de que os dados estão dispostos em um modelo cartesiano, indicado por “FEAT/POINT,CART”, seguido de suas coordenadas x , y e z e os vetores i , j e k .

A próxima linha especifica o comando para que o ponto dado seja medido, e com quantos toques são necessários para fazê-lo, no caso de um ponto, apenas um toque é requerido, no exemplo apresentado isso é indicado por “MEAS/POINT, F(PM001),1”.

Por fim, se têm a informação de resultado do ponto, onde as coordenadas e vetores serão os mesmo que o nominal, pois eles ainda não foram inspecionados. Os dados relacionados à linha em questão, após importado para o sistema de medição e utilizados para a inspeção, serão alterados para o resultado obtido na medição daquele ponto. Já os dados existentes na primeira linha, serão considerados os valores nominais.

Nesta aplicação o Serviço Web armazena o arquivo de acordo com as prescrições exigidas para que seja feita a medição e disponibiliza para download, um exemplo do arquivo pode ser visto na Figura 6.39.

```

F (PM001)=FEAT/POINT,CART,3.677417,11.806711,75.000000,-
0.555380,0.831597,-0.000000
MEAS/POINT,F (PM001),1
PTMEAS/CART,3.677417,11.806711,75.000000,-0.555380,0.831597,-0.000000
ENDMES

F (PM002)=FEAT/POINT,CART,8.271700,14.679445,75.000000,-
0.530377,0.847762,0.000000
MEAS/POINT,F (PM002),1
PTMEAS/CART,8.271700,14.679445,75.000000,-0.530377,0.847762,0.000000
ENDMES Resultado da medição

F (PM003)=FEAT/POINT,CART,12.861420,17.559990,75.000000,-
0.532562,0.846391,-0.000000
MEAS/POINT,F (PM003),1
PTMEAS/CART,12.861420,17.559990,75.000000,-0.532562,0.846391,-0.000000
ENDMES

```

Coordenadas do ponto a ser medido

Coordenadas do vetor normal

6.5 Verificação, Validação e Teste dos aplicativos desenvolvidos.

Erros nos produtos de software podem ocorrer mesmo com a aplicação das técnicas e ferramentas adequadas ao seu desenvolvimento, para minimizar este problema foram desenvolvidas atividades de Verificação, Validação e Teste (VV&T), sendo que das três atividades a mais utilizada é a de teste de software, que consiste na análise dinâmica do software para identificação e posterior eliminação dos erros encontrados (MYERS, 1979).

A terminologia padrão, adotada pela IEEE (IEEE, 1990), diferencia três termos para problemas que podem ser encontrados em sistemas de informação:

- Defeito - Instrução ou comando incorreto;
- Erro – Discrepância entre o esperado e o ocorrido;
- Falha – Comportamento do software diferente do esperado, com a saída incorreta em relação ao especificado.

A Figura 6.40 mostra os domínios onde se encontram os problemas que possam existir em um sistema de informação. Defeitos são ocasionados por pessoas e correspondem a aplicação propriamente dita. Esses, por sua vez, podem causar erros, como por exemplo a construção de um software de modo diferente do que foi especificado. Finalmente, erros geram falhas, afetando diretamente o usuário final (ROCHA e MALDONADO, 2001).

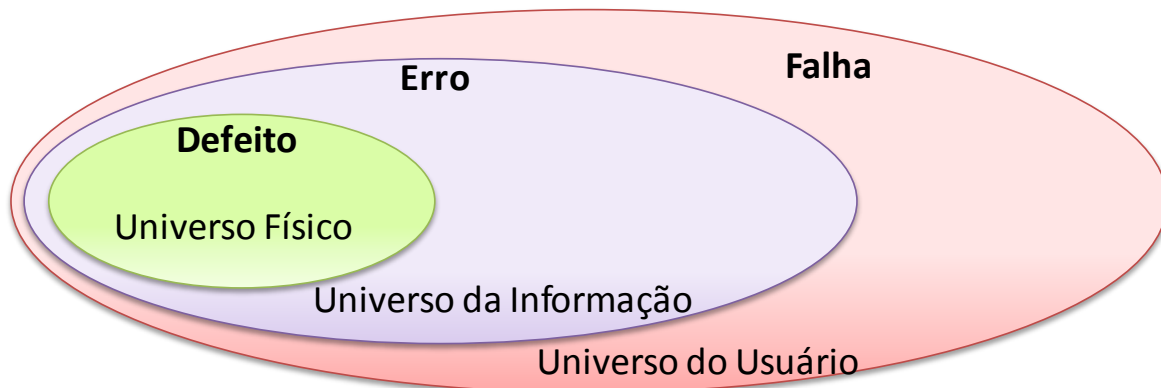


Figura 6.40 Defeito, Erro e Falha (ROCHA e MALDONADO, 2001)

Para a condução da atividade de teste e a consequente avaliação do software são utilizadas basicamente três técnicas, a funcional, a estrutural e a baseada em erros (MYERS, 1979).

O teste funcional trata o software como uma caixa com conteúdo desconhecido, sabe-se somente o que ocorre do lado externo, ou seja, as entradas e saídas, por essa característica este teste também é conhecido como teste da caixa-preta. Para o teste funcional são definidos dois passos principais: a identificação das funções que o software deve realizar e os casos que são capazes de checar se estas funções estão sendo executadas de forma correta (BEIZER, 1995).

A técnica estrutural avalia o comportamento interno do software, trabalhando diretamente com o código fonte considerando testes de condição; teste de fluxo de dados; teste de ciclos e teste de caminhos lógicos (PRESSMAN, 2006). Para este teste é analisado o código fonte e buscam-se casos que cubram as possibilidades de todas as variações existentes, em oposição ao teste funcional este teste é chamado de teste caixa-branca.

A técnica de teste baseada em erros considera os erros mais comuns no processo de desenvolvimento de software para daí definir os requisitos de teste. Essa técnica dá ênfase aos erros que o projetista pode cometer durante o desenvolvimento (MYERS, 1979).

Dentro do escopo dessa tese é necessário verificar o funcionamento correto dos aplicativos e dos serviços Web desenvolvidos, já que a proposta desse trabalho é o modelo de integração e não os aplicativos em si. Erros que possam existir nos softwares não devem invalidar o modelo proposto. A escolha recaiu sobre o teste funcional, portanto.

Como caso de validação foi escolhido acompanhar os dados até o aplicativo de integração, desde os aplicativos inseridos nos sistemas CAD e CAM até o aplicativo de integração. A igualdade dos dados mostra a funcionalidade dos componentes de software dentro da expectativa desse trabalho.

Para o caso em tela foram feitas comparações com três diferentes modelos durante as três fases do processo, analisando a coerência dos dados nos sistemas CAD e CAM, nos arquivos XML armazenados na Web e no aplicativo de integração.

Teste 1 – CAD

Para apreciação da persistência dos dados do sistema CAD foi utilizado o conjunto montado da garra que contém oito componentes, a árvore de construção, de acordo com o projeto está presente no sistema CAD, na janela “*Assembly Navigator*”, como mostrado na Figura 6.41.

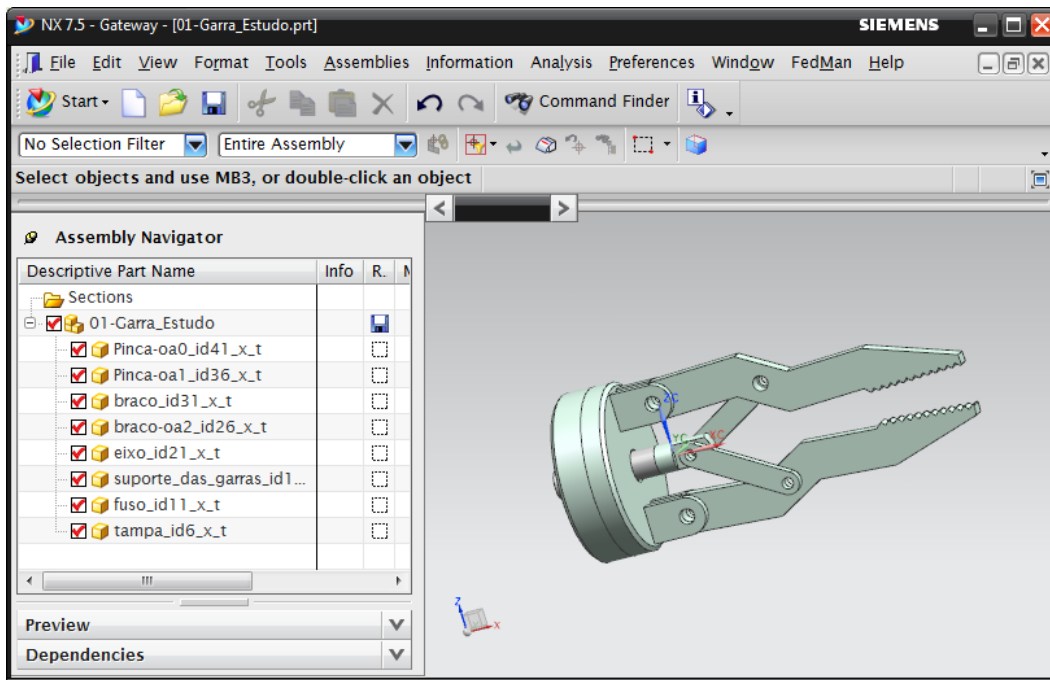


Figura 6.41 Conjunto montado de uma garra.

Para a demonstração das características foi escolhido o eixo central que, como mostra a caixa de diálogo da Figura 6.42, é feito com o material ABNT 1040.

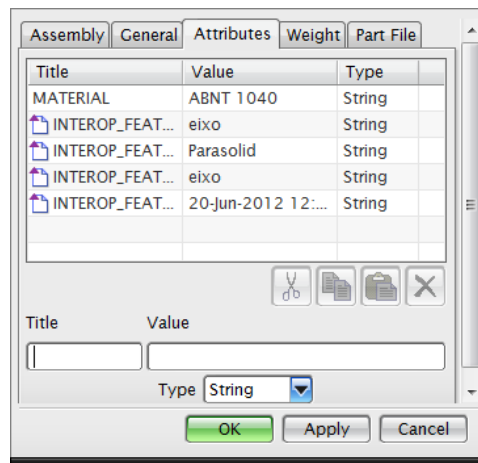


Figura 6.42. Caixa de diálogo com os atributos do componente eixo, existente no conjunto montado.

Os dados recolhidos do sistema CAD são enviados para a Web em um arquivo XML, a Figura 6.42 apresenta um trecho do arquivo XML gerado pelo aplicativo vinculado. Neste arquivo é possível verificar as informações dos componentes, em especial os dados do eixo central. Nota-se que a informação de material, assim como todas as outras, foi corretamente transferida para o arquivo XML.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
-<Assembly Name="">
  -<PartsList>
    -<Part Number="0">
      <name>PINCA-OA0_ID41_X_T</name>
      <quantity>1</quantity>
      <material>ABNT 1020</material>
      <locationX>32,84960610342</locationX>
      <locationY>4,48876233821293E-16</locationY>
      <locationZ>-9,1680758479106</locationZ>
      <type>NXOpen.Assemblies.Component</type>
      <owningPart>01-Garra_Estudo.prt</owningPart>
    </Part>
    (...)
    -<Part Number="4">
      <name>EIXO_ID21_X_T</name>
      <quantity>1</quantity>
      <material>ABNT 1040</material>
      <locationX>25,292748695223</locationX>
      <locationY>-0,014784384966398</locationY>
      <locationZ>-0,205609250207545</locationZ>
      <type>NXOpen.Assemblies.Component</type>
      <owningPart>01-Garra_Estudo.prt</owningPart>
    </Part>
    (...)
  </PartsList>
</Assembly>
```

Figura 6.43 Excerto de arquivo XML com dados do produto

A demonstração da integração nos três domínios é feita por um aplicativo que navega pelas informações de produto, processos e recursos, para o produto escolhido a tela da Figura 6.44 apresenta os oito componentes e para o componente “eixo” é apresentada a propriedade “material”. No caso de produto pode-se perceber a persistência dos dados durante todo o processo.

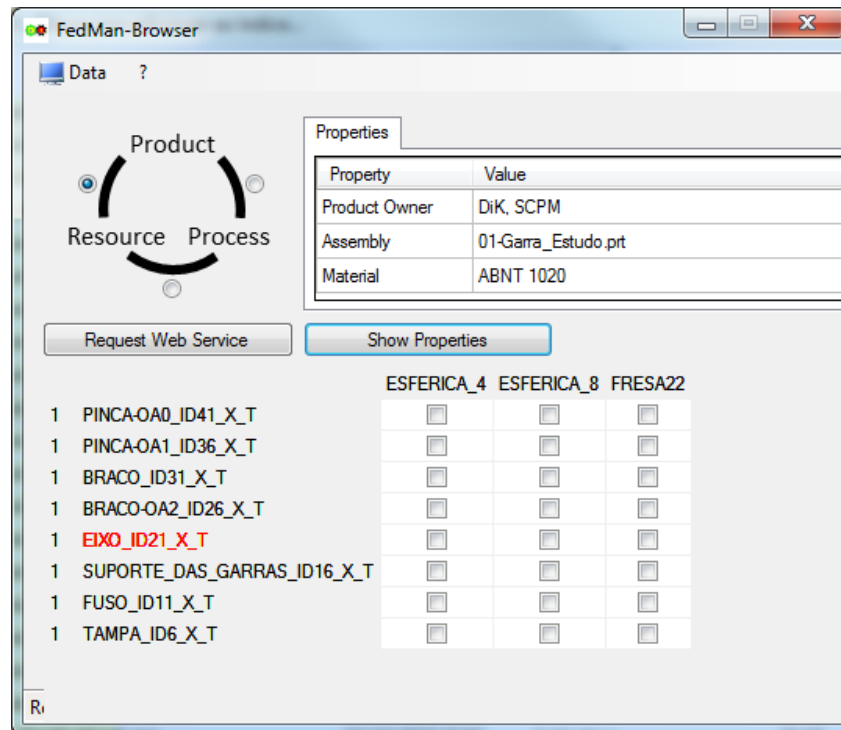


Figura 6.44 Tela do aplicativo com os dados de produto.

Teste 2 – CAM

As informações de manufatura e de ferramentas estão presentes no sistema CAM, no arquivo XML enviado à Web e no aplicativo de integração, para verificação destas condições foi utilizada a base do cilindro pneumático apresentado na Figura 6.45, que mostra as 17 operações previstas para a usinagem dessa peça. Nesse exemplo foi escolhida a operação de contorno, cujas propriedades, obtidas diretamente do sistema CAM são apresentadas na Figura 6.46, adicionalmente também são apresentadas as informações da ferramenta utilizada nessa operação.

Os dados coletados no sistema CAM, seja para as operações seja para as ferramentas, são convertidos em um arquivo XML e enviados para a Web. O arquivo para essa operação é apresentado na Figura 6.48.

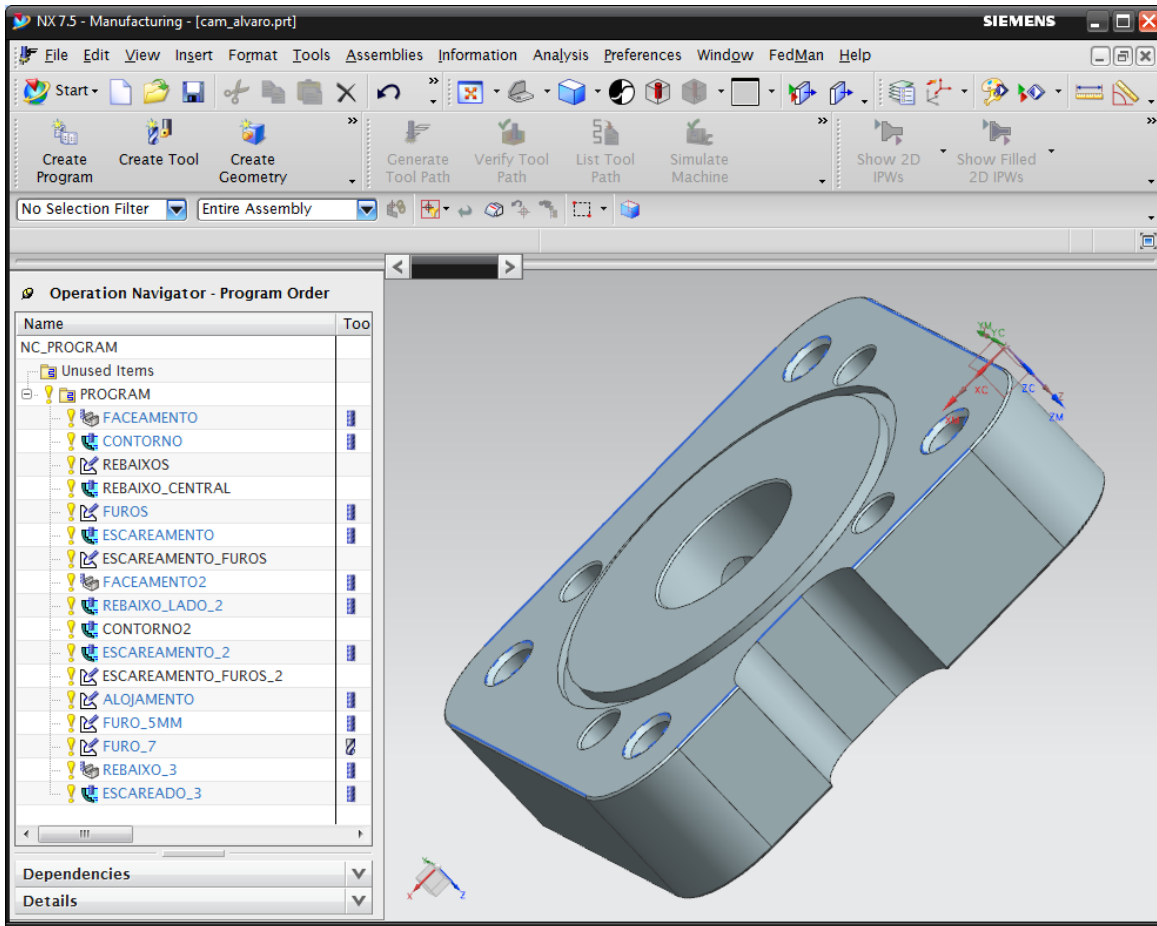


Figura 6.45 Base do cilindro pneumático com as operações de manufatura.

```

----- Object name: CONTORNO -----
Template Type: mill_contour
Template Subtype: CAVITY_MILL
Coordinate System of Data - Work
----- Operation Information -----
Operation Name          CONTORNO
Operation Type          Cavity Milling
Update Status           Need to Post
Last Time Updated       2012/06/11 14:25:24
Gouge Check Status      Not checked
Path Is Edited          No
Path Tag                 30274
In Process Workpiece    None
Order Group             PROGRAM
Method Group             MILL_FINISH
Geometry Group          REFERENCIA_1

```

Figura 6.46 Informações da operação de contorno.

```
-----Tool Information-----  
----- Object name: FRESA_TOPO_10MM -----  
Template Type: mill_planar  
Template Subtype: MILL  
Coordinate System of Data - Work  
Tool Material Name : HSS  
Material Description : High Speed Steel  
Tool Name : FRESA_TOPO_10MM  
Tool Type : Milling Tool-5 Parameters  
Holding System : not specified  
(D) Diameter = 10.000000000  
(R1) Lower Radius = 0.000000000  
(L) Length = 75.000000000  
(B) Taper Angle = 0.000000000  
(A) Tip Angle = 0.000000000  
(FL) Tool Flute Length = 50.000000000  
Number of Flute 2  
Direction = CLW  
Z Offset = Off  
Adjust Register 1  
Cutcom Register = Off  
Tool Number 2  
Catalog Number =  
Defined Holder Section Count: 0  
Defined Tracking Point Count: 0  
-----
```

Figura 6.47. Informações da ferramenta utilizada na operação de contorno.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
-<Product Name="">
  -<OperationList>
    +<Operation Name="FACEAMENTO" Type="FACE_MILLING">
    -<Operation Name="CONTORNO" Type="CAVITY_MILL">
      <diameter>10</diameter>
      <number>2</number>
      <length>75</length>
      <toolName>FRESA_TOPO_10MM</toolName>
    </Operation>
    +<Operation Name="REBAIXOS" Type="COUNTERBORING">
    +<Operation Name="REBAIXO_CENTRAL" Type="CAVITY_MILL">
    +<Operation Name="FUROS" Type="PECK_DRILLING">
    +<Operation Name="ESCAREAMENTO" Type="CAVITY_MILL">
    +<Operation Name="ESCAREAMENTO_FUROS" Type="COUNTERSINKING">
    +<Operation Name="FACEAMENTO2" Type="FACE_MILLING">
    +<Operation Name="REBAIXO_LADO_2" Type="CAVITY_MILL">
    +<Operation Name="CONTORNO2" Type="CAVITY_MILL">
    +<Operation Name="ESCAREAMENTO_2" Type="CAVITY_MILL">
    +<Operation Name="ESCAREAMENTO_FUROS_2" Type="COUNTERSINKING">
    +<Operation Name="ALOJAMENTO" Type="PECK_DRILLING">
    +<Operation Name="FURO_5MM" Type="PECK_DRILLING">
    +<Operation Name="FURO_7" Type="PECK_DRILLING">
    +<Operation Name="REBAIXO_3" Type="FACE_MILLING">
    +<Operation Name="ESCAREADO_3" Type="CAVITY_MILL">
  </OperationList>
</Product>
```

Figura 6.48 Excerto de arquivo XML com dados de manufatura

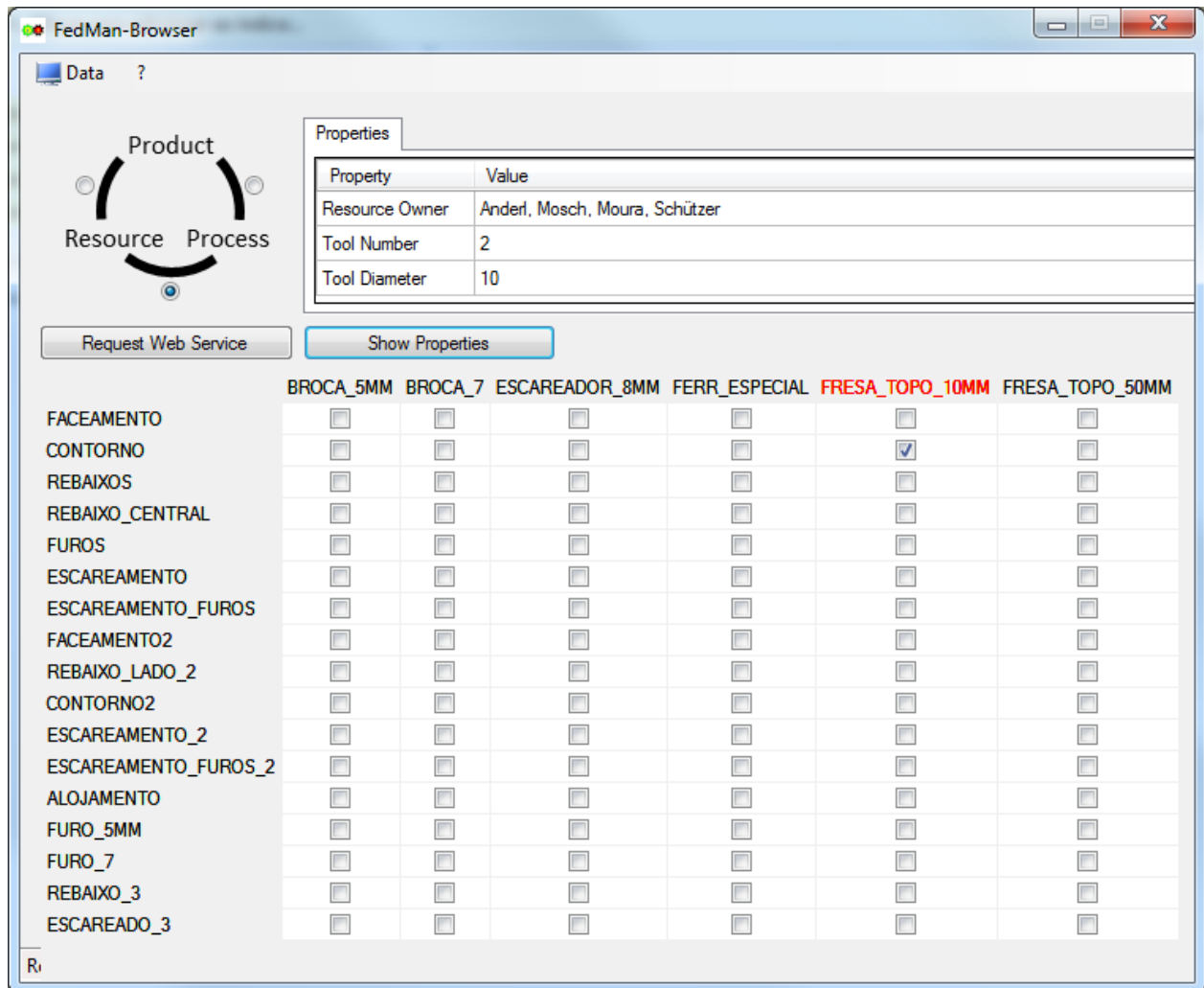


Figura 6.49 Tela do aplicativo com as operações e os dados da ferramenta.

Finalmente, os dados devem ser apresentados corretamente no aplicativo de integração. Nota-se que na figura Figura 6.49 há a persistência dos dados das 17 operações e também dos dados da ferramenta escolhida.

7 Conclusões

Este capítulo apresenta as conclusões desta tese e as sugestões de outros trabalhos que possam advir. Vale ressaltar que este trabalho objetiva uma proposta de integração das informações de um sistema produtivo, para isso estabelecendo relações causais lineares obtidas por um processo de análise. Essa proposta utiliza um modelo de informação baseado na decomposição do sistema de manufatura em três domínios, representados por eixos ortogonais e como tecnologia de suporte a arquitetura orientada a serviços, o uso de serviços Web e bancos de dados federativos. Não faz parte do escopo deste trabalho o desenvolvimento de um sistema de informações completo para a organização do sistema de manufatura, mas sim a propositura de um modelo para esta organização.

O processo de análise do sistema produtivo, partindo-se da divisão em três domínios distintos possibilitou a criação de um modelo central hipotético, com a capacidade de agregar o volume de informações heterogêneas que podem pertencer a um sistema de manufatura. A ortogonalidade dos domínios facilita a compreensão do todo ao possibilitar a visão das relações sempre duas a duas.

Com a redução em componentes menores foi possível atender aos requisitos de autonomia e flexibilidade dos atores que compõem o sistema, sem a necessidade de se caminhar para as soluções propostas dentro do contexto da teoria da complexidade.

O uso de uma arquitetura orientada a serviços se mostrou plenamente aderente à proposta inicial e, em conjunto com os serviços Web, permitiu o acesso a bancos de dados federativos o que manteve a autonomia de cada um dos participantes sem que houvesse perda da integração das informações.

Para que seja atendida a necessidade de distribuição são usados bancos de dados federativos, em que cada etapa ou usuário tem a possibilidade de gerar e gerenciar de maneira autônoma os dados relativos à sua atividade.

Dois requisitos são apresentados na literatura como essenciais para o gerenciamento de um sistema de manufatura, quais sejam: distribuição e colaboração. O requisito de

distribuição é atendido pelo uso de bancos de dados federativos acessíveis pela Web; o requisito de colaboração é atendido pelo uso de arquivos baseados em sistemas neutros e descritos em linguagem extensível, permitindo o acesso independente da plataforma de software; sistema operacional ou aplicativo. No desenvolvimento deste trabalho esses requisitos foram verificados informalmente pela colaboração entre as equipes alemãs e brasileiras (cf. Apêndice B).

A centralização apresentada como solução comercial para sistemas de manufatura pressupõe a garantia de integração completa das informações, o que nem sempre ocorre, especialmente no que concerne a sistemas legados. Por outro lado é patente a lacuna no atendimento à flexibilidade do sistema de informação, sendo muitas vezes necessárias adaptações nem sempre simples e muitas vezes onerosas.

A proposta aqui apresentada garante a integração, pois os dados ficam disponíveis na Web em um formato neutro e garante a flexibilidade por permitir o ajuste observando serviço a serviço, sem a necessidade de uma mudança no sistema como um todo. Aos sistemas legados é franqueado o acesso pela construção de serviços específicos que podem fazer as adaptações necessárias e disponibilizar os dados para os demais atores do processo.

7.1 Sugestões de Trabalhos Futuros

Outros trabalhos poderão ser iniciados a partir da proposta aqui apresentada para ampliar o seu escopo e outros poderão agregar os conceitos de integração aqui mostrados a outros sistemas.

As questões relativas à segurança da informação não foram objeto deste trabalho e é inegável que por estar baseado em um sistema distribuído e que permite o acesso a várias pessoas é preciso cuidar para que não haja uso indevido das informações e também preservação da propriedade intelectual.

Um acréscimo significativo pode ser obtido ao incluir uma semântica às informações trocadas, com isso pode ser possível o uso de máquinas de inferência para a utilização dos dados o que aumentaria a flexibilidade do sistema.

O crescente uso de dispositivos móveis, sejam celulares inteligentes, *tablets* e outros, aponta um caminho para incrementar a característica de distribuição desta proposta. Facilitar o acesso de modo a ganhar mobilidade é um caminho que se apresenta para desenvolvimentos futuros.

A criação de uma biblioteca de serviços pertinentes à manufatura possibilitará uma gama de escolhas para os seus usuários. Para que um serviço esteja disponível é preciso também que haja a sua exposição, de tal maneira que o usuário final tome conhecimento e se utilize do serviço. A existência de uma biblioteca comum facilitaria ao usuário encontrar o serviço desejado.

Posto que este trabalho não seja uma solução baseada na Teoria da Complexidade é possível que trabalhos futuros busquem uma integração da proposta baseada na análise racional com alguma proposta complexa, obtendo as vantagens das duas abordagens.

Referências Bibliográficas

ABBAS, R. **Manufacturing Semantic Interoperability for a SOA Adaptation Strategy.**

Disponível em: <<http://soa.sys-con.com/node/565632>>. Acesso em: 19 abr. 2010.

ABELN, O. **Innovationspotentiale in der Produktentwicklung: Das CAD-Referenzmodell in der Praxis.** Stuttgart: Teubner Verlag, 1997.

ABRAMOVICI, M.; BELLALOUNA, F. **SOA-Architecture for the Integration of Domain-Specific PLM Systems within the Mechatronic Product Development.** 7th International Symposium on Tools and Methods of Competitive Engineering, TMCE. **Proceedings...** Izmir, 2008

ACHTEN, H. A.; LEEUWEN, J. P. V. **Feature-Based High Level Design Tools: A Classification.** (G. Augenbroe & C. Eastman, Eds.) Proceedings of the 8th International Conference on Computer Aided Architectural Design Futures. **Proceedings...** Atlanta, 1999

ALVARES, A. J.; FERREIRA, J. C. E. **Uma metodologia para integração cad/capp/cam voltada para manufatura remota de peças rotacionais via web.** Congresso Brasileiro de Engenharia de Fabricação. **Anais...** Uberlândia: ABCM. , 2003

ANDERL, R. **Visualização Colaborativa do Produto.** (K. Schützer, Ed.) 11o Seminário Internacional de Alta Tecnologia - Inovações Tecnológicas no Desenvolvimento do Produto. **Anais...** Piracicaba: Unimep. , 2006

ANDERL, R.; MALZACHER, J.; RASSLE, J. Proposal for an Object Oriented Process Modeling Language 2 Existing Process Modeling Languages. In: MERTINS, K.; RUGGABER, R.; POPPLEWELL, K.; XU, X. (Eds.). **Enterprise Interoperability III:** Springer London, 2008. p. 533-545.

ANDERL, R.; REZAEI, M.; SCHÜTZER, K.; ASSIS MOURA, A. Á. DE. A Federative Factory Data Management Approach for the Integration of Product and Process Models. **ProduktDaten Journal**, n. 2, p. 49-53, 2009.

ANDERL, R.; SCHÜTZER, K.; MOSCH, C.; ASSIS MOURA, A. Á. DE. Föderatives Fabrikdatenmanagement - Fabrikplanungsprozesse basierend auf Web Services. **Industrie Management**, v. 27, p. 69-72, 2011.

ANDERL, R.; VÖLZ, D.; SCHÜLE, A. **An Approach to use Semantic Annotations in Global Product Development to Bridge the Gap in Interdisciplinary and intercultural Communication.** WMSCI 2010. **Proceedings...** Orland, Florida, USA. Disponível em: <<http://tubiblio.ulb.tu-darmstadt.de/48406/>>, 2010

ANDROUTSELLIS-THEOTOKIS, S.; SPINELLIS, D. A survey of peer-to-peer content distribution technologies. **ACM Computing Surveys**, v. 36, n. 4, p. 335-371, 1 dez 2004.

ANTUNES JR., J. A. V. **Em direção a uma teoria geral do processo na administração da produção: uma discussão sobre a possibilidade de unificação da teoria das restrições e**

da teoria que sustenta a construção dos sistemas de produção com estoque zero. Porto Alegre: Universidade Federal do Rio Grande do Sul, 1998. (Tese de Doutorado)

ARAUJO, O. F. N. D. **Proposta de uma rede de compartilhamento de habilidades no ambiente da Manufatura.** Campinas: Universidade Estadual de Campinas, 2003. (Tese de Doutorado)

ARAÚJO JÚNIOR, L. O. **Método de Programação de Sistemas de Manufatura do Tipo Job Shop Dinâmico não Determinístico.** São Paulo: Universidade de São Paulo, 2006. (Tese de Doutorado)

BANDARA, W.; TAN, H. M.; RECKER, J.; INDULSKA, M.; ROSEMANN, M. **Bibliography of process modeling: An Emerging research field.** Queensland University of Technology. Disponível em: <http://eprints.qut.edu.au/archive/00008745/>. Acessado em 19/7/2012.

BARTON, J. A.; LOVE, D. M.; TAYLOR, G. D. Design determines 70% of cost? A review of implications for design evaluation. **Journal of Engineering Design**, v. 12, n. 1, p. 47-58, mar 2001.

BAUER, A.; BOWDEN, R.; BROWNE, J.; DUGGAN, J.; LYONS, G. **Shop Floor Control Systems - from Design to Implementation.** London: Chapman & Hall, 1991. p. 345

BEIZER, B. **Black-Box Testing: Techniques for Functional Testing of Software.** 1st. ed. New York: John Wiley & Sons, 1995. p. 294

BENAVENTE, J. C. T. **Um sistema para o projeto e fabricação remota de peças prismáticas via Internet.** Florianópolis: Universidade Federal de Santa Catarina, 2007. (Dissertação de Mestrado)

BESSEN, F.; PRIM, C. H.; MACHADO, C. D. R.; PIACENTINI, A. **Sistema de gestão da inovação tecnológica para institutos promotores de ambientes de inovação.** (ABEPRO, Ed.)XVI Simposio de Engenharia de Producao. **Anais...** Bauru: 2009

BIN, B.; LEE, I. **An Enterprise-wide Process Centric Framework for Manufacturing Information Technologies.** . Singapore: Singapore Institut of Manufacturing Technology - Technical Report STR/03/060/PDD , 2003

BIRECREE, A.; KONZELMANN, S.; WILKINSON, F. Productive Systems, Competitive Pressure, Strategic Choices and Work Organisation: An introduction. **1st. International Contributions to Labour Studies**, v. 7, Special Issue on Co-operative and Antagonistic Work Organisation 1997.

BLACK, A. P.; HUANG, J.; KOSTER, R.; WALPOLE, J.; PU, C. Infopipes: An abstraction for multimedia streaming. **Multimedia Systems**, v. 8, n. 5, p. 406-419, 1 dez 2002.

BLECKER, T.; GRAF, G. **Multi-Agent Systems in Internet based Production Environments — an enabling Infrastructure for Mass Customization.** (F. T. Piller, R. Reichwald, & M. Tseng, Eds.) Competitive Advantage Trough Customer Interaction: Leading Mass Customization and Personalization from Emerging State to Mainstream Business Model **Proceedings...** München: 2th. , 2003

BOER, W. **A gestão da produtividade na manufatura como fator indutivo na formação do valor agregado de produtos e serviços: um exemplo de aplicação na indústria aeronáutica.** Bauru: Universidade Estadual Paulista, 2008. (Dissertação de Mestrado)

BOOTH, D. **Web Services Architecture.** Disponível em: <[http://www.w3.org/TR/2004/NOTE-
ws-arch-20040211/](http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/)>. Acesso em: 19 abr. 2010.

BORGATTI NETO, R. **Perspectivas da complexidade aplicadas à gestão de empresas.** São Paulo: Universidade de São Paulo, 2007. (Tese de Doutorado)

BORSATO, M. **Uma plataforma de suporte ao gerenciamento do desenvolvimento rápido de produtos tecnológicos através da engenharia simultanea.** Florianópolis: Universidade Federal de Santa Catarina, 2003. (Tese de Doutorado)

BOSCARIOLI, C.; BEZERRA, A.; BENEDICTO, M. D.; DELMIRO, G. **Uma reflexão sobre Banco de Dados Orientados a Objetos.** Congresso de Tecnologias para Gestão de Dados e Metadados do Cone Sul. **Anais...** Ponta Grossa: UEPG. , 2006

BOYD, A.; NOLLER, D.; PETERS, P. *et al.* **SOA in Manufacturing Guidebook.** . Chandler, AZ: MESA International. , 2008

BRANDAO, R.; WYNN, M. **Product Lifecycle Management Systems and Business Process Improvement – A Report on Case Study Research.** 2008 The Third International Multi-Conference on Computing in the Global Information Technology (iccgi 2008). **Proceedings...** Los Alamitos: IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4591355>>. Acesso em: 27 jul. 2012

BROWN, J. **Tech-Clarity: The Five Dimensions of Product Complexity. Complexity.** Tech-Clarity Issue in Focus: Tech-Clarity, Inc. , 2010

CAREY, M. J.; DEWITT, D. J. **Of Objects and Databases : A Decade of Turmoil.** (T. M. Vijayaraman, A. P. Buchmann, C. Mohan, & N. L. Sarda, Eds.) VLDB '96 Proceedings of the 22th International Conference on Very Large Data. **Proceedings...** Mumbai (Bombay): Morgan Kaufmann. , 1996

CHANDRA, C.; KAMRANI, A. Knowledge management for consumer-focused product design. **Journal of Intelligent Manufacturing**, v. 14, p. 557-580, 2003.

CHAO, P.-YI; WANG, Y.-CHOU. A data exchange framework for networked CAD / CAM. **Computers in Industry**, v. 44, n. August 1999, p. 131-140, 2001.

CHENGYING, L. Research on manufacturing resource modeling based on the O–O method. **Journal of Materials Processing Technology**, v. 139, n. 1-3, p. 40-43, ago 2003.

CHINNICI, R.; MOREAU, J.-J.; RYMAN, A.; WEERAWARANA, S. **Web Services Description Language (WSDL).** Disponível em: <<http://www.w3.org/TR/wsdl20/>>. Acesso em: 19 jan. 2012.

CHRYSSOLOURIS, G.; MAVRIKIOS, D.; PAPAKOSTAS, N. *et al.* Digital manufacturing: history, perspectives, and outlook. **Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture**, v. 223, n. 5, p. 451-462, 1 maio 2009.

CHUNG, S.; SYNDER, C. **ERP Initiation - A Historical Perspective**. Americas Conference on Information Systems (AMCIS). **Proceedings**.... Disponível em: <<http://aisel.aisnet.org/amcis1999/76>>, 1999

CODD, E. F. A relational model of data for large shared data banks. 1970. **M.D. computing : computers in medical practice**, v. 15, n. 3, p. 162-6, 1970.

COLLINS. **Collins English Dictionary: 30th Anniversary Edition**. 30th Anniv ed. London: HarperCollins UK; 30 Rep Anv edition, 2010. p. 1920

DAY, M. **Is PLM the new PDM?** Disponível em: <http://www.caddigest.com/subjects/PLM/select/010704_day_pdm.htm>. Acesso em: 21 jul. 2012.

DEBONI, J. E. Z.; MARTINI, J. S. C. **Um método para desenvolvimento de sistemas de apoio a projetos de engenharia**. . São Paulo: EPUSP. , 1997

DER AALST, W. M. P. VAN; TER HOFSTEDE, A. H. M.; WESKE, M. Business Process Management : A Survey. In: WESKE, MATHIAS (Ed.). **Business Process Management**. Lecture No ed. Berlin: Springer Berlin / Heidelberg, 2003. p. 1019-1019.

DIAS, A. V. C. **Produto mundial, engenharia brasileira: integração de subsidiárias no desenvolvimento de produtos globais no setor automotivo**. São Paulo: Universidade de São Paulo, 2003. (Tese de Doutorado)

DUMAS, M.; DER AALST, W. M. P. VAN; HOFSTEDE, H. M. TER. **PROCESS-AWARE INFORMATION SYSTEMS Bridging People and Software Through Process Technology**. Hoboken, New Jersey: John Wiley & Sons, 2005. p. 409

EIGNER, M. **Industry Based Research for the Holistic Optimization of Product Creation**. (K. Schützer, Ed.) 13o. Seminário Internacional de Alta Tecnologia. **Anais**... Piracicaba: Universidade Metodista de Piracicaba. 2008

EIGNER, M. On the Development of New Modeling Concepts for Product Lifecycle Management in Engineering Enterprises. **Computer-Aided Design and Applications**, v. 7, n. 2, p. 203-212, 1 abr 2010.

EIGNER, M.; STELZER, R. Produktdaten-Management und Product Lifecycle Management. **Product Lifecycle Management**. Berlin Heidelberg: Springer Berlin Heidelberg, 2009. p. 27-45.

FARIN, G. A History of Curves and Surfaces in CAGD. In: FARIN, G.; HOSCHED, J.; KIM, M.-S. (Eds.). **Handbook of Computer Aided Geometric Design**. Amsterdam, The Netherlands: Elsevier Science Publishers, 1984. p. 1-23.

FELICIANO, R. A. **Uma proposta de gerenciamento integrado da demanda e distribuição, utilizando sistema de apoio à decisão (sad) com business intelligence (bi)**. São Paulo: Universidade de São Paulo, 2009. (Dissertação de Mestrado)

FILETO, R. **A abordagem POESIA para a integração de dados e serviços na Web Semantica**. Campinas: Universidade Estadual de Campinas, 2003. (Tese de Doutorado)

FILETO, R., MEDEIROS, C. M. B. **A Survey on Information Systems Interoperability**, Technical report – IC-03-030, Dec.2003.

FITSILIS, P.; GEROGIANNIS, V.; KAMEAS, A.; PAVLIDES, G. **Producing relational database schemata from an object oriented design**. Proceedings of Twentieth Euromicro Conference. System Architecture and Integration. **Proceedings...** IEEE Comput. Soc. Press. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=390386>>. Acesso em: 27 jul. 2012. , 1994

FONG, J. **Information Systems Reengineering and Integration**. 2nd. ed. London: Springer London, 2006. p. 383

FRANCO, G. N. **Aplicação de Sistemas Holônicos à Manufatura Inteligente Aplicação de Sistemas Holônicos à Manufatura Inteligente**. Campinas: Universidade Estadual de Campinas, 2003. (Tese de Doutorado)

GARLAN, D.; MONROE, R.; WILE, D.; Acme: An architecture Description Interchange Language. In: Johnson Howard, eds. **CASCON '97 Proceedings fo the 1997 conference of the Centre for Advanced Studies on Collaborative Research**. Canada; 1997:1-15.

GIUGLIANI, E.; VARVAKIS, G. **Gestão do Conhecimento através do Mapeamento de Processos em Empresas Intensivas em Conhecimento**. XXVII Encontro Nacional de Engenharia de Producao. **Anais...** Foz do Iguacu, 2007

GRABOWSKI, H.; MEIS, E. **Using ontologies for the Integrated Product Model Development**. (A. Farquhar & M. Gruninger, Eds.) AAAI Spring Symposium. **Proceedings...** Stanford: Association for the Advancement of Artificial Intelligence. Disponível em: <www.aaai.org>. , 1997

GROOVER, M. P.; EMORY W. ZIMERS JR. **CAD/CAM: Computer-Aided Design and Manufacturing**. Englewood Cliffs, New Jersey: Prentice-Hall, 1983. p. 512

GUDGIN, M.; HADLEY, M.; MENDELSON, N. *et al.* **SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)**. Disponível em: <<http://www.w3.org/2000/xp/Group/2/06/LC/soap12-part1.xml>>. Acesso em: 19 jan. 2012.

GUJARATHI, G. P.; MA, Y.-S. Parametric CAD/CAE integration using a common data model. **Journal of Manufacturing Systems**, v. 30, n. 3, p. 118-132, ago 2011.

HALTTUNEN, V.; TUIKKA, T. **Augmenting virtual prototyping with physical objects**. Proceedings of the working conference on Advanced visual interfaces - AVI '00. **Proceedings...** New York, New York, USA: ACM Press. Disponível em: <<http://portal.acm.org/citation.cfm?doid=345513.345363>>. Acesso em: 27 jul. 2012. , 2000

HANS-ERIK, E.; PENKER, M. **Business Modeling with UML: Business Patterns as Work**. New York: John Wiley & Sons, 2000. p. 459

HANSEN, Z. N. L.; KRISTENSEN, S. A. Global product development: the impact on the product development process and how companies deal with it. **International Journal of Product Development**, v. 15, n. 4, p. 205, 2011.

HARDING, J. A.; SHAHBAZ, M.; KUSIAK, A. Data Mining in Manufacturing: A Review. **Journal of Manufacturing Science and Engineering**, v. 128, n. 4, p. 969, 2006.

HELLENO, A. L. **Contribuição para a Manufatura de Superfícies Complexas com Altas Velocidades Baseadas em Novos Métodos de Interpolação da Trajetória da Ferramenta**. Santa Bárbara d'Oeste: Universidade Metodista de Piracicaba, 2008. (Tese de Doutorado)

HEO, E.-Y.; KIM, D.-W.; KIM, B.-H.; CHEN, F. F. Estimation of NC machining time using NC block distribution for sculptured surface machining. **Robotics and Computer-Integrated Manufacturing**, v. 22, n. 5-6, p. 437-446, out 2006.

HYPOLITO, C. M.; PAMPLONA, E. D. O. **Sistemas de Gestão Integrada: Conceitos e Considerações em uma Implantação**. XIX Enegep. **Anais...** Rio de Janeiro: ABEPRO. , 1999

HÄRDER, T.; SAUTER, G.; THOMAS, J. The intrinsic problems of structural heterogeneity and an approach to their solution. **The VLDB Journal The International Journal on Very Large Data Bases**, v. 8, n. 1, p. 25-43, 1 abr 1999.

IEEE. **IEEE Standard Glossary of Software Engineering Terminology - Standard 610.12**. [S.l.]: IEEE Comput. Soc. Press, 1990.

ISO/IS 10303-238. **Industrial automation systems and integration — Product data representation and exchange — Part 238: Application protocol: Application interpreted model for computerized numerical controllers**. Troy: Loffredo, Dave. , 2006

JAGENBERG, J. T.; GILSDORF, E. A.; ANDERL, R.; BORNKESSEL, T. **DETC2009-86063**. ASME 2009 International Design Engineering Technical Conferences. **Proceedings...** San Diego: ASME. , 2009

JONG, W.-REN; LIN, T.-CHUN; LAI, P.-JUNG; *et al.* **Integration of Mold Design and Mold Manufacturing**. ANTEC 2009 Plastics: Annual Technical Conference Proceedings. **Proceedings...** Chigado, IL: Society of Plastics Engineers. Disponível em: <http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=2647>. , 2009

JOSUTTIS, N. M. **SOA na Prática**. 1a. ed. Rio de Janeiro: Alta Books, 2008. p. 280

KEMPER, A.; LOCKERMANN, P. C.; MOERKOTTE, G.; WALTER, H. D.; LANG, S. M. **Autonomy over Ubiquity: Coping with the Complexity of a Distributed World**. International Conference Entity-Relationship Approach. **Proceedings...** Lausanne, 1990

KERN, V. M.; BØHN, J. H. **STEP databases for product data exchange**. I International Congress of Industrial Engineering. **Proceedings...** São Carlos, 1997

- KHOSHAFIAN, S. **Banco de Dados Orientado a Objetos**. Rio de Janeiro: Infobook, 1994.
- KIM, G. Y.; NOH, S. D.; RIM, Y. H.; MUN, J. H. XML-based concurrent and integrated ergonomic analysis in PLM. **The International Journal of Advanced Manufacturing Technology**, v. 39, n. 9-10, p. 1045-1060, fev 2008.
- KIM, W. **Research directions in object-oriented database systems**. Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS '90. **Proceedings...** New York, New York, USA: ACM Press. Disponível em: <<http://portal.acm.org/citation.cfm?doid=298514.298537>>. Acesso em: 19 jan. 2012, 1990
- KLEIN, M. Supporting conflict resolution in cooperative design systems. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 21, n. 6, p. 1379-1390, 1991.
- KOCHAN, D. **Integration of CAD/CAM**. Amsterdam: North-Holland, 1984. p. 284
- KOTLER, P. **Princípios de Marketing**. São Paulo: Pearson Frentic, 2003. p. 593
- KOZACZYNSKI, W.; LILIEN, L. **An Extended Entity-Relationship (E2R) database specification and its automatic verification and transformation into the logical relational design**. (S. T. March, Ed.) Sixth internat. Conf. on E-R Approach. **Proceedings...** New York, New York, USA: Entity-Relationship Approach, Elsevier Science Publishers, 1987
- KRAFZIG, D.; BANKE, K.; SLAMA, D. **Enterprise SOA Service-Oriented Architecture: Best Practices**. 1st. ed. New Jersey: Pearson Education, 2005.
- KRAUSE, F.-L.; KIMURA, F.; KJELLBERG, T.; LU, S. C.-Y. **Product Modelling**. Annals of the CIRP, v.42 n.2, 695-706,1993
- KULVATUNYOU, B.; WYSK, R.; CHO, H.; JONES, A. Integration framework of process planning based on resource independent operation summary to support collaborative manufacturing. **International Journal of Computer Integrated Manufacturing**, v. 17, n. 5, p. 377-393, jul 2004.
- KURTZ, R. B. **Toward a New Era in U.S. Manufacturing: The Need for a National Vision**. Washington: Natl Academy Pr, 1986. p. 174
- LAI-YUEN, S. K.; LEE, Y.-S. **Turn-Mill Tool Path Planning and Manufacturing Cost Analysis for Complex Parts Machining**. Industrial Engineering Research (IERC) Conference. **Proceedings...** Orlando, FL: , 2002
- LAURINDO, F. J. B.; MESQUITA, M. A. D. Material Requirements Planning: 25 Anos de Historia - Uma Revisão do Passado e Prospecção do Futuro. **Gestão & Produção**, v. 7, n. 3, p. 320-337, ago 2000.
- LEE, H.-J.; LEE, J.-W.; LEE, J.-O. Development of Web services-based Multidisciplinary Design Optimization framework. **Advances in Engineering Software**, v. 40, n. 3, p. 176-183, mar 2009.

LENTNER, M.; KOWALSKI, J. Odra : a next – generation object – oriented environment for rapid database application development. Motivation , general assumptions and architecture. In: IOANNIDIS, Y.; NOVIKOV, B.; RACHEV, B. (Eds.). **Advances in Databases and Information Systems**. Berlin / Heidelberg: Springer Berlin / Heidelberg, 2007. p. 130-140.

LITWIN, W.; MARK, L.; ROUSSOPOULOS, N. Interoperability of multiple autonomous databases. **ACM Computing Surveys**, v. 22, n. 3, p. 267-293, set 1990.

LUBELL, J.; PEAK, R. S.; SRINIVASAN, V.; WATERBURY, S. C. STEP, XML, and UML: Complementary Technologies. **Volume 4: 24th Computers and Information in Engineering Conference**, p. 915-923, 2004.

MACHADO, M. C. **Princípios enxutos no Processo de Desenvolvimento de Produtos: Proposta de uma Metodologia para Implementação**. São Paulo: Universidade de São Paulo, 2006. (Tese de Doutorado)

MAROPOULOS, P. Integration of tool selection with design Part 2: Aggregate machining time estimation. **Journal of Materials Processing Technology**, v. 107, n. 1-3, p. 135-142, nov 2000.

MASSON, C. **SOA can light up manufacturing IT**. Disponível em: <<http://www.computerweekly.com/Articles/2006/11/09/219795/soa-can-light-up-manufacturing-it.htm>>. Acesso em: 27 jul. 2012.

MCCARTHY, J. **Marketing Essencial: Uma abordagem gerencial e global**. São Paulo: Atlas, 1997. p. 297

MCMAHON, C. **CAD/CAM Principles, Practice and Manufacturing Management**. Harlow: Addison-Wesley, 1998. p. 665

MERRIAM-WEBSTER. **Merriam-Webster's Collegiate Dictionary**. 11th. ed. Springfield: Merriam-Webster, Inc., 2008. p. 1664

MIDLER, C. **El Auto Que No Existia**. Buenos Aires: Ediciones Fadu, 2005. p. 319

MIKOS, W. L. **Modelo baseado em agentes em apoio à solução de problemas de não conformidade em ambientes de manufatura com recursos distribuídos**. Florianópolis: Universidade Federal de Santa Catarina, 2008. (Tese de Doutorado)

MING, X.; YAN, J.; WANG, X. *et al.* Collaborative process planning and manufacturing in product lifecycle management. **Computers in Industry**, v. 59, n. 2-3, p. 154-166, mar 2008.

MONARO, R. L. G.; HELLENO, A. L. **DESENVOLVIMENTO DE UM ÍNDICE DE DESEMPENHO DINÂMICO (IDDyn) PARA AVALIAÇÃO DA ESTRATÉGIA DE USINAGEM EM MÁQUINAS-FERRAMENTAS**. (C. F. L. Leitão, Ed.) MOLDES 2010 - 8o Encontro da Cadeia de Ferramentas, Moldes e Matrizes. **Anais...** São Paulo: Associação Brasileira de Metalurgia e Materiais. , 2010

MONTAU, R. **Föderatives Produktdatenmanagement anhand semantischer Informationsmodellierung**. Düsseldorf: VDI Verlag, 1995.

MORGAN, G. **Imagens da organização**. 1a. ed. São Paulo: Atlas, 1996. p. 421

MORIN, E. **O Método I: a natureza da natureza**. 3ª. Edição. Portugal: Europa-América, 1997.

MORIN, E. Restrity Complexity, General Complexity. In: GERSHENSON, C.; AERTS, D.; EDMONDS, B. (Eds.). **Worldviews, Science and Us: Philosophy and Complexity**. 1st. ed. Liverpool: World Scientific Publishing, 2007. p. 5-29.

MOSCH, C.; ANDERL, R.; ASSIS MOURA, A. Á. DE; SCHÜTZER, K. **Prototype of a Federative Factory Data Management for the Support of Factory Planning Processes**. World Academy of Science, Engineering and Technology. **Proceedings...** Bangkok: Academy of Science, Engineering and Technology. , 2011

MOSCH, C.; ANDERL, R.; SCHÜTZER, K.; ASSIS MOURA, A. Á. DE. **Integrated Process Planning Based on a Federative Factory Data Management**. Intenational Mechanical Engineering Congress and Exposition. **Proceedings...** Vancouver, British Columbia: ASME. , 2010

MYERS, J. G. **The art fo software testing**. 1st. ed. New York: John Wiley & Sons, 1979. p. 177

NAKAI, A. M. **Uma Infra-Estrutura para Coordenação de Atividades em Cadeias Produtivas Baseada em Coreografia de Servicos Web**. Campinas: Universidade Estadual de Campinas, 2007. (Dissertação de Mestrado)

NI, Q.; YARLAGADDA, P. K. D. V.; BETTS, M. An Extensible Manufacturing Resource Model for Process Integration. **AIJST-Asian International Journal of Science and Technology: Production and Manufacturing**, v. 1, n. 4, p. 1-16, 2010.

NIELSEN, J. **Information Modeling of Manufacturing Processes : Information Requirements for Process Planning in a Concurrent Engineering Environment**. Stockholm: Royal Institute of Technology, 2003. (Doctoral Thesis)

NORRIE, H.; MATURANA, F. P. Multi-agent distributed Mediator architecture manufacturing. **Journal of Intelligent Manufacturing**, v. 7, p. 257-270, 1996.

ONG, S. K.; SUN, W. W. Application of mobile agents in a web-based real-time monitoring system. **The International Journal of Advanced Manufacturing Technology**, v. 22, n. 1-2, p. 33-40, set 2003.

PEDERSEN, S. **Interoperabilität heterogener Informationsquellen im Gesundheitswesen auf Grundlage von Standards für die medizinische Kommunikation und Dokumentation** Universität Oldenburg, 2005 (Tese de Doutorado).

PIDD, M. **Five Simple Principles of Modelling**. (J. M. Charnes, D. J. Morrice, D. T. Brunner, & J. J. Swain, Eds.) Winter Simulation Conference. **Proceedings...** Coronado, CA.: IEEE. , 1996

PIRNAU, M. General Information And Main Characteristics Regarding Web Services. Protocol Soap And Rest. **Annals of Faculty of Economics**, v. 4, n. 1, p. 1021-1024, 2009.

PRESSMAN, R. S. **Engenharia de Software**. 6a. ed. São Paulo: McGraw-Hill, 2006. p. 720

PRIETO JUNIOR, V. S. **Projetos Automotivos: Proposta para Redução de Tempo de Desenvolvimento**. São Paulo: Universidade de São Paulo, 2002. (Dissertação de Mestrado Profissional)

REMBOLD, U.; NNAJI, B. O.; STORR, A. **Computer Integrated Manufacturing and Engineering**. 1st. ed: Addison-Wesley, 1993. p. 664

RIBEIRO, D. D.; ROZENFELD, H. CAM (Computer Aided Manufacturing). São Paulo: NUMA, 1999. Disponível em : <numa.org.br/conhecimentos_port/pag_conhec/cam.htm>. Acesso em: 30/9/2012.

ROCHA, A. R. C. DA; MALDONADO, J. C. **Qualidade de Software: Teoria e Prática**. 1. ed. Sao Paulo: Prentice-Hall, 2001. p. 303

ROZENFELD, H.; FORCELLINI, F. A.; AMARAL, D. C. *et al.* **Gestão de desenvolvimento de produtos**. 1a. ed. São Paulo: Saraiva, 2006. p. 542

SCHLEICH, H.; SCHAFFER, J.; SCAVARDA, L. F. **Managing Complexity in Automotive Production**. ICPR - 19th International Conference on Production Research. **Proceedings...** Valparaiso, Chile, 2007

SCHUH, G.; BERGHOH, M.; WESTKÄMPER, E. Collaborative Production on the Basis of Object Oriented Software Engineering Principles. **CIRP Annals - Manufacturing Technology**, v. 52, n. 1, p. 393-396, 2003.

SCHWARZENBACHER, K.; WAGNER, J. **The Federative Principle in Business Architecture 2 Business Architecture : Vision and Reality**. (D. Konstantas, J.-P. Bourrières, M. Léonard, & N. Boudjlida, Eds.) Interoperability of Enterprise Software and Applications. **Proceedings...** Geneva: Springer. , 2005

SCHÜTZER, K.; ASSIS MOURA, A. Á. DE; ANDERL, R.; MOSCH, C. **Implemented Web Services for Data Integration in a Federative Environment**. 21st International Congress of Mechanical Engineering (COBEM). **Proceedings...** Natal, Brasil, 2011a

SCHÜTZER, K.; ASSIS MOURA, A. Á. DE; ANDERL, R.; MOSCH, C. **Implemented Web Services to Release Data of Product, Process and Resources for a Federative Management**. 9th International Conference on Advanced Manufacturing Systems and Technology (AMST). **Proceedings...** Vancouver, 2011b

SCHÜTZER, K.; ASSIS MOURA, A. Á. DE; ANDERL, R.; MOSCH, C. A Web Service Application to Support the Distributed Manufacturing. **Journal of the Brazilian Society of Mechanical Sciences and Engineering (JBSMSE)**, v. 2, n. 280, 2012.

SHETH, A. P.; LARSON, J. A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. **ACM Computing Surveys**, v. 22, n. 3, p. 183-236, 1 set 1990.

SHIGLEY, J. E. **Mechanical Engineering Design**. New York: McGraw-Hill, 1977.

SHIN, M.; JUNG, M. MANPro: mobile agent-based negotiation process for distributed intelligent manufacturing. **International Journal of Production Research**, v. 42, n. 2, p. 303-320, jan 2004.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 3a. ed. São Paulo: Pearson Makron Books, 1999. p. 778

SLACK, N. **Vantagem competitiva em manufatura**. São Paulo: Atlas, 2002. p. 224

SMITH, P. G.; REINERTSEN, D. G. **Developing products in half the time: new rules, new tools**. 2nd. Ed. ed. New York: John Wiley & Sons, Inc., 1998. p. 316

SO, B.; JUNG, Y.; PARK, J.; LEE, D. Five-axis machining time estimation algorithm based on machine characteristics. **Journal of Materials Processing Technology**, v. 187-188, p. 37-40, jun 2007.

SORDI, J. O. DE. **Gestão por Processos: uma abordagem da moderna administração**. 2nd. ed. São Paulo: Saraiva, 2005. p. 223

STADNICK, K. T. **Modelo de Adaptação Evolucionária da Vantagem da Complexidade - Desenvolvimento de um Instrumento de Avaliação**. Florianópolis: UFSC, 2006. (Dissertação de Mestrado)

SUJITHAN, K. R. **An object model of data, based on the ODMG industry standard for database applications**. International Seminar on Client/Server Computing. **Proceedings...** La Hulpe: IEE. Disponível em: <<http://link.aip.org/link/IEESEM/v1995/i184/pv1-13/s1&Agg=doi>>. Acesso em: 27 jul. 2012. , 1995

SZWARCFITER, C. & DALCOL, P.R. **Economias de Escala e de Escopo: Desmistificando Alguns Aspectos da Transição**. Revista Produção, v.7, n.2, ABEPRO, pp. 117-129, 1997.

TÁLAMO, J. R. **Formação e gestão de redes de cooperação empresarial**. São Paulo: Universidade de São Paulo, 2008. (Tese de Doutorado)

TEÓFILO, A.; SILVA, A. R. D. **CBPEL - Linguagem para definição de processos de negócio interorganizacionais Descrição de processos interorganizacionais**. (J. C. Ramalho, A. Simoes, & J. C. Lopes, Eds.)XML: Aplicações e Tecnologias Associadas. **Anais...** Braga: Universidade do Minho. Disponível em: <<http://hdl.handle.net/1822/865>>. , 2005

THOMAS, G.; THOMPSON, G. R.; CHUNG, C.-W. *et al.* Heterogeneous distributed database systems for production use. **ACM Computing Surveys**, v. 22, n. 3, p. 237-266, set 1990.

THOMASMA, T.; NOLLER, D. **SOA in Manufacturing Guidebook**. Mesa International 2008 Plant-to-Enterprise Conference. **Proceedings...** Orlando: MESA. , 2008

TUDORACHE, T. **Employing Ontologies for an Improved Development Process in Collaborative Engineering**. Berlin: Technische Universität Berlin, 2006. (Doctoral Thesis)

URBANIC, R. J.; ELMARAGHY, W. H. Modeling of Manufacturing Process Complexity. In: PHAM, D. T.; ELMARAGHY, H. A.; ELMARAGHY, W. H. (Eds.). **Advances in Design**. Springer S ed. London: Springer London, 2006. p. 425-436.

VIEIRA, V. H. **Web Services de apoio a aplicações voltadas ao trabalho em grupo**. São Carlos: Universidade de São Paulo - São Carlos, 2006. (Dissertação de Mestrado)

WESTKÄMPER, E.; BRIEL, R. VON. Continuous Improvement and Participative Factory Planning by Computer Systems. **CIRP Annals - Manufacturing Technology**, v. 50, n. 1, p. 347-352, 2001.

WIGHT, O. **Manufacturing Resource Planning M.R.P. II: Unlocking America's Potential**. 1st. ed. Essex Jn: Oliver Wight Ltd Pub, 1981. p. 484

WILKINSON, F. Productive Systems. **Cambridge Journal of Economics**, v. 7, p. 413-429, 1983.

YUAN, Y.; NIAN, M.; SUN, W. Manufacturing Grid Workflow framework and model. **Advanced Materials Research**, v. 37, p. 1431-1438, 2008.

ZANCUL, E. D. S. **Gestão do ciclo de vida de produtos : seleção de sistemas PLM com base em modelos de referência**. São Carlos: Escola de Engenharia de Sao Carlos da Universidade de Sao Paulo, 2009. (Tese de Doutorado)

ZARIFIAN, P. **Objetivo e Competência**. São Paulo: Atlas, 2001.

ZHANG, Q. Object-oriented database systems in manufacturing: selection and applications. **Industrial Management & Data Systems**, v. 101, n. 3, p. 97-105, 2001.

ZIMMERMANN, J. U. **INFORMATIONAL INTEGRATION OF PRODUCT DEVELOPMENT SOFTWARE IN THE AUTOMOTIVE INDUSTRY**. Twente: University of Twente, 2005. (Doctoral Thesis)

Apêndice A – Programa Fonte

O programa fonte está disponível no CD que acompanha a tese.

Apêndice B – Projeto *FEDMAN*

Esta tese está inserida no contexto do projeto *FEDMAN* - “*Federative Factory Data Management based on Service Oriented Architecture and Semantic Model Description on XML and RDF for Manufacturing Products*” é um projeto de pesquisa desenvolvido em parceria pela *Technische Universität Darmstadt*, através do DiK - *Institut für Datenverarbeitung in der Konstruktion* e pela Universidade Metodista de Piracicaba, através do Laboratório de Sistemas Computacionais para Projeto e Manufatura. O projeto é financiado pelo DFG (*Deutschen Forschungsgemeinschaft*), pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), pela FINEP (Financiadora de Estudos e Projetos) e pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) dentro do programa BRAGECRIM.

Por ser um projeto com proposta de elevada integração, o desenvolvimento foi feito em conjunto, através do trabalho de parceria entre os pesquisadores de ambos os institutos, que possuem tradição em pesquisas sobre gerenciamento de dados da manufatura. A arquitetura central do projeto é apresentada na Figura B.1, de onde se destacam:

- Uma camada de apresentação para a qual foi desenvolvido um aplicativo do tipo navegador para apresentar os dados dos bancos de dados federativos;
- Uma camada de serviços Web³ para o acesso aos dados distribuídos e disponibilização ao modelo central;
- Uma camada federativa para integração dos dados.

Para o adequado desenvolvimento destas camadas também é necessário o desenvolvimento um modelo central capaz de abrigar a informações distribuídas.

³ Componente de software que agrega uma funcionalidade acessível através de protocolos padrões de Internet.

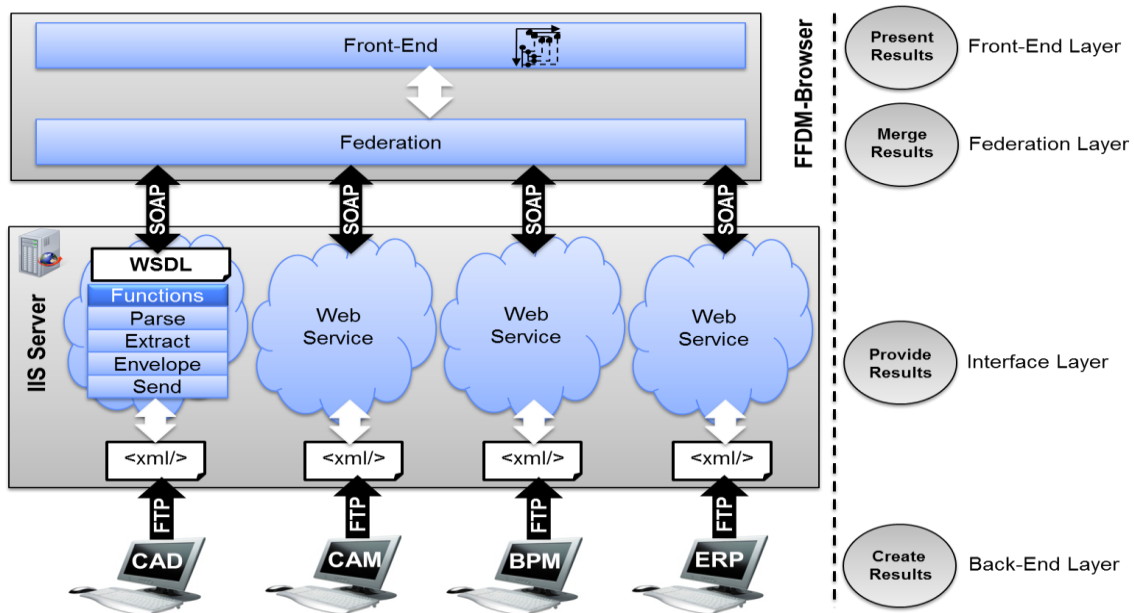


Figura B.1 Arquitetura central do projeto FEDMAN

Agregados a estes desenvolvimentos centrais estão os módulos referentes à implantação de requisitos de segurança da informação e da criação de um modelo virtual para testes e validações da proposição.

Os requisitos centrais do projeto *FEDMAN* descritos na sua propositura foram publicados por Mosch *et al.* (2010) e podem ser apresentados em um sistema que atenda à integração dos dados de um sistema de manufatura utilizando ferramentas de TI atuais, legadas, e isoladas com base em um Gerenciamento Federativo de Dados.

Um sistema de manufatura pode ser tratado em vários níveis desde componentes, máquinas, planta, fábrica e até companhia. Como para cada nível existem ferramentas de TI específicas, o aumento do nível implica, necessariamente, em um aumento da quantidade de dados a serem gerenciados.

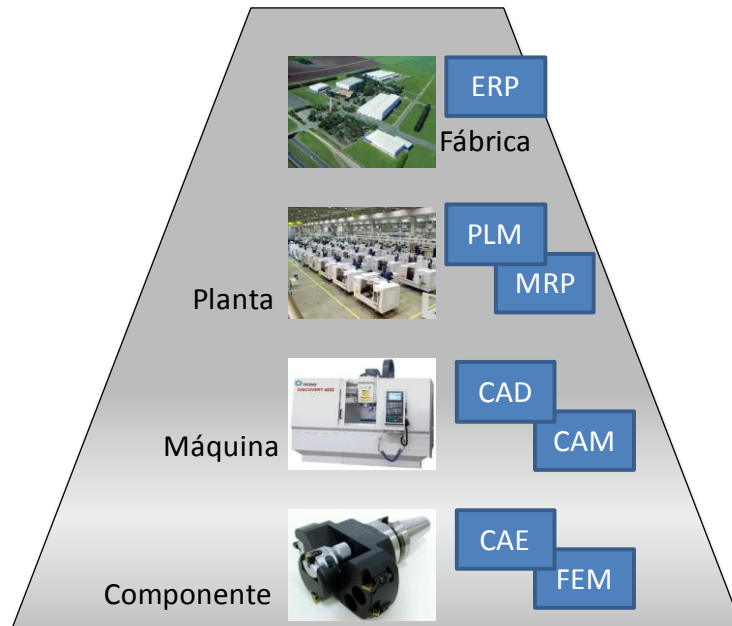


Figura 7.1 Níveis em sistema de manufatura e exemplos de sistemas de TI associados.

O uso destas ferramentas específicas é feita, normalmente, na forma de soluções isoladas. Estas soluções, por sua vez, utilizam bancos de dados isolados. Como consequência desta situação surgem dados redundantes, inconsistentes e heterogêneos que precisam ser gerenciados adequadamente. O núcleo da pesquisa do projeto *FEDMAN* está justamente localizado no número crescente de ferramentas de TI envolvidas nos processos e sua relação com os níveis de planejamento da fábrica.

Para efetivar esta integração entre dados de várias origens o projeto *FEDMAN* utiliza o conceito de dados federativos onde a integração é feita através de um acoplamento fraco das ferramentas de TI. Esta abordagem federativa trata os dados de forma encapsulada de modo a manter a autonomia parcial das ferramentas específicas, sendo a integração feita através de interseções e compartilhamento das bases de dados autônomas de modo a organizar o sistema de maneira completa e coerente, sem perda significativa de flexibilidade.

A abordagem federativa requer uma comunicação entre as ferramentas de TI que possibilite o compartilhamento dos dados gerados e administrados autonomamente por cada uma das ferramentas. Esta comunicação é realizada com base em uma

arquitetura orientada a serviços (SOA - *Services Oriented Architecture*) onde os serviços Web são os agentes responsáveis pela comunicação e integração dos dados.

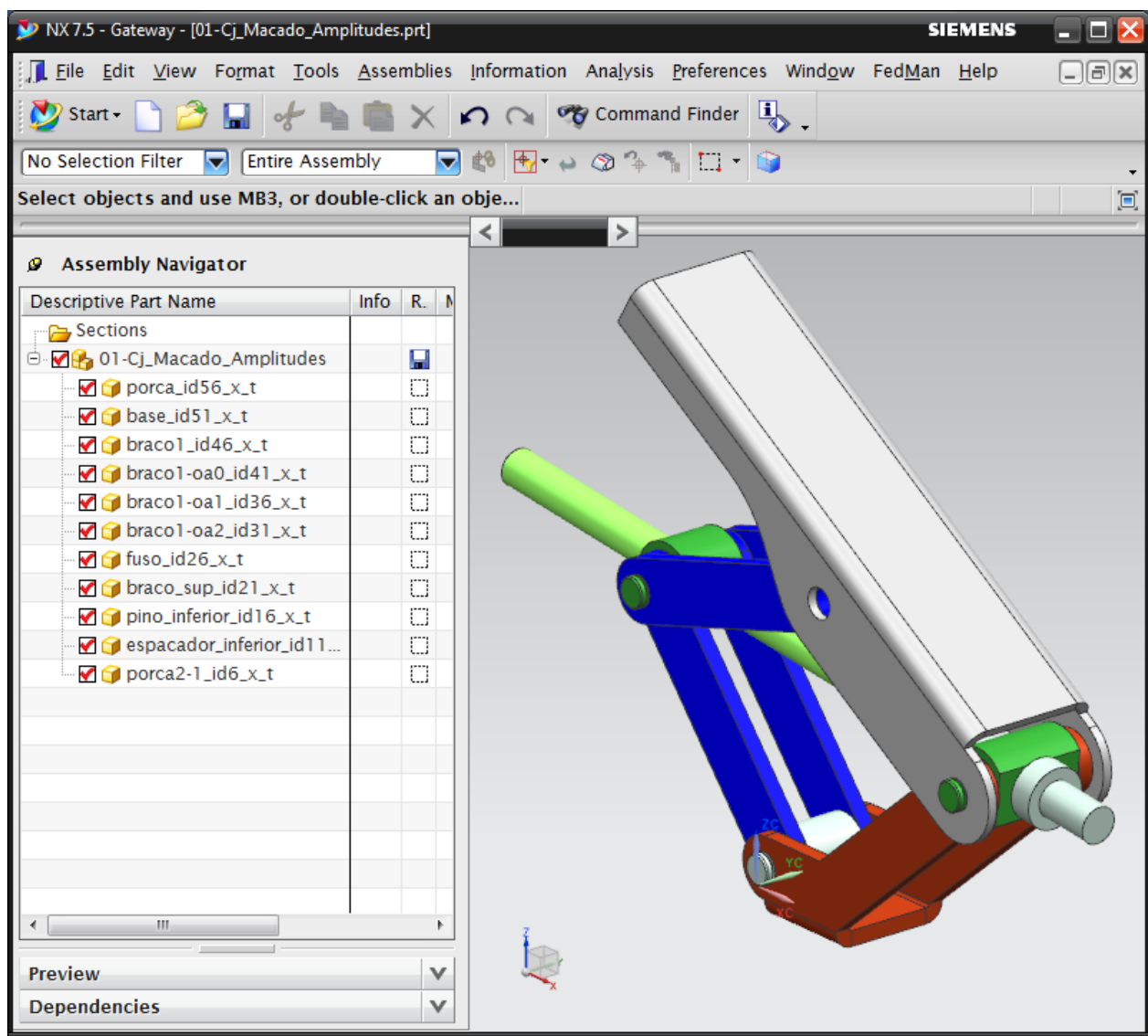
Os serviços estão disponíveis em servidores Web e têm acesso aos dados resultantes das ferramentas específicas de TI (por exemplo, CAD, CAM e ERP), o serviço específico para estimativa de tempo de manufatura foi publicado por SCHÜTZER *et al.* (2011a). No projeto *FEDMAN* foi criado um aplicativo que permite a um usuário fazer a solicitação dos dados através dos serviços disponibilizados, o trabalho resultante deste aplicativo foi descrito por Mosch *et al.* (2011). Os dados das ferramentas de TI específicas são disponibilizados em linguagem XML em bancos de dados hospedados nos servidores Web. Esta disponibilização é realizada através da liberação, pelo usuário da ferramenta específica, através de um aplicativo especialmente desenvolvido que utiliza o protocolo FTP (*File Transfer Protocol*) implantado nos servidores Web.

Apêndice C – Modelos para testes

Nesse apêndice são apresentados os resultados das trocas de informações entre os componentes de software durante as três fases do processo. O modelo no sistema CAD/CAM; os dados no arquivo disponibilizado na Web e os dados apresentados no aplicativo de integração.

Modelo 1 – Macaco

Tela do Sistema CAD



Informações do Conjunto no Sistema CAD

```

=====
Information listing created by : a3moura
Date : 27/07/2012 17:36:06
Current work part : D:\Tese\Conjuntos\Conjuntos\Macaco\01-Cj_Macado_Ampli
tudes.prt
Node name : pfungstaedter
=====
Component Report
Components of D:\Tese\Conjuntos\Conjuntos\Macaco\01-Cj_Macado_Amplitudes.prt :
on 07-
-27--2012 17:36
Part Name Ref Set Name Component Name Count Comment
Units Directory
base_id51_x_t Entire Part BASE_ID51_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
bracol-oa0_id41_x_t Entire Part BRACO1-OA0_ID41_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
bracol-oa1_id36_x_t Entire Part BRACO1-OA1_ID36_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
bracol-oa2_id31_x_t Entire Part BRACO1-OA2_ID31_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
bracol_id46_x_t Entire Part BRACO1_ID46_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
braco_sup_id21_x_t Entire Part BRACO_SUP_ID21_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
espacador_inferior_id11_x_t Entire Part ESPACADOR_INFERIOR_ID11_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
fuso_id26_x_t Entire Part FUSO_ID26_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
pino_inferior_id16_x_t Entire Part PINO_INFERIOR_ID16_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
porca2-1_id6_x_t Entire Part PORCA2-1_ID6_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco
porca_id56_x_t Entire Part PORCA_ID56_X_T 1 Partiall
y loaded Mm D:\Tese\Conjuntos\Conjuntos\
Macaco

```

All components are positioned as: Absolute

Arquivo XML gerado pelo aplicativo interno

PORCA_ID56_X_T

1

none

-47,9830207033275

15

78,8329086274689

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

BASE_ID51_X_T

1

none

0

0

0

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

BRACO1_ID46_X_T

1

none

-66,5755006839592

-2,14233666609894E-14

118,505757408991

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

BRACO1-OA0_ID41_X_T

1

none

-66,5755006839525

25

118,505757408981

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

BRACO1-OA1_ID36_X_T

1

none

-91,794513379103

30

78,4226016797485

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

BRACO1-OA2_ID31_X_T

1

none

-91,7945133794932

-5

78,4226016796748

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

FUSO_ID26_X_T

1

none

119,242485390606

15

60,8835006906008

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

BRACO_SUP_ID21_X_T

1

none

108,620607702053

15

44,6812585352577

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

PINO_INFERIOR_ID16_X_T

1

none

10

15

10

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

ESPACADOR_INFERIOR_ID11_X_T

1

none

10,0000000000113

15

9,99999999998456

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

PORCA2-1_ID6_X_T

1

none

102,043515631041

15

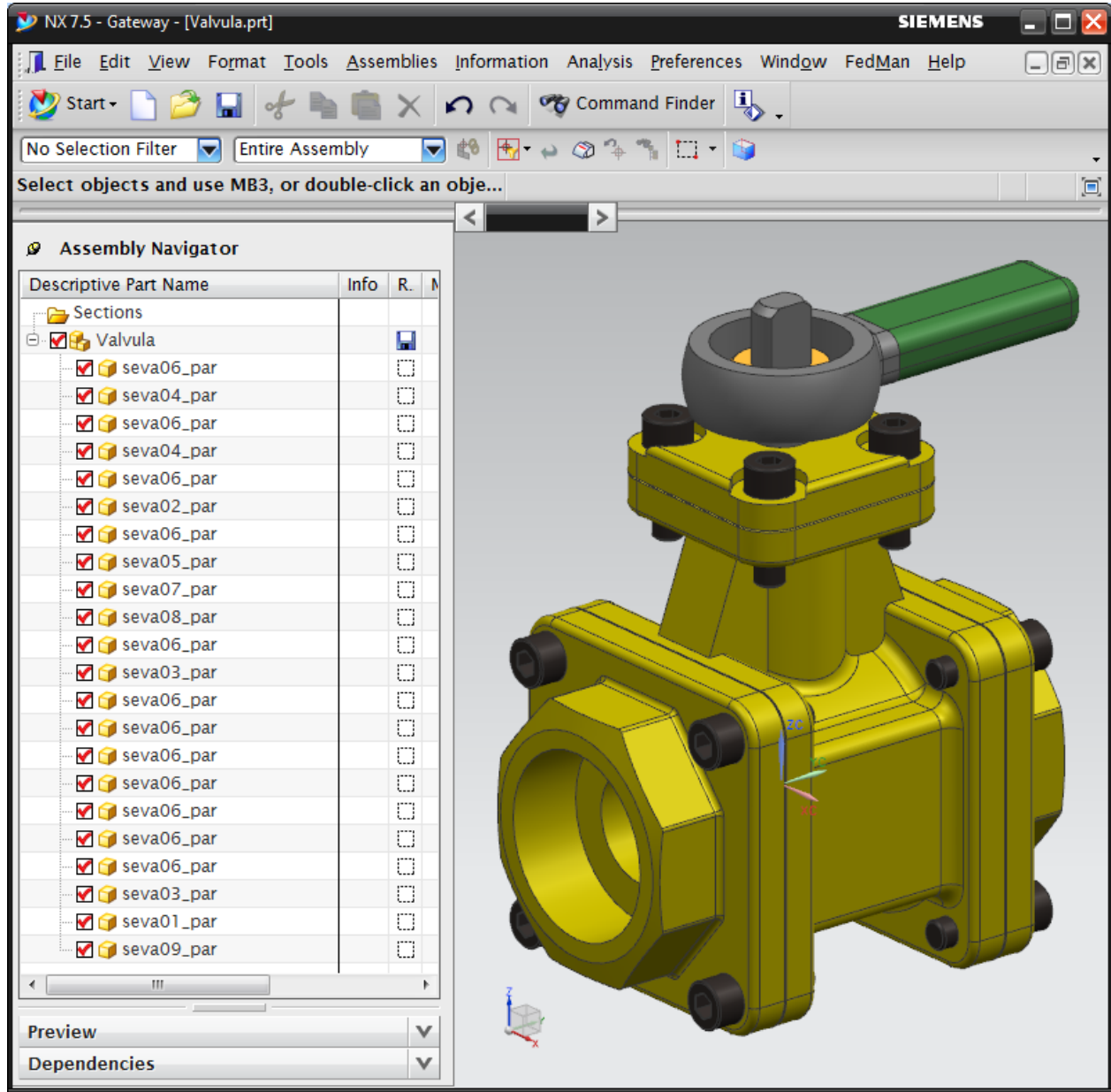
62,729578669488

NXOpen.Assemblies.Component

01-Cj_Macado_Amplitudes.prt

Modelo 2 – Válvula

Tela do Sistema CAD



Informações do Conjunto no Sistema CAD

```

=====
Information listing created by : a3moura
Date                          : 27/07/2012 17:46:24
Current work part              :
D:\Tese\Conjuntos\Conjuntos\Valvula\Valvula.prt
Node name                      : pfungstaedter
=====

```

Component Report

Components of D:\Tese\Conjuntos\Conjuntos\Valvula\Valvula.prt : on 07--27--2012 17:46

Part Name	Ref Set Name	Component Name	Count	Comment	Units
Directory					
seva01_par	Entire Part	SEVA01.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva02_par	Entire Part	SEVA02.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva03_par	Entire Part	SEVA03.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva03_par	Entire Part	SEVA03.PAR_2	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva04_par	Entire Part	SEVA04.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva04_par	Entire Part	SEVA04.PAR_2	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva05_par	Entire Part	SEVA05.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_10	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_11	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_12	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_13	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_14	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_15	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_16	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_5	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_7	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_8	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva06_par	Entire Part	SEVA06.PAR_9	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					

seva07_par	Entire Part	SEVA07.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva08_par	Entire Part	SEVA08.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					
seva09_par	Entire Part	SEVA09.PAR_1	1	Partially loaded	Mm
D:\Tese\Conjuntos\Conjuntos\Valvula					

All components are positioned as: Absolute

Arquivo XML gerado pelo aplicativo interno

SEVA06.PAR_13

1

none

35

-45,5

-34,99999999999999

NXOpen.Assemblies.Component

Valvula.prt

SEVA04.PAR_2

1

none

-2,0064204474632E-14

35,5

3,59306569391502E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_5

1

none

34,99999999999999

45,5

-35

NXOpen.Assemblies.Component

Valvula.prt

SEVA04.PAR_1

1

none

2,9662308323485E-15

-35,5

2,33991714704289E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_16

1

none

34,99999999999999

45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA02.PAR_1

1

none

-7,541233292811E-14

-2,09171138729922E-14

23

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_14

1

none

-35

45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA05.PAR_1

1

none

-3,62469079311604E-15

-8,06354101760521E-16

83,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA07.PAR_1

1

none

4,42710729375063E-18

28,1674329480672

110

NXOpen.Assemblies.Component

Valvula.prt

SEVA08.PAR_1

1

none

6,20850609793979E-15

28,1674329480672

110

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_12

1

none

20

-20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA03.PAR_2

1

none

-8,99145046059736E-14

3,70542465753424

2,18172819583352E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_11

1

none

20

20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_8

1

none

-34,9999999999999

-45,5

-35

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_10

1

none

-20

20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_15

1

none

-35

45,5

-34,9999999999999

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_1

1

none

-34,9999999999999

-45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_9

1

none

-20

-20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_7

1

none

35

-45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA03.PAR_1

1

none

-9,99040696940614E-14

-3,7054246575343

2,33025047916777E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA01.PAR_1

1

none

0

0

0

NXOpen.Assemblies.Component

Valvula.prt

SEVA09.PAR_1

1

none

-7,46712516556209E-15

0,00142245955046771

106

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_13

2

none

35

-45,5

-34,9999999999999

NXOpen.Assemblies.Component

Valvula.prt

SEVA04.PAR_2

2

none

-2,0064204474632E-14

35,5

3,59306569391502E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_5

2

none

34,9999999999999

45,5

-35

NXOpen.Assemblies.Component

Valvula.prt

SEVA04.PAR_1

2

none

2,9662308323485E-15

-35,5

2,33991714704289E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_16

2

none

34,9999999999999

45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA02.PAR_1

2

none

-7,541233292811E-14

-2,09171138729922E-14

23

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_14

2

none

-35

45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA05.PAR_1

2

none

-3,62469079311604E-15

-8,06354101760521E-16

83,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA07.PAR_1

2

none

4,42710729375063E-18

28,1674329480672

110

NXOpen.Assemblies.Component

Valvula.prt

SEVA08.PAR_1

2

none

6,20850609793979E-15

28,1674329480672

110

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_12

2

none

20

-20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA03.PAR_2

2

none

-8,99145046059736E-14

3,70542465753424

2,18172819583352E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_11

2

none

20

20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_8

2

none

-34,9999999999999

-45,5

-35

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_10

2

none

-20

20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_15

2

none

-35

45,5

-34,99999999999999

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_1

2

none

-34,99999999999999

-45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_9

2

none

-20

-20

90,5

NXOpen.Assemblies.Component

Valvula.prt

SEVA06.PAR_7

2

none

35

-45,5

35

NXOpen.Assemblies.Component

Valvula.prt

SEVA03.PAR_1

2

none

-9,99040696940614E-14

-3,7054246575343

2,33025047916777E-14

NXOpen.Assemblies.Component

Valvula.prt

SEVA01.PAR_1

2

none

0

0

0

NXOpen.Assemblies.Component

Valvula.prt

SEVA09.PAR_1

2

none

-7,46712516556209E-15

0,00142245955046771

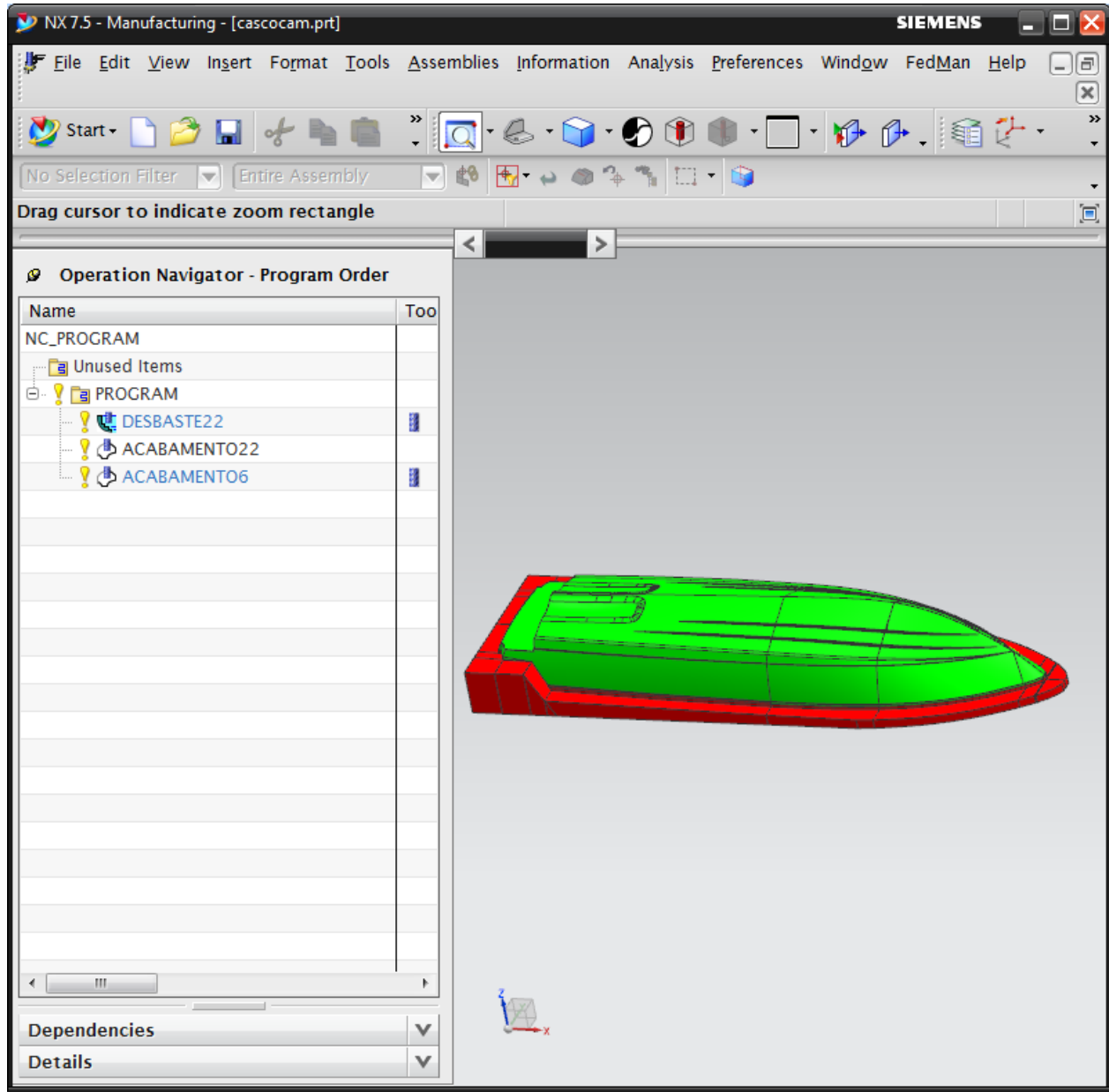
106

NXOpen.Assemblies.Component

Valvula.prt

Modelo 3 – Casco

Tela do Sistema CAM



Arquivo XML gerado pelo aplicativo interno

22

1

100

TOPO22

22

1

100

TOPO22

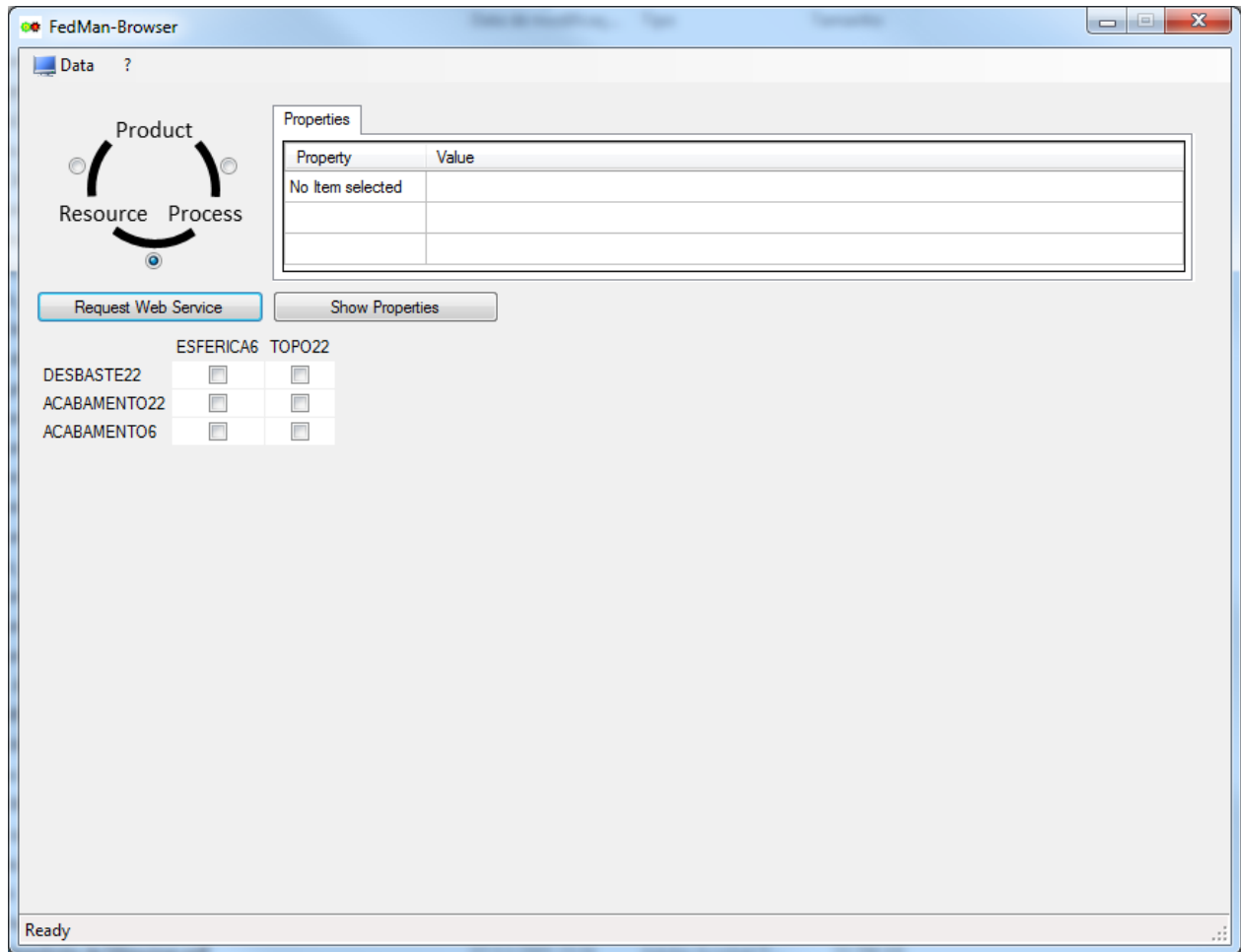
6

2

120

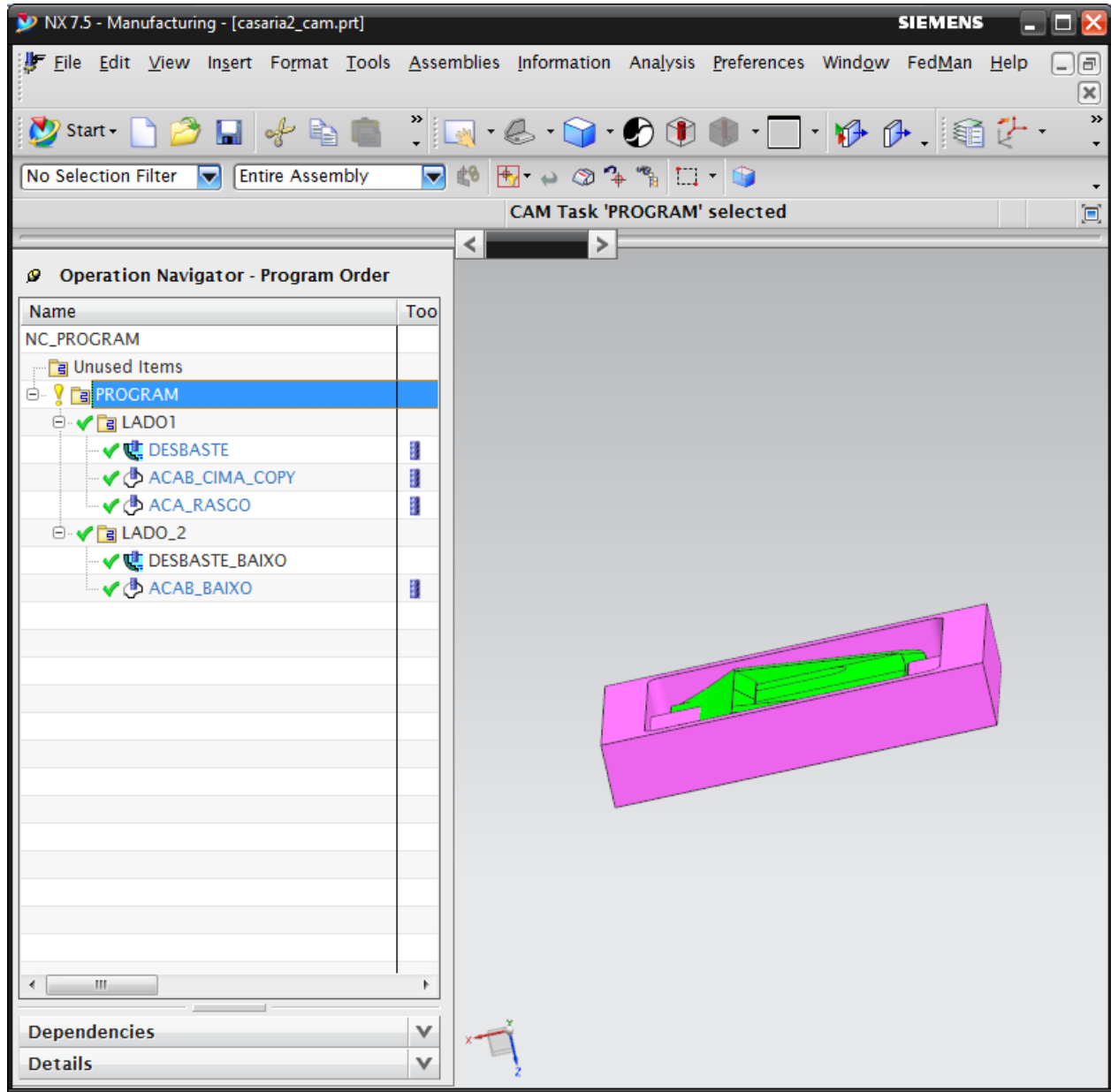
ESFERICA6

Tela do Aplicativo de Integração



Modelo 4 – Entrada de Ar

Tela do Sistema CAM



Arquivo XML gerado pelo aplicativo interno

10

4

75

TOPO_10

6

2

75

ESFERICA_6

10

4

75

TOPO_10

10

4

75

TOPO_10

6

2

75

ESFERICA_6

Tela do Aplicativo de Integração

