

CAPÍTULO IV

ESTUDO DE CASO

A melhor forma de testar e validar o aplicativo PlanPAS proposto, é através da sua aplicação em um sistema prático. Dentre diversos problemas práticos da engenharia que são passíveis de técnicas de planejamento para a sua solução, existem os problemas de distribuição logística. Um problema clássico de distribuição logística que exige técnicas de planejamento diz respeito às operações de cabotagem, incluindo carga e descarga de contêineres bem como carga e descarga de reservatórios de óleo cru, na indústria petroleira (LI et al., 2006). Para que o trabalho desenvolvido fosse de fato avaliado em um sistema prático, optou-se por utilizar uma bancada para fins didáticos, capaz de simular um sistema de distribuição de produto entre fornecedor, distribuidores e clientes, inseridos em uma cadeia de suprimentos. A bancada escolhida simula um sistema de distribuição controlado por um CLP Omron CJ1M CPU13 ETN21 (OMRON CORPORATION, 2001). O sistema consiste basicamente numa cadeia de suprimento simples, composta por um reservatório fornecedor, capaz de abastecer reservatórios distribuidores a partir de serviços prestados por um veículo transportador. Cada reservatório distribuidor é capaz ainda de atender demandas específicas de clientes.

4.1. Apresentação do Sistema Analisado no Estudo de Caso

O sistema físico simula um sistema de logística e transporte de um fornecedor para dois distribuidores utilizando um veículo controlado por um CLP. Cada distribuidor possui uma eletrobomba responsável por atender diversos clientes e o sistema de planejamento deve definir as ações de carregamento, descarregamento e movimentação do veículo de forma a atender as demandas dos clientes atendidos por cada distribuidor. O serviço de entregas do veículo é requisitado em função da variação do nível dos distribuidores. A Fig.

4.1 apresenta uma representação esquemática e o esboço tridimensional da bancada didática.

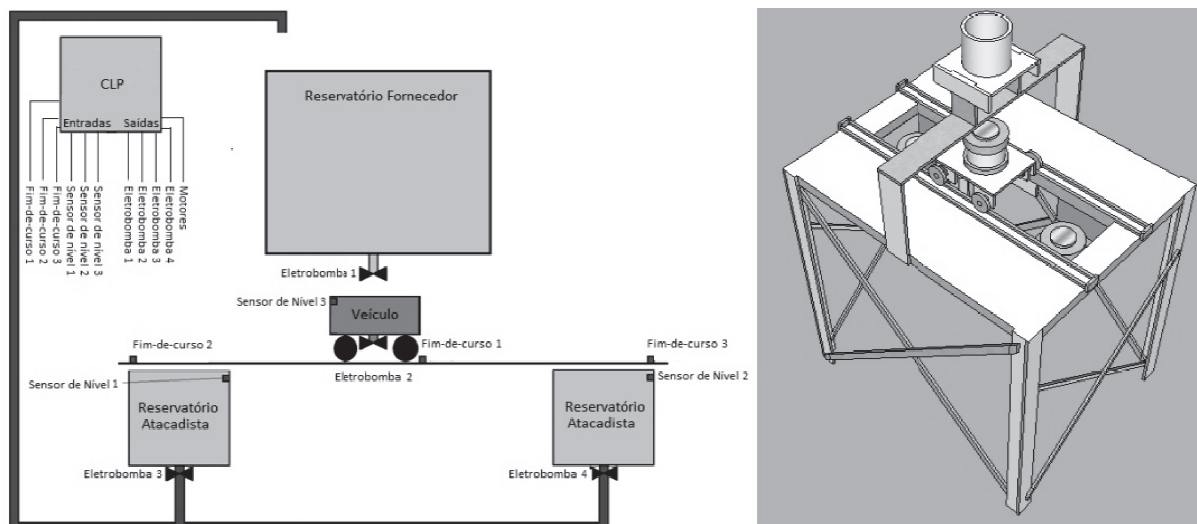


Figura 4.1. Representação esquemática do domínio utilizado para análise (adaptada de Fonseca e Tavares (2012)).

A bancada possui quatro eletrobombas, uma no fornecedor (para abastecer o veículo), outra no veículo (para atender os distribuidores), e mais duas em cada distribuidor para atender a demanda de seus clientes. Dois motores 12Vcc para a movimentação do veículo completam os atuadores do sistema. A posição do veículo é monitorada por três chaves fins de curso mecânicos, uma no fornecedor e mais uma em cada distribuidor. Tanto o veículo como os distribuidores possuem sensores de nível potenciométricos para monitorar o ciclo de estoque entre os reservatórios e a quantidade de produto entregue aos clientes. O produto fornecido como demanda para os clientes retorna ao reservatório fornecedor, fechando-se o ciclo e garantindo uma característica de reservatório ideal para o fornecedor, ou seja, com capacidade de fornecimento infinita. O produto transportado e distribuído nesta bancada é água. Uma foto da bancada física é apresentada na Fig. 4.2.



Figura 4.2. Foto da bancada didática.

4.2. Modelagem do Sistema Real

O processo de modelagem do domínio foi totalmente idealizado em diagramas UML do itSIMPLE. Esta modelagem é dividida em duas etapas: modelagem do domínio, através dos diagramas de casos de uso, classes e estados; e modelagem do problema, através dos diagramas de objetos, sistematizando os cenários inicial e objetivo do problema.

4.2.1. Modelagem do Domínio de Planejamento

De acordo com Vaquero (2007), o processo de modelagem de domínios de planejamento no itSIMPLE inicia-se com o diagrama de casos de uso. Nele são definidos os atores do domínio e os casos de uso aos quais estes estão ligados.

A bancada possui dois atores: veículo (*Vehicle*) e distribuidor (*Distributor*). O veículo pode se mover (*Move*), carregar (*Load*) e descarregar (*Unload*). Já o distribuidor pode efetuar uma entrega completa (*CompleteOrder*), ou uma entrega parcial (*PartialOrder*) e sua complementação (*FinalOrder*). Especificamente, o caso de uso *Unload* está vinculado

diretamente tanto com o ator *Vehicle* quanto com o ator *Distributor*. O diagrama de casos de uso é também o diagrama onde o projetista define os requisitos e restrições do sistema real. Para isso, ressaltam-se as características de cada ator e definem-se as pré-condições e efeitos para cada caso de uso. Estas informações armazenadas aqui servem como base para a elaboração das regras dos objetos que compõem o modelo. A Fig. 4.3 apresenta o diagrama de casos de uso do domínio em questão.

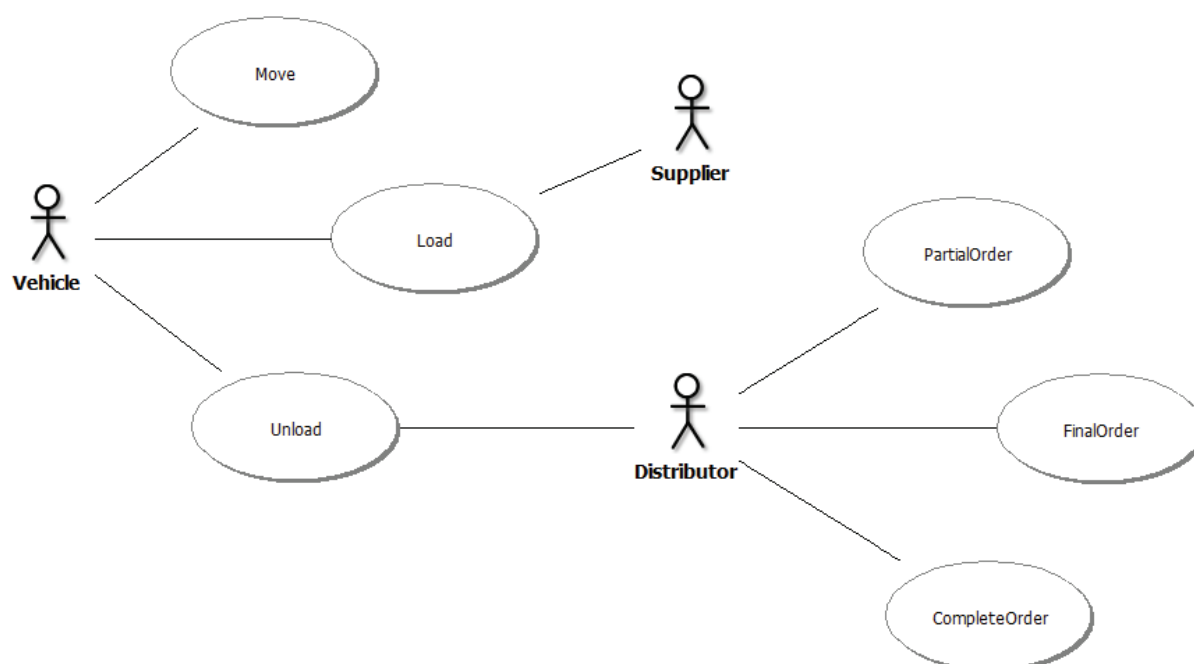


Figura 4.3. Diagrama de casos de uso da bancada didática (FONSECA et al., 2012).

Dando sequência à modelagem do domínio, o projetista deve definir as classes que irão instanciar os objetos do modelo e as relações entre estas classes. Este processo é sistematizado através do diagrama de classes. A Fig. 4.4 apresenta o diagrama de classes da bancada didática.

Para modelar as classes, o projetista deve conhecer bem o domínio, visto que cada classe deve possuir os atributos e métodos que representem o comportamento dos objetos no sistema real. Por exemplo, a classe *Vehicle* possui as informações necessárias para instanciar todos os objetos do tipo *Vehicle*, ou seja, um objeto do tipo *Vehicle* possui informações que representam o seu nível máximo, nível crítico e o seu nível percebido em um determinado momento, estas informações foram definidas na classe *Vehicle* como os atributos *maxlevel*, *critical_level* e *lev*, respectivamente. Além disso, um objeto do tipo veículo deve ser capaz de realizar determinadas ações como movimentar-se de um lugar de origem a um lugar de destino, carregar seu reservatório no momento em que estiver estacionado em um fornecedor e descarregar o seu reservatório, abastecendo um

distribuidor no qual está estacionado. Estas ações foram definidas na classe *Vehicle* como os operadores *move*, *load* e *unload*.

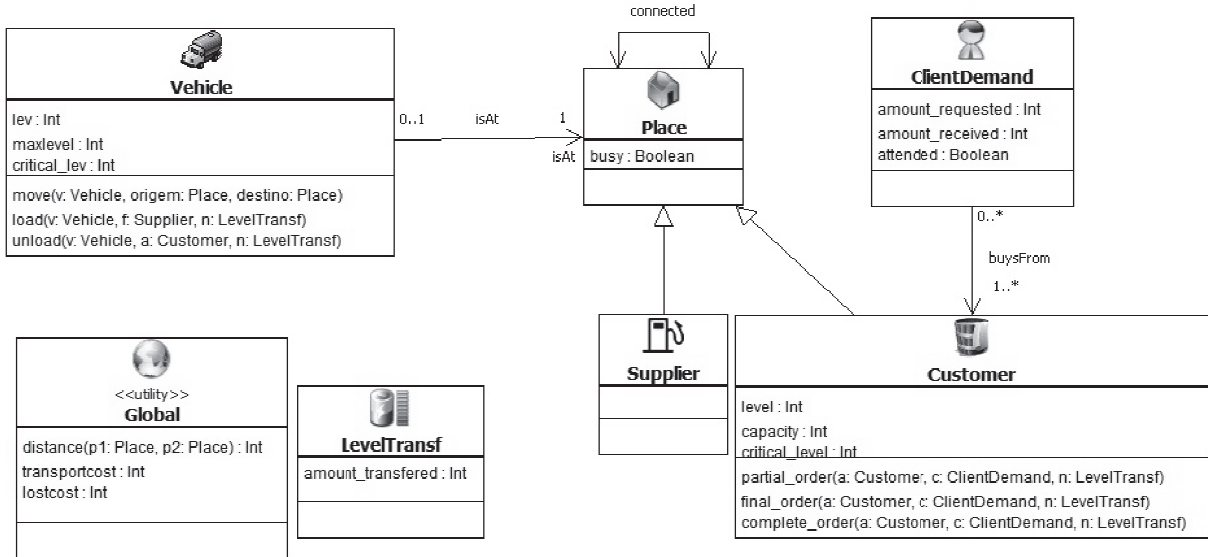


Figura 4.4. Diagrama de classes da bancada didática (FONSECA et al., 2012).

Os planejadores automáticos trabalham de maneira tal que, dada uma situação inicial e uma situação final que se deseja atingir, um algoritmo varre uma árvore de busca de forma a encontrar a solução que mais se ajuste às características do projeto, que podem ser minimizar ou maximizar tempo de processamento, tempo de uma operação, custo de uma operação, etc. O processo de otimização de parâmetros em um modelo de planejamento automático é baseada na definição de métricas.

As métricas desenvolvidas neste modelo dizem respeito à minimização de custos. No domínio da bancada existem dois custos passíveis de minimização: custo de transporte (*transportCost*) e custo de perda (*lostCost*). A Eq. 4.1 atualiza o custo de transporte a cada operação *move* em função do custo de transporte atual anterior, da quantidade de produto transportada pelo veículo e da distância entre os locais de origem e destino.

$$transportCost(n) = transportCost(n - 1) + distance(origem, destino). (v. lev(n) + 1) \quad (4.1)$$

O custo de perda é contabilizado sempre que um distribuidor não consegue atender a demanda de um de seus clientes com apenas uma operação de entrega. Normalmente, o custo de perda é algo acordado através de uma cláusula do contrato entre distribuidor e cliente, e, para este caso específico, definiu-se que o custo de perda será atualizado a cada operação *partialOrder* em função do custo de perda anterior e de uma penalidade no valor

de 50 vezes a diferença entre a quantidade de produto solicitada e a quantidade de produto recebida pelo cliente. A Eq. 4.2 atualiza o valor do custo de perda nessas situações.

$$lostCost(n) = lostCost(n - 1) + 50(amounted_requested - amount_received(n)) \quad (4.2)$$

Estas variáveis são definidas como atributos da classe *Global* e sua representação lógica é dada nos Diagramas de Estados como regras das ações do Veículo (Fig. 4.5) e do Distribuidor (Fig. 4.6). Por definição, o Diagrama de Estados é onde são modeladas as classes com características dinâmicas relevantes, neste caso, foram modelados dois Diagramas de Estados, um referente à classe *Vehicle*, cujos objetos podem se movimentar de um lugar a outro, ou ter seu nível afetado pelas operações *Load* ou *Unload*, e outro referente à classe *Distributor*, cujos objetos podem ter seu nível modificado por operações do tipo *Unload*, *PartialOrder*, *FinalOrder* ou *CompleteOrder*. Na prática, os diagramas de estados definem as “leis” da dinâmica do sistema modelado.

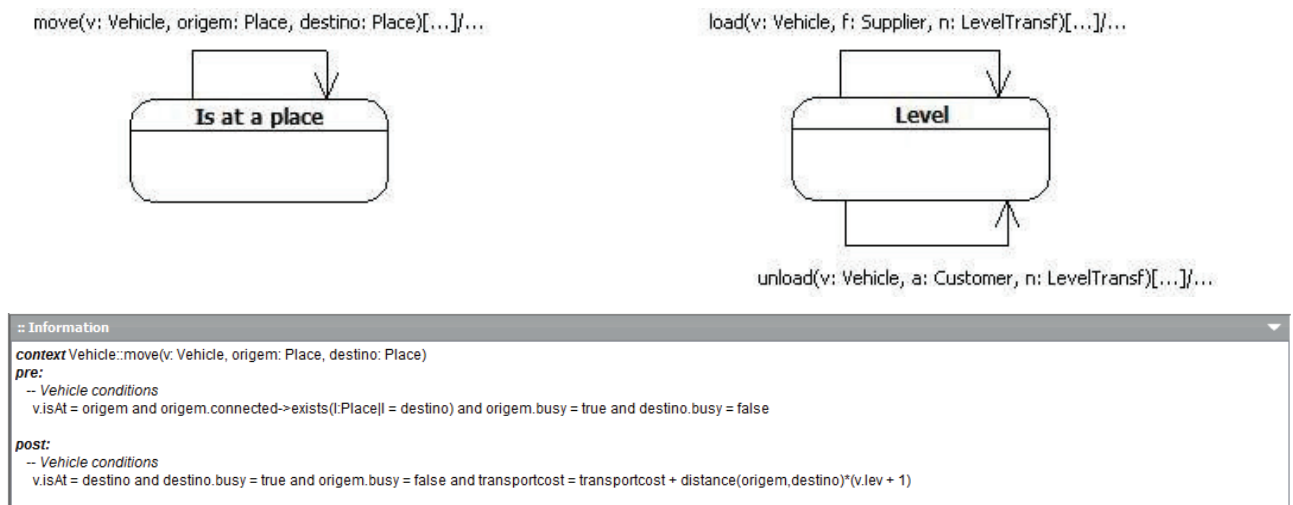


Figura 4.5. Diagrama de estados para o veículo (FONSECA et al., 2012).

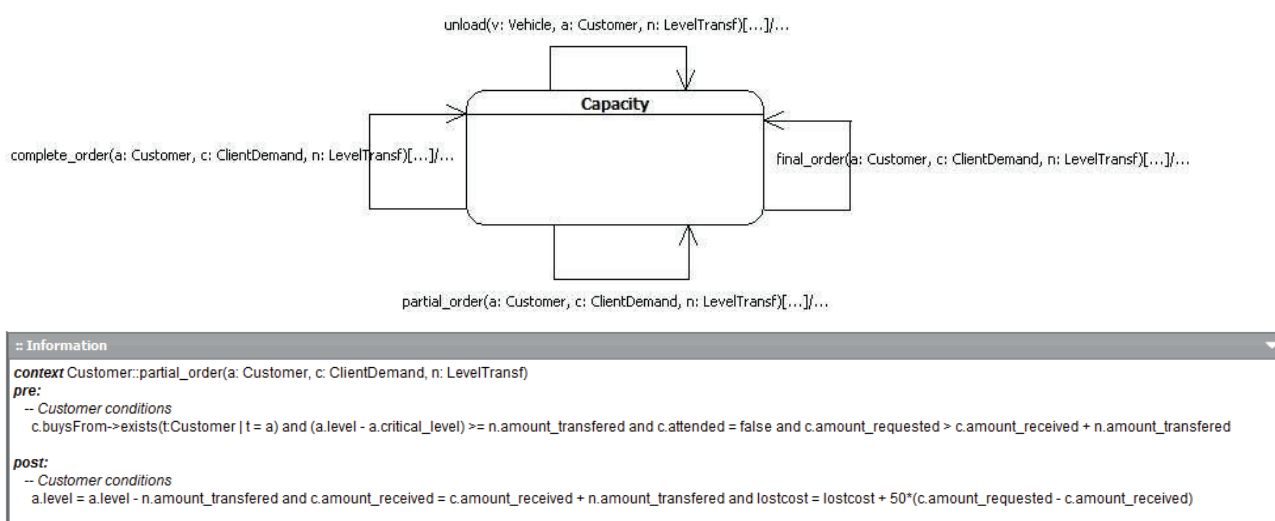


Figura 4.6. Diagrama de estados para o distribuidor (FONSECA et al., 2012).

As pré-condições e os efeitos de cada caso de uso, definidos no diagrama de casos de uso, são inseridos no diagrama de estados como as regras necessárias para a ocorrência das ações relacionadas com os casos de uso apresentados na Fig. 4.3. Por exemplo, a pré-condição para que a ação *move* ocorra é que: o veículo deve estar estacionado no lugar de origem; o lugar de destino deve estar conectado, através de uma rota, com o lugar de origem; o lugar de origem deve estar ocupado e o lugar de destino desocupado. O efeito desta ação é o veículo estacionado na posição referente ao lugar de destino; o lugar de destino passa a estar ocupado e o lugar de origem passa a estar desocupado. Além disso, o acontecimento desta ação atualiza o valor do custo de transporte de acordo com a Eq. 4.1.

Como o intuito dessa dissertação não é detalhar a modelagem do domínio de planejamento, mas sim definir como as ações planejadas serão implementadas no sistema real, maiores detalhes a respeito da modelagem do domínio de planejamento deste estudo de caso podem ser encontrados em Fonseca e Tavares (2012) ou Fonseca et al. (2012), disponíveis no Anexo A e no Anexo B, respectivamente.

4.2.2. Modelagem do Problema

O problema de transporte a ser resolvido consiste em movimentar, carregar e descarregar o veículo *v1* de forma a atender a demanda dos clientes dos distribuidores. Como caso de teste, escolheu-se analisar a seguinte situação: inicialmente o veículo está estacionado sob o reservatório fornecedor; o distribuidor *a1* contém 20% de estoque disponível sendo o estoque crítico de 20%; enquanto o distribuidor *a2* contém 20% de estoque disponível e um nível crítico de 20%. Além disso, o cliente *c1* possui uma demanda

não atendida de 70%, enquanto o cliente *c2* possui uma demanda não atendida de 100%. O estado final, definido como cenário objetivo do sistema é: veículo estacionado na posição original (fornecedor); demandas de produto dos clientes *c1* e *c2* totalmente atendidas; distribuidores *a1* e *a2* com reservatórios em nível intermediário, especificamente 50%. Na prática, este estoque intermediário dos distribuidores poderia estar associado a uma previsão de demanda para os clientes em um espaço de tempo subsequente. A Fig. 4.7 e a Fig. 4.8 ilustram os cenários inicial e objetivo correspondentes a este problema.

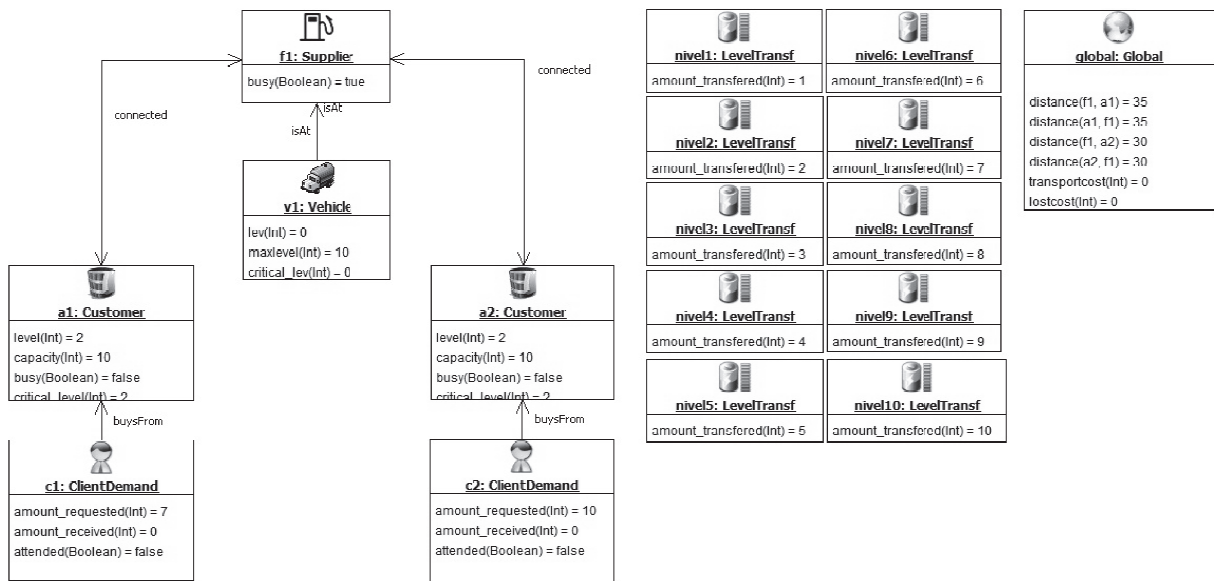


Figura 4.7. *Snapshot* inicial (FONSECA et al., 2012).

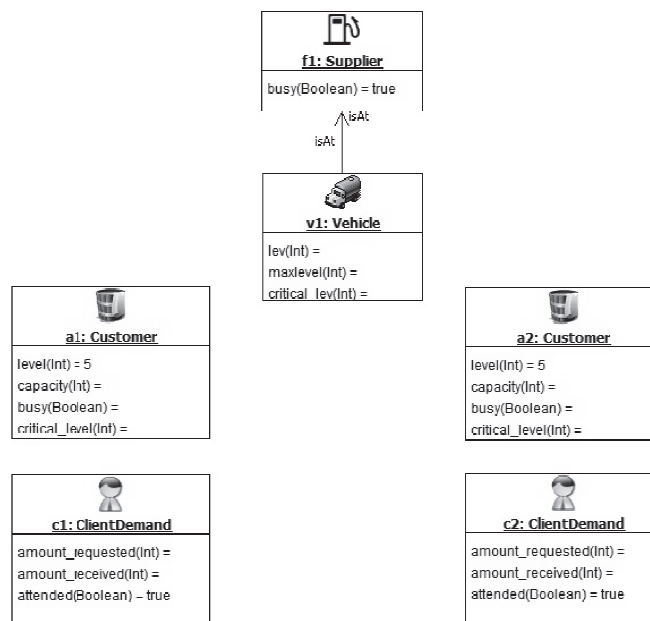


Figura 4.8. *Snapshot* objetivo (FONSECA et al., 2012).

4.3. Resposta dos Planejadores

Conforme já mencionado, o sistema itSIMPLE possui um número considerável de planejadores automáticos embarcados, tais como SGPlan6 (HSU; WAH, 2008), Metric-FF (HOFFMANN, 2003), MIPS-XXL (EDELKAMP; JABBAR, 2008), LPG-td (GEREVINI et al., 2004) e HSP (BONET; GEFFNER, 2000). Neste trabalho será apresentado o resultado obtido com o planejador SGPlan6, executado na distribuição Ubuntu do sistema operacional Linux. O “plano-solução” referente ao problema proposto está ilustrado na Fig. 4.9.

- | | |
|--------------------------------|--------------------------------|
| 1. LOAD V1 F1 LEVEL8 | 11. MOVE V1 F1 A1 |
| 2. MOVE V1 F1 A1 | 12. UNLOAD V1 A1 LEVEL2 |
| 3. UNLOAD V1 A1 LEVEL8 | 13. MOVE V1 A1 F1 |
| 4. MOVE V1 A1 F1 | 14. PARTIAL_ORDER A2 C2 LEVEL8 |
| 5. LOAD V1 F1 LEVEL8 | 15. LOAD V1 F1 LEVEL5 |
| 6. MOVE V1 F1 A2 | 16. MOVE V1 F1 A2 |
| 7. UNLOAD V1 A2 LEVEL8 | 17. UNLOAD V1 A2 LEVEL5 |
| 8. MOVE V1 A2 F1 | 18. MOVE V1 A2 F1 |
| 9. COMPLETE_ORDER A1 C1 LEVEL7 | 19. FINAL_ORDER A2 C2 LEVEL2 |
| 10. LOAD V1 F1 LEVEL2 | |

Figura 4.9. Plano solução obtido com o planejador SGPlan6 (FONSECA et al., 2012).

4.4. Configuração do sistema PlanPAS

A configuração do sistema PlanPAS é necessária para cada caso específico. As principais informações necessária para realizar esta configuração diz respeito à tabela de relacionamento entre as ações planejadas e as saídas digitais do CLP, e aos parâmetros de acesso aos endereços de E/S do CLP via servidor OPC. No caso da bancada didática, inicialmente definiu-se o relacionamento entre os sensores e atuadores do sistema com os respectivos endereços de E/S do CLP. Na sequência, a *PLC Interface* foi configurada para o problema, através da definição dos parâmetros de acesso aos dispositivos de campo. A Tab. 4.1 relaciona todos os dispositivos de campo do sistema com seus respectivos endereços no módulo de E/S do CLP e o seu parâmetro de acesso via servidor OPC. Por exemplo, o dispositivo de campo responsável por abastecer o veículo, denominado “Bomba 1”, está associado com o endereço “Q1.00” do módulo de E/S do CLP, e, de acordo com a

configuração do servidor OPC, o acesso a este endereço se dá através do parâmetro “.Output1”. Todos os sensores e atuadores da bancada foram associados com os parâmetros de acesso no servidor OPC, assim como mostrado na Tab. 4.1.

A definição da tabela de sensores e atuadores do sistema facilita a elaboração da tabela de relacionamento carregada ou escrita diretamente na interface do PlanPAS. Como antecipado anteriormente, a tabela de relacionamento, ou “tabela de-para”, contém todas as ações possíveis do domínio que se deseja aplicar ao PlanPAS. Nesse caso, um levantamento das características do domínio permitiu mapear um total de 127 ações possíveis de ocorrer em algum momento no sistema. Todas estas ações devem ser relacionadas com um atuador correspondente e os seus respectivos parâmetros de acesso via servidor OPC. A Tab. 4.2 apresenta uma parte da “tabela de-para” referente ao domínio da bancada didática estudada. A tabela de relacionamento completa está disponível no Apêndice B.

Tabela 4.1. Tabela de Sensores e atuadores.

Dispositivo de campo (detalhe do dispositivo)	Endereço no Módulo de E/S	Parâmetro de acesso no servidor OPC
Bomba 1 (abastece o veículo)	Q1.00	.Output1
Bomba 2 (descarrega o veículo e abastece distribuidor 1 ou 2)	Q1.01	.Output2
Motores (move veículo para a esquerda)	Q1.02	.Output3
Motores (move veículo para o lado direito)	Q1.03	.Output4
Bomba 3 (abastece clientes do distribuidor1)	Q1.04	.Output5
Bomba 4 (abastece clientes do distribuidor2)	Q1.05	.Output6
SN1 (sensor de nível do veículo)	I2011	Entradas.AIn1
SN2 (sensor de nível do distribuidor1)	I2012	Entradas.AIn2
SN3 (sensor de nível do distribuidor2)	I2013	Entradas.AIn3
FC1 (veículo posicionado no fornecedor)	I0.00	Entradas.DIn1
FC2 (veículo posicionado no distribuidor1)	I0.01	Entradas.DIn2
FC3 (veículo posicionado no distribuidor2)	I0.02	Entradas.DIn3

Tabela 4.2. “Tabela de-para” simplificada para o estudo de caso analisado.

Ação	Parâmetro de acesso no servidor OPC	Atuador correspondente
MOVE V1 F1 A2	.Output4	Motores para Direita
MOVE V1 A1 F1	.Output4	Motores para Direita
MOVE V1 A1 A2	.Output4	Motores para Direita
MOVE V1 F1 A1	.Output3	Motores para Esquerda
MOVE V1 A2 F1	.Output3	Motores para Esquerda
MOVE V1 A2 A1	.Output3	Motores para Esquerda
LOAD V1 F1 LEVEL8	.Output1	Bomba 1
LOAD V1 F1 LEVEL7	.Output1	Bomba 1
(...)	(...)	(...)
UNLOAD V1 A1 LEVEL8	.Output2	Bomba 2
UNLOAD V1 A1 LEVEL7	.Output2	Bomba 2
(...)	(...)	(...)
UNLOAD V1 A2 LEVEL8	.Output2	Bomba 2
UNLOAD V1 A2 LEVEL7	.Output2	Bomba 2
(...)	(...)	(...)
PARTIAL_ORDER A1 C1 LEVEL8	.Output5	Bomba 3
PARTIAL_ORDER A1 C2 LEVEL8	.Output5	Bomba 3
FINAL_ORDER A1 C1 LEVEL8	.Output5	Bomba 3
FINAL_ORDER A1 C2 LEVEL8	.Output5	Bomba 3
COMPLETE_ORDER A1 C1 LEVEL8	.Output5	Bomba 3
COMPLETE_ORDER A1 C2 LEVEL8	.Output5	Bomba 3
PARTIAL_ORDER A1 C1 LEVEL7	.Output5	Bomba 3
PARTIAL_ORDER A1 C2 LEVEL7	.Output5	Bomba 3
FINAL_ORDER A1 C1 LEVEL7	.Output5	Bomba 3
FINAL_ORDER A1 C2 LEVEL7	.Output5	Bomba 3
COMPLETE_ORDER A1 C1 LEVEL7	.Output5	Bomba 3
COMPLETE_ORDER A1 C2 LEVEL7	.Output5	Bomba 3
(...)	(...)	(...)
PARTIAL_ORDER A2 C1 LEVEL8	.Output6	Bomba 4
PARTIAL_ORDER A2 C2 LEVEL8	.Output6	Bomba 4
FINAL_ORDER A2 C1 LEVEL8	.Output6	Bomba 4

FINAL_ORDER A2 C2 LEVEL8	.Output6	Bomba 4
COMPLETE_ORDER A2 C1 LEVEL8	.Output6	Bomba 4
COMPLETE_ORDER A2 C2 LEVEL8	.Output6	Bomba 4
PARTIAL_ORDER A2 C1 LEVEL7	.Output6	Bomba 4
PARTIAL_ORDER A2 C2 LEVEL7	.Output6	Bomba 4
FINAL_ORDER A2 C1 LEVEL7	.Output6	Bomba 4
FINAL_ORDER A2 C2 LEVEL7	.Output6	Bomba 4
COMPLETE_ORDER A2 C1 LEVEL7	.Output6	Bomba 4
(...)	(...)	(...)

Os dados da Tab. 4.2 devem ser inseridos na Tabela de Relacionamentos do PlanPAS, habilitando assim a sua operação no domínio da bancada didática. Esta inserção pode ser feita manualmente, linha por linha, pelo usuário ou então, caso a tabela já tenha sido armazenada em um arquivo de extensão .XML ou .TXT, o usuário poderá carregá-la automaticamente para a interface do PlanPAS.

Para finalizar a configuração da interface do PlanPAS, o usuário deve ainda carregar o arquivo XML com o “plano-solução” para habilitar o acionamento dos atuadores do sistema via *PLC Interface* e via servidor OPC. A partir das informações armazenadas no arquivo que contém o “plano-solução” é que a interface do PlanPAS consegue atualizar a tabela de monitoração para acompanhamento do processo pelo usuário. A Fig. 4.10 apresenta a interface do PlanPAS configurada para o domínio da bancada didática. Note que a tabela de monitoramento na parte inferior da interface contém a representação dos sensores (fins de curso e sensores de nível) instalados na bancada. Esta estruturação foi realizada especificamente para este domínio, caso o PlanPAS estivesse sendo utilizado em um outro domínio, uma reestruturação da tabela de monitoramento seria necessária. Um ensaio realizado com a interface do PlanPAS, rodando lado a lado com a *PLC Interface* e um cliente OPC, é mostrado na Fig. 4.11. As informações da *PLC Interface* foram confrontadas com as informações do cliente OPC comercial para validação do seu funcionamento.

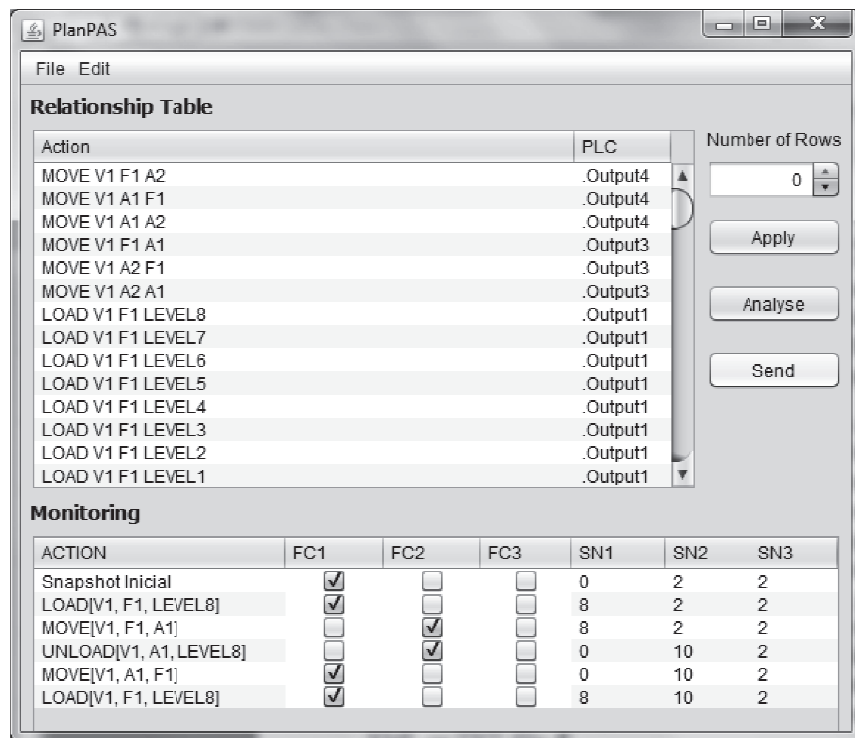


Figura 4.10. Interface do PlanPAS configurada para implementar este estudo de caso.

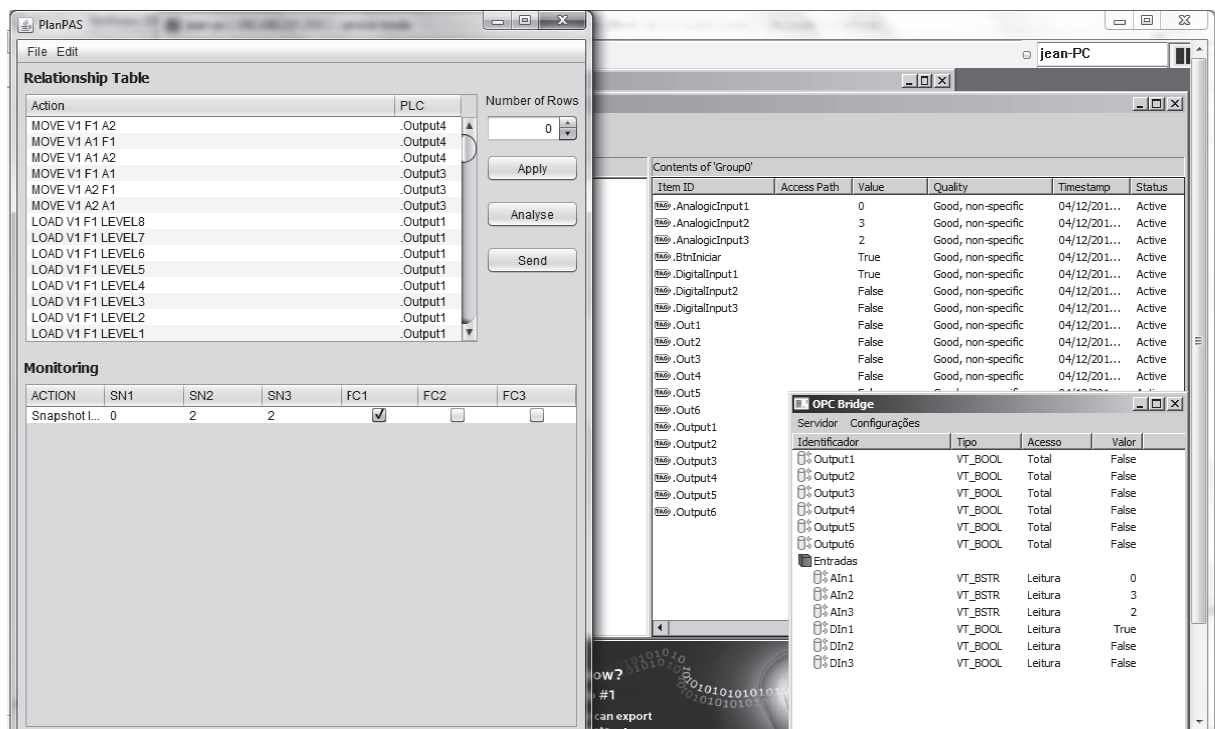


Figura 4.11. Parser, PLC Interface e Matrikon OPC Client rodando lado a lado.

4.5. Configuração do CLP como Driver e Intertravamentos

Além da configuração da interface do PlanPAS é também necessária a configuração do CLP para o domínio do problema que será resolvido pelas ações do “plano-solução”. O principal objetivo da configuração do CLP é permitir que as *tags* utilizadas como intermediário no servidor OPC sejam capazes de habilitar as saídas digitais correspondentes aos atuadores responsáveis pelas ações planejadas. Porém, a configuração do CLP não se restringe somente na definição do método de acesso às saídas digitais, mas também às estratégias auxiliares para promover a correta operação do sistema. Dentre estas estratégias auxiliares destacam-se: o processo de discretização dos sensores potenciométricos – responsáveis por indicar o nível dos reservatórios, os intertravamentos para resguardar a segurança operacional do sistema e uma lógica auxiliar utilizada para que o veículo busque a sua posição inicial antes do início de cada plano.

O relacionamento entre os dispositivos de campo, seus endereços no módulo de E/S e seus parâmetros de acesso via servidor OPC já foram apresentados na Tab. 4.1, no entanto, o servidor OPC não possui acesso de escrita direta nos módulos de E/S do CLP. Sendo assim, a alternativa para habilitar os atuadores via módulo de saídas do CLP é através do acesso à uma memória auxiliar via servidor OPC. No diagrama *Ladder*, esta memória auxiliar deve ser inserida como um contato normalmente aberto na linha de habilitação de uma determinada saída digital, operando então como um *driver* para habilitação desta saída. A Tab. 4.3 apresenta o relacionamento entre os parâmetros de acesso no servidor OPC e as saídas digitais a serem habilitadas via memória auxiliar no CLP. A Fig. 4.12 mostra o programa principal para implementação do “plano-solução” no sistema real.

Tabela 4.3. Relacionamento entre o parâmetro de acesso no servidor OPC e a saída digital a ser habilitada via memória auxiliar no CLP.

Parâmetro de acesso no servidor OPC	Memória auxiliar como <i>driver</i> no CLP	Saída digital a ser habilitada
.Output1	A900.00	Q1.00
.Output2	A900.01	Q1.01
.Output3	A900.02	Q1.02
.Output4	A900.03	Q1.03
.Output5	A900.04	Q1.04
.Output6	A900.05	Q1.05

Neste programa destacam-se as linhas de habilitação das saídas digitais, bem como seus disparadores e intertravamentos. A memória auxiliar A900.00 é o disparador da saída digital Q1.00, correspondente ao acionamento da bomba B1 (Fornecedor), conforme mostrado na Tab. 4.1.

O acionamento da bomba B1 está condicionado, ainda, ao seguinte intertravamento: I0.00 NA, Q1.01, Q1.02 e Q1.03 ambos NF. O contato I0.00 NA indica que o acionamento da bomba de fornecimento está condicionado ao estado “alto” do sensor de fim de curso FC1, ou seja, o veículo deve estar estacionado sob o reservatório fornecedor. O contato Q1.01 NA indica que, para que a bomba de fornecimento seja acionada, a bomba do veículo (B2) deve estar desligada; os contatos Q1.02 e Q1.03 NF indicam que os motores de movimentação do veículo devem estar em estado “baixo”, ou seja, o veículo não pode estar em movimento. A redundância entre os contatos I0.00 NA e Q1.02 e Q1.03 NF garante maior segurança ao sistema. A linha 48 da Figura 4.12 mostra a lógica *Ladder* de acionamento da bomba B1.

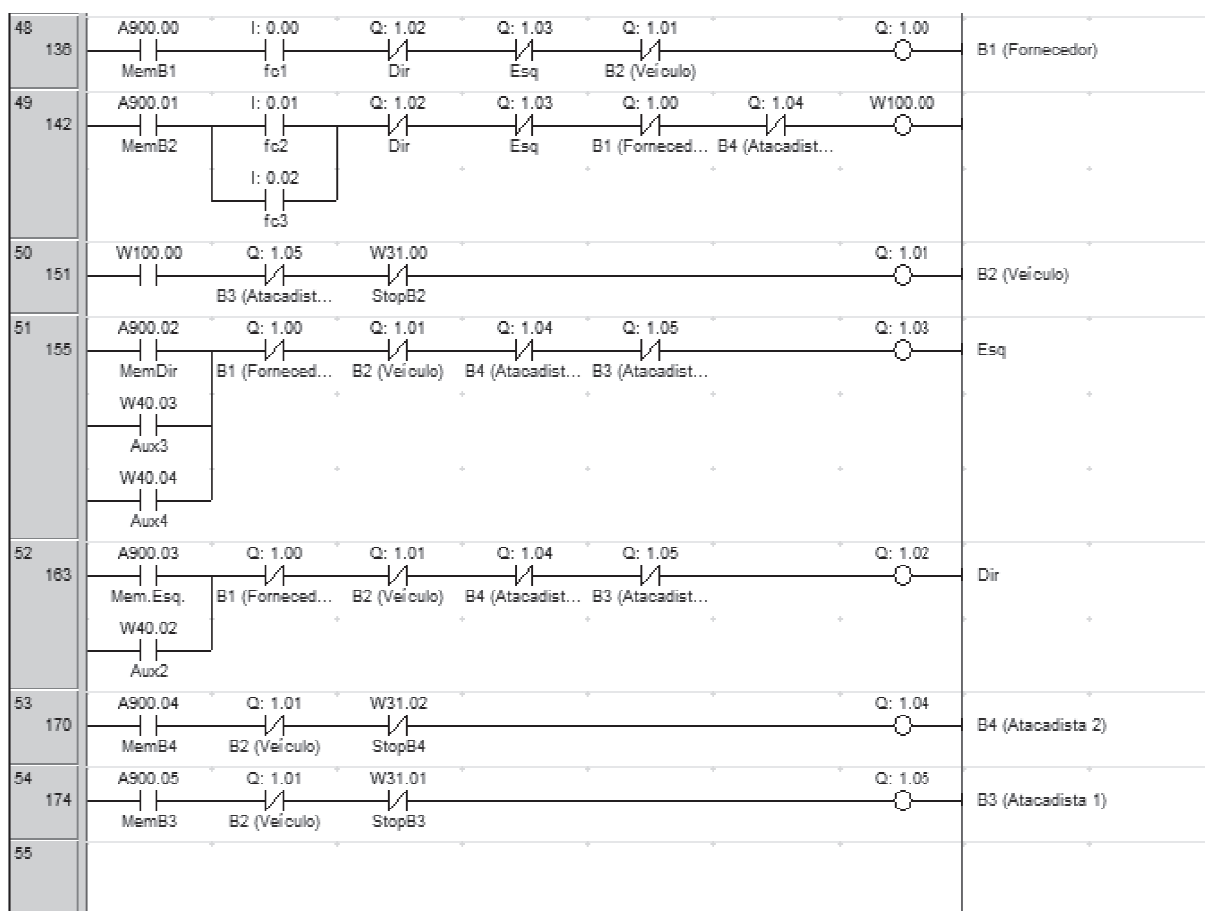


Figura 4.12. Programa principal.

Para habilitar a saída digital Q1.01 (B2 (veículo)), tem-se a memória auxiliar A900.01 como disparador. Para acionar a bomba B2 é necessário ainda que o veículo esteja estacionado em algum dos reservatórios distribuidores, ou seja, estado “alto” para o sensor FC2 ou FC3 e motores do veículo desligados, e que as bombas de abastecimento do veículo (B1) e de entrega de produtos para os clientes dos distribuidores (B3 e B4) estejam desligadas. O contato W31.00 NF corresponde ao nível de crítico do reservatório do veículo. A linha 50 da Figura 4.12 apresenta a lógica correspondente em *Ladder*.

O acionamento dos motores do veículo está condicionado à não habilitação de nenhuma das eletrobombas do sistema, ou seja, contatos NF para Q1.00, Q1.01, Q1.04 e Q1.05. Os contatos A900.02 e A900.03 NA são os disparadores dos motores para esquerda e para direita, respectivamente. As memórias de trabalho W40.02, W40.03 e W40.04 foram programadas de acordo com a Fig. 4.15 para garantir autonomia ao veículo na busca pela sua posição inicial. As linhas 51 e 52 da Figura 4.12 ilustram a lógica *Ladder* para movimentação do veículo para a esquerda e direita, respectivamente.

Para acionar a bomba de entrega de produtos para os clientes do distribuidor1 (B3), tem-se o habilitador A900.05. Além disso, não é necessário que o veículo esteja estacionado; a única condição para acionar a saída digital Q1.05 é que o veículo não esteja realizando nenhuma entrega para distribuidores, ou seja, saída Q1.01 desabilitada. A bomba de entrega de produtos para os clientes do distribuidor 2 (B4) é habilitada pelo contato NA da memória auxiliar A900.04. Assim como na bomba B3, a única condição para acionar a saída digital Q1.04 é que o veículo não esteja realizando nenhuma entrega para distribuidores. Os contatos W31.01 e W31.02 NF correspondem ao nível de segurança dos reservatórios distribuidores. As linhas 53 e 54 da Figura 4.12 apresentam a lógica *Ladder* de acionamento da bomba B4 e da bomba B3, respectivamente.

Sabendo que uma das premissas básicas para o planejamento clássico é a de sistema determinístico, a característica contínua dos sensores potenciométricos foi um obstáculo para a abordagem sugerida. Para tornar o sistema representado pela bancada passível de ser representado em um modelo para planejamento clássico, optou-se pela discretização das variáveis que indicavam o nível dos reservatórios, este processo dividiu o range dos sensores em 10 partes iguais, definidas de 0% a 100% da capacidade máxima de cada reservatório. Para sistematizar este processo, criou-se um programa em *Ladder* que verifica em qual das 10 divisões se encontra cada sensor de nível e armazena o valor correspondente em uma memória de dados correspondente a cada sensor. O servidor OPC terá, então, acesso de leitura a estas três memórias de dados que conterão, ao invés do valor real dos sensores, algo dentro do intervalo [0 4200], um valor inteiro dentro do intervalo [0 10], onde zero indica que o reservatório está totalmente vazio e 10 indica que o

reservatório está com carga máxima. A Fig. 4.13 ilustra uma parte do programa utilizado para a discretização dos níveis dos sensores potenciométricos. A lógica completa para este processo está disponível no diagrama *Ladder* completo apresentado no Apêndice C.

A segunda estratégia auxiliar de destaque é o sistema de intertravamentos, utilizado para garantir uma operação segura ao sistema. A Fig. 4.14 apresenta os intertravamentos aplicados para desligar as eletrobombas sempre que os sensores de nível indicarem valores abaixo de um valor de segurança, evitando que as mesmas operem em vazio. As memórias de trabalho W31.00, W31.01 e W31.02 aparecerão como contatos normalmente fechados nas linhas de habilitação das saídas digitais responsáveis pelo disparo das eletrobombas do veículo e dos distribuidores. Os demais intertravamentos, tais como: impedir que o veículo se movimente durante uma ação de carga ou descarga, ou ainda, impedir que um distribuidor execute entregas para clientes durante o seu processo de abastecimento, foram inseridos diretamente no programa principal (Fig. 4.12) como contatos normalmente fechados nas linhas de habilitação das saídas digitais.

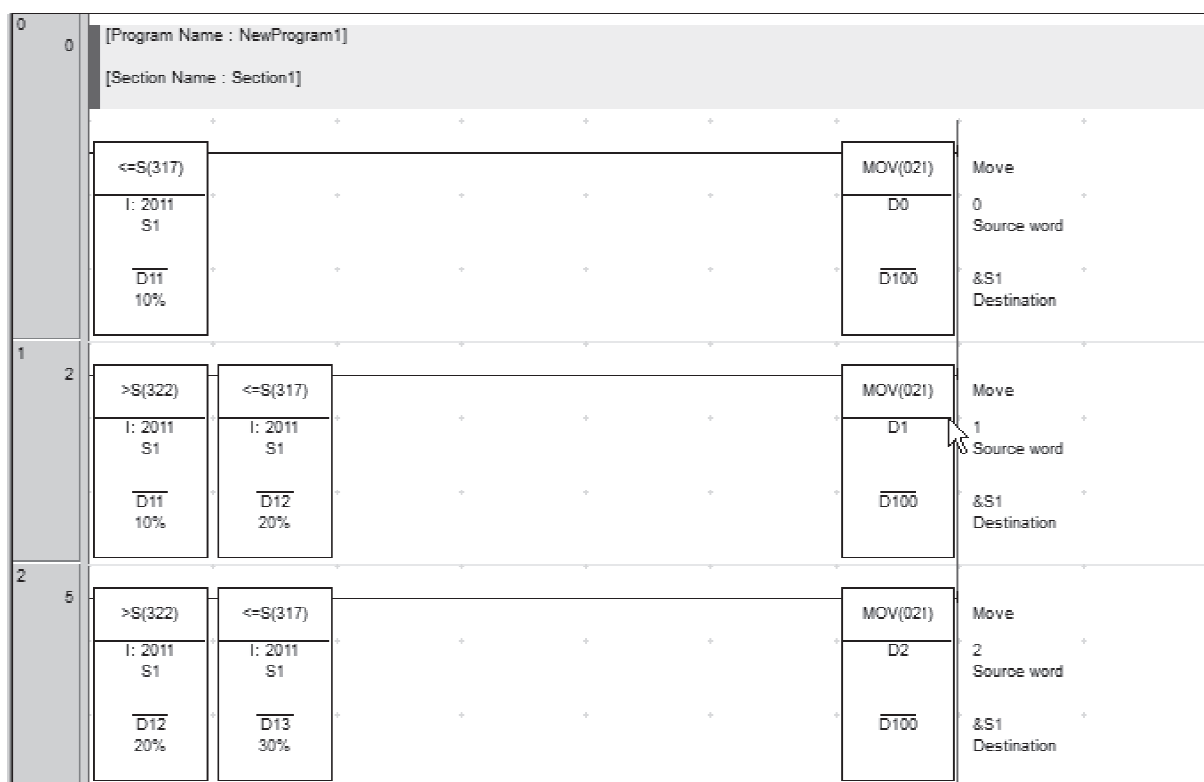


Figura 4.13. Discretização dos sensores de nível.



Figura 4.14. Intertravamentos para níveis de segurança.

Além da discretização dos níveis e definição dos intertravamentos, foi idealizada uma lógica auxiliar que promove certa autonomia ao veículo. Esta lógica permite que o veículo busque sua posição inicial sempre que algum “plano-solução” estiver para ser inicializado. Quando o sistema inicia, caso o veículo não esteja na posição inicial (estacionado sob o reservatório fornecedor), e em nenhum dos distribuidores, os motores do veículo são ligados movendo-o para a esquerda até que o mesmo atinja uma chave fim-de-curso. Caso a nova posição ocupada pelo veículo não seja sob o reservatório fornecedor, o sentido de rotação dos motores é invertido e o veículo move-se para a direita até atingir a chave fim-de-curso posicionada sob o reservatório fornecedor. A Fig. 4.15 apresenta o programa auxiliar descrito. As memórias de trabalho W40.02, W40.03 e W40.04 aparecem no programa principal (Fig. 4.12) como contatos normalmente abertos, em paralelo com as memórias auxiliares acessadas via servidor OPC, nas linhas de habilitação das saídas digitais responsáveis pelo giro dos motores para esquerda ou direita.

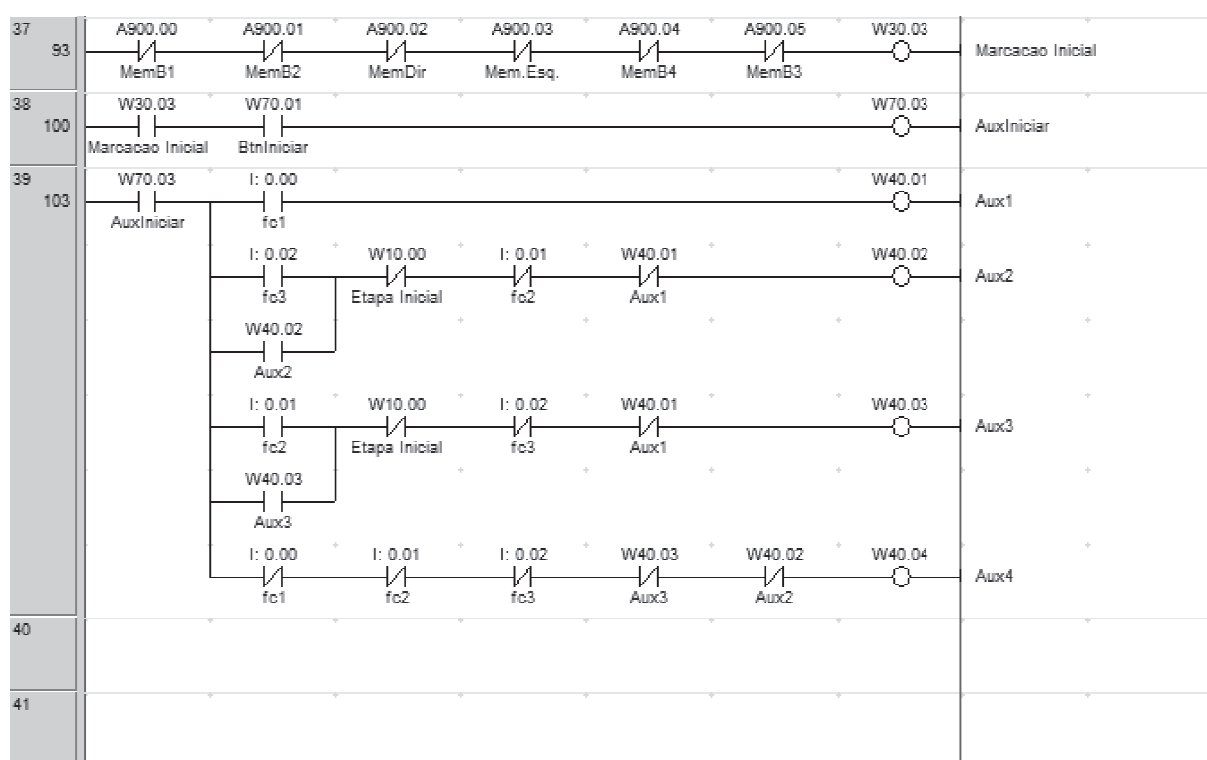


Figura 4.15. Programa auxiliar.

Conforme já informado, o diagrama *Ladder* completo para esta configuração da bancada didática pode ser visualizado em detalhes no Apêndice C desta dissertação.

4.6. Análise do Sistema usando a Abordagem Clássica IEC 61131-3

Como método de comparação com a abordagem sugerida, de aplicação direta de soluções de planejamento automático, optou-se por analisar o sistema em estudo, também, de acordo com método tradicional encontrado na literatura. A abordagem tradicional consiste, inicialmente, na modelagem do problema em SFC e, posteriormente, na construção do diagrama *Ladder* correspondente ao modelo em SFC, assim como apresentado na seção 2.1.3.

Nessa abordagem clássica, optou-se por um controle de nível dos reservatórios distribuidores praticamente “on/off”, onde um setup de 20% da capacidade do reservatório indica nível baixo para o distribuidor, que solicita uma entrega de produto por parte do veículo, entrega esta feita sempre no sentido de completar o reservatório até seu nível cheio, no caso, 90% da capacidade máxima). Esta opção facilita a modelagem e implementação, porém métodos mais complexos de controle poderiam ser utilizados, tais como o controle PI (proporcional-integral) ou PID (proporcional-integral-derivativo).

Na abordagem específica, proporcionada pelo método com planejamento automático, o nível dos reservatórios pôde ser discretizado em 10 subníveis e as entregas de produtos contaram com uma flexibilidade maior.

O modelo em SFC da bancada didática utilizada como estudo de caso neste trabalho pode ser visualizado na Fig. 4.16. Nas ocasiões onde os sensores de nível dos reservatórios distribuidores indicam nível baixo, simultaneamente, o modelo deve assumir a prioridade de atender um distribuidor ou outro – no modelo abordado, esta prioridade foi ofertada ao *distribuidor 1*. Quando o nível do *distribuidor 1* estiver em nível baixo, ele será atendido, independentemente do nível do *distribuidor 2*. Já o *distribuidor 2* só será atendido quando estiver com nível baixo e o nível do *distribuidor 1* for alto. As operações de entrega para os clientes não são previstas neste modelo, sendo implementadas na prática através do acionamento manual das eletrobombas B3 e B4.

A relação entre os sensores e atuadores da bancada com as entradas e saídas do CLP está mostrada na Tab. 4.4. Já as informações das posições de memória referentes às etapas e transições do diagrama *Ladder* desenvolvido e utilizado no projeto podem ser visualizadas na Tab. 4.5 e na Tab. 4.6, respectivamente.

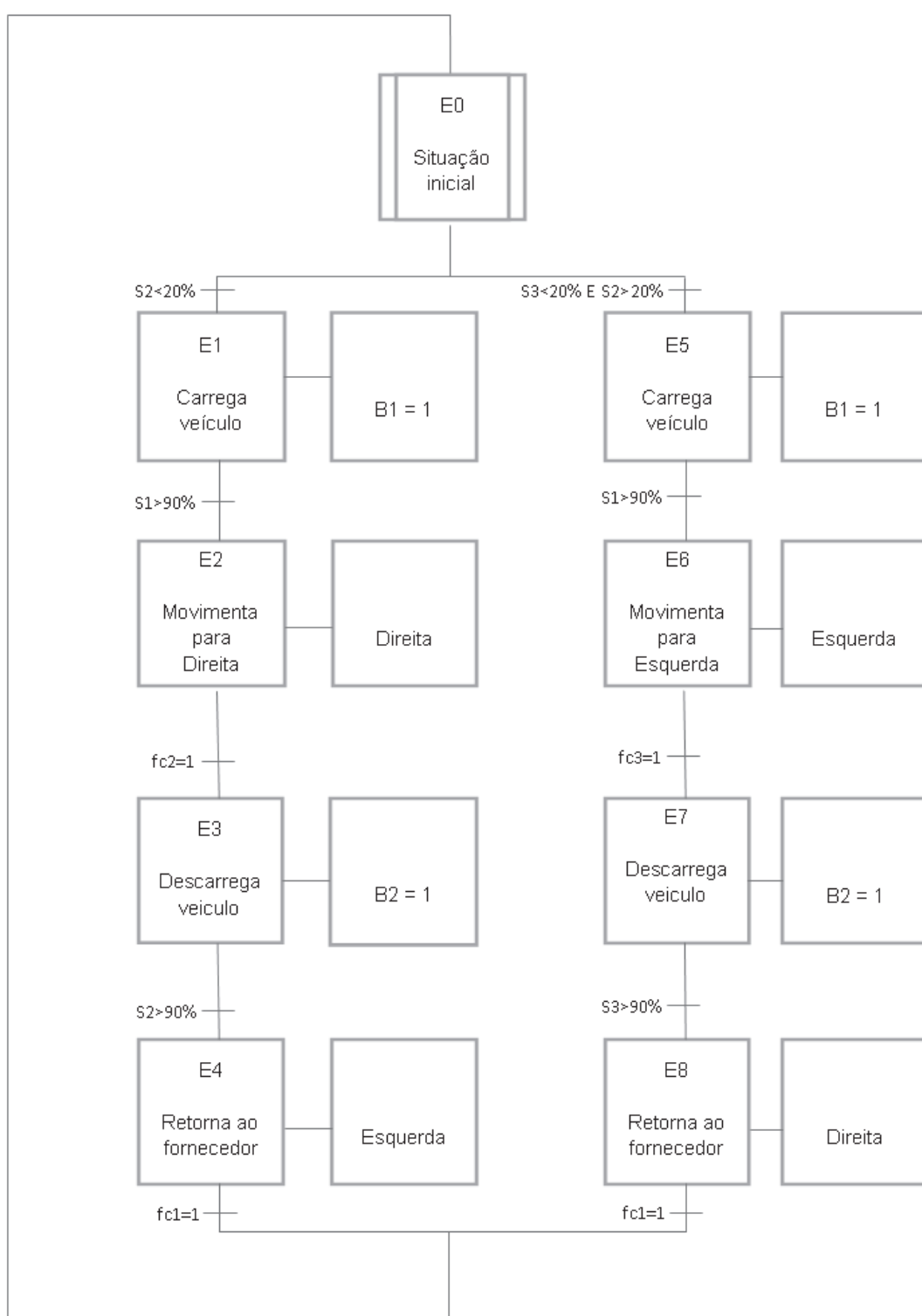


Figura 4.16. SFC genérico do sistema real

O SFC ilustrado na Fig. 4.16 é a base para o desenvolvimento do diagrama *Ladder* utilizado no CLP. Este diagrama foi desenvolvido através do software *CX-Programer*, (OMRON CORPORATION, 2001) em um microcomputador e transferido para o CLP *Omron CJ1M CPU13 ETN21* via comunicação serial. Como a sua apresentação é longa, e não é o

principal objeto de estudo deste trabalho, o diagrama *Ladder* completo referente à abordagem clássica com padrão IEC 61131-3 foi alocado no Apêndice D. A Fig. 4.17 ilustra parcialmente o diagrama em questão.

Tabela 4.4. Tabela de dispositivos de campo vs endereço no módulo de E/S.

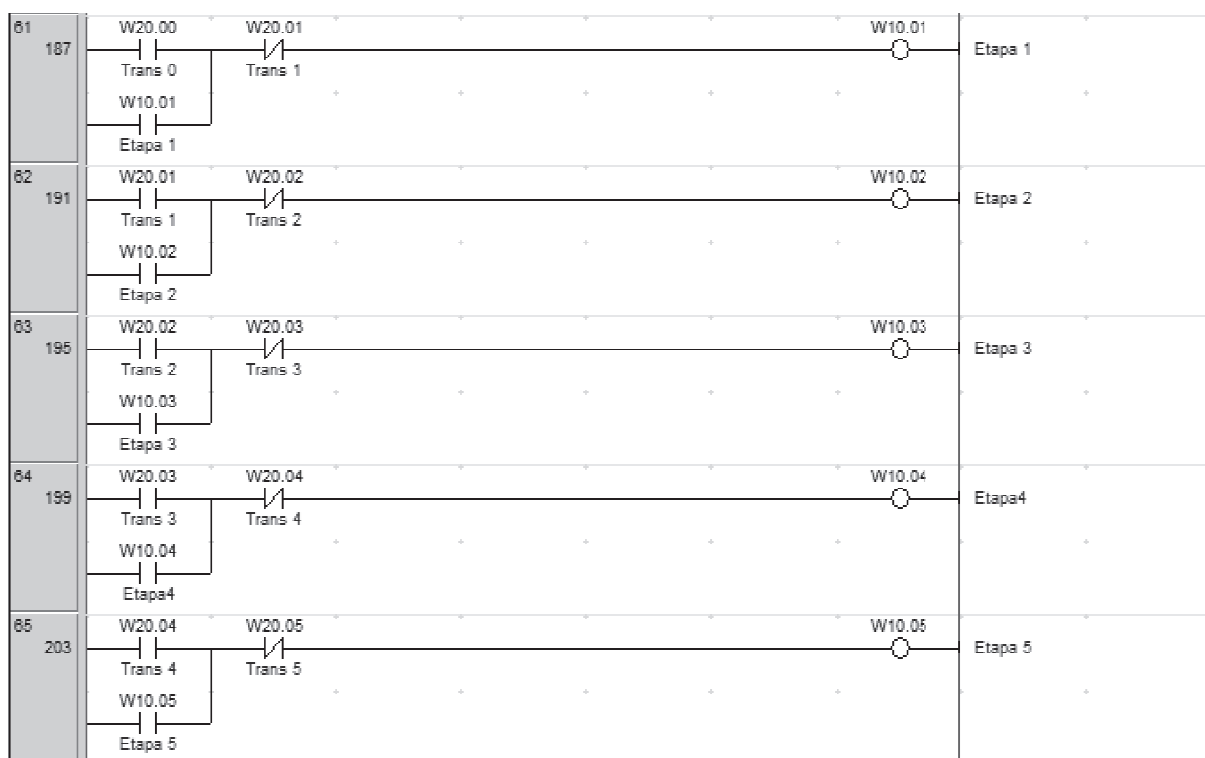
Dispositivo de campo (detalhe do dispositivo)	Endereço no Módulo de E/S
Bomba 1 (abastece o veículo)	Q1.00
Bomba 2 (descarrega o veículo e abastece distribuidor 1 ou 2)	Q1.01
Motores (move veículo para a esquerda)	Q1.02
Motores (move veículo para o lado direito)	Q1.03
Bomba 3 (abastece clientes do distribuidor1)	Q1.04
Bomba 4 (abastece clientes do distribuidor2)	Q1.05
SN1 (sensor de nível do veículo)	I2011
SN2 (sensor de nível do distribuidor1)	I2012
SN3 (sensor de nível do distribuidor2)	I2013
FC1 (veículo posicionado no fornecedor)	I0.00
FC2 (veículo posicionado no distribuidor1)	I0.01
FC3 (veículo posicionado no distribuidor2)	I0.02

Tabela 4.5. Tabela Etapa vs Memória de trabalho no CLP.

Etapa	Endereço Memória de Trabalho no CLP	Comentário
E0	W10.00	Etapa Inicial
E1	W10.01	Etapa 1
E2	W10.02	Etapa 2
E3	W10.03	Etapa 3
E4	W10.04	Etapa 4
E5	W10.05	Etapa 5
E6	W10.06	Etapa 6
E7	W10.07	Etapa 7
E8	W10.08	Etapa 8

Tabela 4.6. Tabela Transição vs Memória de trabalho no CLP.

Transição	Endereço Memória de Trabalho no CLP	Comentário
Entre E0 e E1	W20.00	Trans 0
Entre E1 e E2	W20.01	Trans 1
Entre E2 e E3	W20.02	Trans 2
Entre E3 e E4	W20.03	Trans 3
Entre E4 e E0	W20.04	Trans 4
Entre E0 e E5	W20.05	Trans 5
Entre E5 e E6	W20.06	Trans 6
Entre E6 e E7	W20.07	Trans 7
Entre E7 e E8	W20.08	Trans 8
Entre E8 e E0	W20.09	Trans 9

Figura 4.17. Diagrama *Ladder* parcial de acordo com a abordagem clássica.

As análises e discussões a respeito dos resultados encontrados, tanto com a abordagem proposta, utilizando planejamento automático via PlanPAS, bem como com a abordagem clássica utilizando o padrão IEC 61131-3 são ponderadas no próximo capítulo dessa dissertação.

CAPÍTULO V

RESULTADOS E ANÁLISES

O Capítulo 3 desta dissertação apresentou o projeto do PlanPAS, explanado toda sua estrutura de construção e operação. Para validar a funcionalidade do aplicativo proposto, o Capítulo 4 expôs um estudo de caso utilizando o PlanPAS para atuar uma solução de planejamento automático diretamente em um sistema controlado por CLP. Este capítulo descreve os resultados obtidos e suas análises. A seção 5.1 apresenta os resultados obtidos nos ensaios, com abordagem clássica e com planejamento automático, enquanto a seção 5.2 levanta as análises para cada abordagem.

5.1. Resultados

Além do método de ação proposto, optou-se por utilizar como forma de comparação a método clássico baseado nas linguagens (*Ladder* e SFC) de programação de CLPs da norma IEC 61131-3. Para isso, programou-se o CLP de acordo com os diagramas *Ladder* ilustrados no Apêndice C (Fig. C.1), para planejamento automático, e no Apêndice D (Fig. D.1), para a abordagem clássica. A Tab. 5.1 apresenta a evolução do sistema para a abordagem utilizando PA, enquanto que a evolução do sistema para a abordagem clássica é apresentada na Tab. 5.2.

Na abordagem por Planejamento Automático (PA) foi executado o “plano-solução” com 19 ações apresentado na Fig. 4.9 em um tempo total de 13 minutos e 46 segundos. Já na abordagem clássica, foram necessários 13 minutos e 57 segundos para a execução de um total de 15 ações. Em PA as ações de entrega para os clientes são automáticas, de acordo com o “plano-solução”. Porém, na abordagem clássica, como o modelo em SFC apresentado na Fig. 4.16 não prevê demanda de clientes, as operações de entrega (*PartialOrder*, *FinalOrder* e *CompleteOrder*) foram realizadas manualmente. Ambos os

ensaios atenderam a demanda dos clientes. No entanto, somente em PA o estado final do sistema (definido na Fig. 4.8) foi alcançado após a última operação dos atuadores.

Tabela 5.1. Evolução do sistema para a abordagem utilizando PA.

Ação	Estado	SN1	SN2	SN3	fc1	fc2	fc3	Cronômetro (min:seg)
	S0	2	2	2	1	0	0	00:00
1: (Load v1 f1 level 8)								
	S1	10	2	2	1	0	0	01:30
2: (Move v1 f1 a1)								
	S2	10	2	2	0	1	0	01:42
3: (Unload v1 a1 level 8)								
	S3	2	10	2	0	1	0	03:36
4: (Move v1 a1 f1)								
	S4	2	10	2	1	0	0	03:47
5: (Load v1 f1 level 8)								
	S5	10	10	2	1	0	0	05:24
6: (Move v1 f1 a2)								
	S6	10	10	2	0	0	1	05:35
7: (Unload v1 a2 level 8)								
	S7	2	10	10	0	0	1	07:37
8: (Move v1 a2 f1)								
	S8	2	10	10	1	0	0	07:47
9: (CompleteOrder a1 c1 nivel 7)								
	S9	2	3	10	1	0	0	08:24
10: (Load v1 f1 level 2)								
	S10	4	3	10	1	0	0	09:10
11: (Move v1 f1 a1)								
	S11	4	3	10	0	1	0	09:22
12: (Unload v1 a1 level 2)								
	S12	2	5	10	0	1	0	09:49
13: (Move v1 a1 f1)								
	S13	2	5	10	1	0	0	09:59
14: (PartialOrder a2 c2 nivel 8)								
	S14	2	5	2	1	0	0	10:45
15: (Load v1 f1 level 5)								
	S15	7	5	2	1	0	0	11:27
16: (Move v1 f1 a2)								

	S16	7	5	2	0	0	1	11:37
17: (Unload v1 a2 level 5)								
	S17	2	5	7	0	0	1	13:13
18: (Move v1 a2 f1)								
	S18	2	5	7	1	0	0	13:24
19: (Final_order a2 c2 level 2)								
	S19	2	5	5	1	0	0	13:46

Tabela 5.2. Evolução do sistema para a abordagem clássica.

Ação	Estado	SN1	SN2	SN3	fc1	fc2	fc3	Cronômetro (min:seg)
	S0	2	2	2	1	0	0	00:00
1: (Load v1 f1 level 8)								
	S1	10	2	2	1	0	0	01:23
2: (Move v1 f1 a2)								
	S2	10	2	2	0	1	0	01:34
3: (Unload v1 a2 level 8)								
	S3	2	2	10	0	1	0	03:21
4: (Move v1 a2 f1)								
	S4	2	2	10	1	0	0	03:31
5: (Load v1 f1 level 8)								
	S5	10	2	10	1	0	0	05:00
6: (Move v1 f1 a1)								
	S6	10	2	10	0	0	1	05:11
7: (Unload v1 a1 level 8)								
	S7	2	10	10	0	0	1	06:55
8: (Move v1 a1 f1)								
	S8	2	10	10	1	0	0	07:06
9: (CompleteOrder a1 c1 nivel 7)								
	S9	2	3	10	1	0	0	07:50
10: (PartialOrder a2 c2 nivel 8)								
	S10	2	3	2	1	0	0	08:09
11: (Load v1 f1 level 8)								
	S11	10	3	2	0	1	0	09:10
12: (Move v1 f1 a2)								
	S12	10	3	2	0	1	0	10:36
13: (Unload v1 a2 level 8)								
	S13	2	3	10	1	0	0	10:49

14: (Move v1 a2 f1)								
	S14	2	3	10	1	0	0	12:59
15: (Final_order a2 c2 level 2)								
	S15	2	3	8	1	0	0	13:57

5.2. Análise dos Resultados

Os ensaios permitiram que não somente os resultados, mas também a implantação e operação das duas abordagens, PA e clássica, fossem confrontadas. A primeira análise diz respeito ao processo de aplicação da solução: as etapas de implementação dos dois métodos de abordagem foram avaliadas comparativamente. Basicamente, as etapas de implementação analisadas foram as seguintes: levantamento e análise dos requisitos, modelagem, etapa intermediária entre a modelagem e a programação do CLP, programação do CLP e execução.

A etapa de levantamento e análise dos requisitos refere-se à primeira etapa do projeto, tanto na abordagem com PA, via itSIMPLE, planejador e PlanPAS, quanto na abordagem clássica. Com PA, o levantamento e análise dos requisitos está inserido na modelagem do diagrama de casos de uso, o que torna este processo mais sistematizado, facilitando tanto a definição dos requisitos, quanto a sua documentação através das descrições dos casos de uso. Já com a abordagem clássica, fundamentada exclusivamente no CLP, o processo de levantamento e análise dos requisitos é realizado de forma puramente intuitiva, o que dificulta a documentação dos requisitos em si, podendo interferir na evolução das etapas posteriores.

A segunda etapa abordada diz respeito ao processo de modelagem do problema. Utilizando-se PA, o projetista deve modelar o domínio e definir o problema específico através dos diagramas da UML no itSIMPLE. Este processo é caracterizado por uma complexidade elevada, que demanda muito tempo e conhecimento total do domínio. A dificuldade é amplificada para profissionais que desconhecem Engenharia de Software e o processo de modelagem em UML. Com a abordagem clássica, o modelo do domínio é definido através de um SFC, que nada mais é do que um diagrama sequencial de ações. Neste caso, o modelo atende, ciclicamente, problemas genéricos, fundamentados no controle de nível dos reservatórios distribuidores. Como este método de modelagem é bem disseminado entre engenheiros do ambiente industrial, e como o enfoque do trabalho é na aplicação de soluções no nível operacional, a etapa de modelagem se mostra menos

complexa com CLP do que com PA. Para fins quantitativos, define-se esta complexidade com CLP como moderada.

A etapa intermediária entre a modelagem do domínio e a programação do CLP foi assim denominada devido à falta de melhor relação entre as abordagens analisadas. Com PA, esta etapa é dividida em duas distintas: “geração e análise do plano” e “definição dos relacionamentos (‘tabela de-para’)”. A etapa de geração e análise do plano pode ser considerada simples, visto que basta o projetista saber utilizar corretamente o software itSIMPLE e ter conhecimento das características do domínio para análise do “plano-solução”; trata-se de uma etapa relativamente rápida, em virtude principalmente da performance dos planejadores. Por outro lado, dependendo da complexidade do sistema e do número de operações possíveis, a definição dos relacionamentos pode demandar grande esforço físico e consumo de tempo – esta etapa ainda é executada manualmente e está sujeita à explosão de estados. Com programação clássica, esta etapa de transição corresponde à conversão SFC-*Ladder*, com a definição das transições, etapas e ações, de acordo com a abordagem proposta por Silveira e Santos (2003) e apresentada na seção 2.1. A principal característica desta etapa é, apesar de trabalhosa, ser de fácil execução, principalmente para os profissionais de engenharia em ambiente industrial, já familiarizados com o método. No entanto, é fato que programadores experientes abrem mão da abordagem clássica para programar intuitivamente e, na maioria das vezes, sem qualquer documentação. Isso dificulta que outros programadores efetuem possíveis alterações na lógica do sistema.

A quarta etapa analisada corresponde à programação do CLP. Com PA, esta etapa se mostra rápida e intuitiva, orientada à definição do *driver* (memórias auxiliares disparadoras das ações planejadas) e dos intertravamentos para segurança operacional. Já na abordagem clássica, a programação do CLP está fundamentada na escrita do diagrama *Ladder* obtido na etapa anterior; esta tarefa é caracterizada por ser bastante extensa, visto que o operador deve transferir para o CLP, via software de programação, todo o programa baseado nas transições etapas e ações, bem como nos intertravamentos para segurança operacional. Geralmente, o diagrama *Ladder* obtido pela abordagem clássica é bem mais extenso do que o obtido com PA via PlanPAS. Neste estudo de caso, o diagrama *Ladder* obtido através da abordagem com PA possui 175 linhas, enquanto o diagrama obtido para abordagem clássica possui 271 linhas (vide Apêndice C para abordagem com PA e Apêndice D para abordagem clássica).

A última etapa analisada em ambas as abordagens foi a execução da solução. Em PA, esta execução é específica para cada problema, de modo que novos problemas exigem modificações nos diagramas de objetos UML. Some ainda o fato de que este tipo de

abordagem está sujeito à quedas e falhas de comunicação da rede. Por outro lado, a abordagem clássica possui maior robustez por atender casos genéricos, sendo ainda independente de rede externa. No entanto, existem alguns níveis de desempenho do sistema que são abordados apenas com PA, como é o caso dos custos de transporte, custos de perda, métricas de otimização e possibilidade de operar segundo previsão de demanda. Além disso, dificilmente um modelo cíclico, com uma lógica centralizada – como visto na abordagem clássica – conseguiria prever todos os cenários possíveis em um domínio como a Bancada analisada, ainda mais se fossem consideradas as possíveis variações de demanda dos clientes.

A Tab. 5.3 resume as etapas de implantação de cada abordagem, bem como suas características relevantes, e apresenta uma comparação indicando qual abordagem obteve melhor desempenho no estudo de caso.

Tabela 5.3. Comparativo entre os métodos de implantação utilizando PA e CLP clássica.

Etapas	Planejamento Automático (PA)	Abordagem Clássica CLP	Melhor Desempenho
Levantamento / Análise dos requisitos	Diagrama de casos de uso	Totalmente intuitivo	PA
Modelagem do domínio	Diagrama de classes e de estados	Modelagem em SFC	Clássica
Transcrição da solução em linguagem de programação.	Diagrama de objetos, “plano-solução” e “tabela de-para”	Conversão SFC - <i>Ladder</i>	Clássica
Programação do CLP (driver e intertravamentos)	Programação rápida e intuitiva.	Programação extensa, baseada nas transições, etapas e ações e nos intertravamentos.	PA
Execução	Específica a cada plano; sujeita a falhas de comunicação da rede.	Atende casos genéricos com maior robustez; não depende de rede externa.	Depende da aplicação, PA melhor para específicos; clássica melhor para genéricos.

Além de analisar as etapas de aplicação das soluções via itSIMPLE, planejador e PlanPAS, ou via abordagem clássica, uma outra análise realizada foi com relação ao desempenho dessas abordagens quanto a métricas qualitativas e quantitativas. Para isto, avaliou-se o desempenho, tanto de PA como do método clássico, com relação ao tempo de atendimento aos clientes da bancada, à facilidade de implementação, à capacidade de comparação, à previsão de demanda, ao número de ações, ao custo total, e à manutenção do sistema. A Tab. 5.4 apresenta as métricas analisadas, suas descrições e a abordagem que obteve melhor desempenho.

Tabela 5.4. Desempenho do sistema de acordo com as métricas analisadas.

Métrica	Análise	Sistema de melhor desempenho
Tempo total de atendimento aos clientes.	A operação manual no processo com a abordagem clássica atrasa o tempo de atendimento aos clientes.	PA
Facilidade de implementação	Para profissionais conhecedores de CLPs e IEC 61131-3, a implementação de PA é mais complexa.	Clássica
Capacidade de comparação	A interface de PA possibilita a comparação entre estados planejados e obtidos, o que não ocorre em CLP e supervisão.	PA
Previsão de demanda	Em CLP, o sistema age em função da variação do nível dos distribuidores e não em relação às demandas dos clientes, como ocorre em PA.	PA
Nº de ações	Como o sistema não atingiu o estado objetivo na abordagem clássica, a análise desta métrica não é aplicável.	Não se aplica
Custo total	Apesar de apenas PA lidar com métricas de otimização, como o sistema não atingiu o estado objetivo na abordagem clássica, a análise desta métrica não é aplicável.	Não se aplica
Manutenção do sistema	Qualquer modificação necessária é melhor realizada em PA, visto que o domínio está modelado graficamente e qualquer alteração de parâmetros ou inserção de novas ontologias se torna simples. Em CLP, o programador deve reestudar o diagrama <i>Ladder</i> e descobrir onde será implementada a modificação, sendo que talvez até mesmo o SFC deva ser modificado.	PA

A primeira métrica analisada diz respeito ao tempo total para atendimento dos clientes. Em PA foram necessários 13 minutos e 46 segundos para que o sistema executasse 19 ações e atendesse as demandas dos dois clientes envolvidos no processo. Trabalhando com a abordagem clássica, onde toda a lógica é centralizada e processada no próprio CLP, para atender às demandas dos clientes, o sistema levou 13 minutos e 57 segundos em um

total de 15 ações. Percebe-se que, mesmo operando um número menor de ações, a abordagem clássica despendeu maior tempo para finalizar o processo. Isso pode ser explicado pela particularidade levantada na seção 4.6, de que modelo em SFC não prevê as demandas dos clientes, sendo necessário o acionamento manual das eletrobombas B3 e B4 para simular a entrega de produtos aos mesmos. Na prática, o sistema de atendimento ao cliente fica separado da lógica de resuprimento. Tal particularidade surge como um complicador do sistema operando com abordagem clássica, pois atrasa o processo de entrega de produto aos clientes.

Com respeito à facilidade de implementação das duas soluções, para profissionais conhecedores de CLPs e norma IEC 61131-3, a implementação da solução de planejamento automático via PlanPAS é mais complexa do que a solução clássica, visto que as representações em UML são desconhecidas por este tipo de profissional.

A métrica “capacidade de comparação” é bem mais amparada por PA, visto que esta abordagem possibilita a comparação entre estados esperados e obtidos, permitindo uma monitoração constante do ambiente. Já a abordagem clássica não possibilita este tipo de comparação, visto que a execução das ações não se dá de acordo com uma sequência de ações pré-definidas, mas sim conforme as transições etapas e ações correspondentes ao modelo em SFC. Portanto, sob o ponto de vista da capacidade de comparação, a abordagem com PA se destaca.

Falando sobre previsão de demanda, na abordagem clássica (CLP) não é possível prever o momento exato das operações de entrega de produto para os clientes, neste caso o sistema age apenas em função da variação do nível dos distribuidores. Considerando a provável situação onde, em um determinado momento, um cliente demanda certa quantidade de produto, cujo valor seja superior ao nível corrente do distribuidor, o sistema só agirá no sentido de reabastecer o distribuidor, fornecendo produto necessário para o atendimento do cliente, quando o nível crítico deste distribuidor for atingido. Já com PA a sequência de ações dada pelo “plano-solução” já prevê os pedidos dos clientes e antecipa o abastecimento dos reservatórios distribuidores, otimizando o processo de entrega aos clientes. No caso de atualização dos pedidos dos clientes, um novo planejamento deve ser solicitado.

Para analisar o número de ações executadas e o custo total do processo, vamos recorrer aos dados apresentados na Tab. 5.5. Esta tabela levanta uma comparação entre os estados alcançados nos ensaios e o estado final desejado, definido no *snapshot* objetivo da Fig. 4.8. No estado final esperado, os sensores de nível dos reservatórios distribuidores (S2 e S3) devem indicar 50% da capacidade, sendo indiferente a quantidade de produto no reservatório do veículo (S1). Além disso, o veículo deve estar estacionado sob o reservatório

fornecedor, ou seja, sensor FC1 em nível “alto” e sensores FC2 e FC3 em nível “baixo”. Analisando os dados da Tab. 5.5, percebe-se que somente o cenário obtido através da abordagem com PA condiz com o *snapshot* objetivo apresentado na Fig. 4.8; com abordagem clássica o sistema não conseguiu finalizar o processo com o nível dos reservatórios distribuidores em 50%. Tal resultado ratifica a principal diferença entre as duas abordagens: planejamento automático é indicado para solução de problemas específicos, daí ser possível atingir um estado final pré-definido; a abordagem clássica opera genericamente em função do nível dos distribuidores, não ficando preso a atingir um estado final específico. Assim, dado que o sistema não atingiu o estado objetivo com a abordagem clássica, a análise das métricas “nº de ações” e “custo total” não é aplicável.

Quanto à manutenção do sistema, em PA o domínio está modelado graficamente e qualquer alteração de parâmetros ou inserção de novas ontologias se torna simples. Já na abordagem clássica, o programador deve reestudar o diagrama *Ladder* e descobrir onde será implementada a modificação, sendo que talvez até mesmo o SFC deva ser modificado.

Tabela 5.5. Comparação do estado esperado para os métodos testados.

	Estado	S1	S2	S3	FC1	FC2	FC3
Modelo	Estado objetivo	-	5	5	1	0	0
Abordagem Clássica	Estado 15	2	3	8	1	0	0
Planejamento Automático	Estado 19	2	5	5	1	0	0

O objetivo das análises apresentadas neste capítulo não se trata de definir qual a melhor abordagem de ataque para solução do problema, mas sim apontar as características de cada abordagem e seu respectivo comportamento em uma determinada etapa de implantação ou em uma determinada métrica de operação, para o estudo de caso em questão. As características da aplicação é que definirá quais as métricas devem ter maior peso de decisão na hora de escolha do método mais adequado.

5.3. Comparativo com os Trabalhos Relacionados

Relacionando o trabalho apresentado com aqueles apresentados na seção 2.6, é possível levantar um comparativo entre as principais características de cada estudo.

O sistema PlanPAS, proposto neste trabalho, utiliza uma técnica de inteligência artificial conhecida com Planejamento Automático para solucionar um problema logístico de distribuição de produtos em uma cadeia de suprimentos controlada por CLP. A validação

deste sistema é realizada através da transcrição da solução de PA diretamente para o CLP, via sistema PlanPAS, e comparação com o método de abordagem clássico (programação do CLP).

Em 2012, Peixoto propôs a técnica de MAS/SOA para solucionar um problema da manufatura com um controlador do tipo CLP. Seu estudo foi validado através de uma simulação do ambiente com MAS/SOA e comparação com o método de abordagem clássico.

Em 2010, Dumitru et al. propuseram a utilização de lógica *fuzzy* para controlar um servomecanismo através da aplicação direta da lógica *fuzzy* em um CLP.

Ainda em 2010, Vrba e Marik utilizaram MAS para solucionar um sistema de transporte com um controlador do tipo CLP, validando seu estudo em uma bancada da *Rockwell Automation*.

Por fim, em 2001, Barata et al. propuseram a utilização de MAS para solução de um problema da manufatura através do acesso ao controlador específico de um robô através de um *proxy* e servidor, por meio da internet.

A Tabela 5.6 apresenta, resumidamente, o comparativo entre os trabalhos.

Tabela 5.6. Comparativo entre o PlanPAS e os trabalhos relacionados.

Autor	Técnica de IA	Sistema	Controlador	Validação
Fonseca (2013)	PA	Logística	CLP	Aplicação direta de uma solução de PA em um CLP, via sistema PlanPAS
Peixoto (2012)	MAS/SOA	Manufatura	CLP	Simulação do ambiente com MAS/SOA e <u>comparação</u> com a abordagem clássica (CLP)
Dumitru et al. (2010)	Lógica <i>fuzzy</i>	Servomecanismo controlado por CLP	CLP	Aplicação direta de lógica <i>fuzzy</i> em um CLP
Vrba e Marik (2010)	MAS	Transporte	CLP	Bancada da <i>Rockwell Automation</i>
Barata et al. (2001)	MAS	Manufatura	Controlador específico do Robô	Conexão com o controlador do robô através de um <i>proxy</i> e servidor, por meio da internet

Percebe-se, por essa comparação, que o esforço em se aplicar técnicas de IA na solução de problemas da Engenharia está em ascensão. Dentre os trabalhos pesquisados, apenas um não aplica a solução em um controlador do tipo CLP. Além disso, o trabalho

desenvolvido é o único que se propõe a utilizar a técnica de Planejamento Automático para a solução de um problema prático, enquanto a técnica mais utilizada são os sistemas multiagentes.

CAPÍTULO VI

CONCLUSÃO

Este projeto diz respeito ao desenvolvimento de uma ferramenta de integração denominada PlanPAS. Esta ferramenta possibilita a aplicação direta de soluções de planejamento automático em sistemas reais controlados por CLP através da comunicação via arquitetura cliente/servidor TCP/IP e cliente/servidor OPC.

A principal ideia deste trabalho é caminhar no sentido de introduzir “inteligência” aos processos de automação. Para a validação do projeto foram realizados ensaios com uma bancada didática de distribuição de produto. O domínio da bancada didática foi modelado em UML e a ferramenta de conversão proporcionada pelo sistema itSIMPLE foi responsável pela tradução da descrição do sistema em UML para PDDL, o que possibilitou a avaliação dos problemas de planejamento pelos planejadores automáticos. Com o aplicativo proposto, toda a lógica de operações do sistema passou a ser fornecida por um “agente inteligente”, no caso o planejador automático, enquanto o CLP passou a operar apenas como um *driver* do ambiente, carregando consigo, ainda, os intertravamentos necessários para garantir a segurança operacional do sistema.

O desenvolvimento do projeto mostrou que a integração entre planejamento automático e sistemas reais baseados em CLPs está diretamente ligada com o relacionamento correto entre as ações planejadas e os atuadores do sistema real. O conhecimento prévio destas ações, ou seja, o conhecimento do domínio explorado, é de fundamental importância, tanto no processo de modelagem do domínio de planejamento quanto na definição da tabela de relacionamentos, durante a configuração da interface do PlanPAS. A habilitação das saídas digitais do CLP via PlanPAS é realizada com o auxílio de um servidor OPC, o servidor possui acesso de leitura/escrita a um conjunto de memórias auxiliares, cuja função é atuar, via diagrama *Ladder*, como disparadores das saídas digitais do CLP. Nesta configuração, o controlador passa então a operar somente como um “driver” do sistema, onde toda a lógica de execução das ações é dada por um “plano-solução”. Para

evitar possíveis problemas, tais como queda ou congestionamento de rede, o diagrama *Ladder* de programação do CLP inclui também todos os intertravamentos necessários para garantir a segurança operacional do sistema.

O desenvolvimento da ponte de comunicação com o servidor OPC, denominada *PLC Interface*, levou em consideração a utilização de um protocolo de comunicação aberto, neste caso XML através de TCP/IP, para facilitar a inserção de novas implementações. A partir do momento em que se utiliza um protocolo de comunicação aberto, a utilização de linguagens de programação independentes de sistema operacional, tal como Java, facilita a inserção de novos componentes ao sistema como um todo. Foi o que ocorreu no desenvolvimento do Parser e da *interface* do PlanPAS, ambos implementados em Java.

Após a análise dos resultados e a comparação dos métodos de abordagem do problema, percebe-se que, com relação à implantação dos sistemas, a abordagem clássica se destaca na modelagem do domínio e transcrição do modelo em linguagem de programação, enquanto que a abordagem com PA destaca-se nas etapas de levantamento e análise dos requisitos e programação do CLP. A etapa de execução da solução se mostrou dependente do tipo de aplicação, destacando-se PA para situações específicas e CLP clássica para situações genéricas.

Com relação ao desempenho do sistema, a análise dos resultados obtidos nos testes realizados mostrou que a abordagem com PA se destacou no tempo total de atendimento aos clientes, capacidade de comparação, previsão de demanda e manutenção do sistema. A abordagem clássica se mostrou incapaz de lidar com previsão de demanda e antecipação das ações em prol do atendimento dos clientes, levando vantagem apenas no quesito facilidade de implementação da solução. Os quesitos número de ações e custo total não foram passíveis de comparação devido à diferença nos estados alcançados com cada abordagem.

Como pôde ser visto, o trabalho não aborda a tarefa de replanejamento. Esta funcionalidade encontra-se em fase de desenvolvimento e sua complexidade é considerada alta, visto que demanda acesso ao código fonte do itSIMPLE para uma correta implementação. Como o itSIMPLE é um sistema de código aberto e de conteúdo bastante extenso, compreender seu código e ousar escrever alguma modificação exige tempo e sobretudo disciplina.

O trabalho apresenta uma análise da integração entre Planejamento Automático e sistemas reais de automação. Foi desenvolvido um software aplicativo capaz de atuar uma solução de planejamento automático diretamente em um sistema controlado por CLP. O que fica registrado é o processo no sentido de integrar sistemas “inteligentes” com sistemas reais de automação de uma maneira clara e objetiva, e que pode servir como referência

para futuros trabalhos sobre o mesmo assunto. Além disso, destaca-se o procedimento de desenvolvimento do software, que pode ser estendido para integração de planejamento automático com outros tipos de controladores, bem como para a integração entre uma solução oriunda de qualquer sistema “inteligente” com dispositivos de campo, com ou sem uso de *drivers* de acionamento.

O principal tópico para trabalho futuro é, justamente, a implementação do replanejamento automático. A ideia é que o sistema seja capaz de detectar automaticamente o estado corrente do sistema e enviar esta informação para o itSIMPLE como sendo um novo cenário inicial, para daí, então, solicitar a tarefa de replanejamento. Além disso, almeja-se abordar a integração de sistemas multiagentes com planejadores automáticos para a solução de problemas práticos de automação.

Do ponto de vista da interface do PlanPAS, são necessárias melhorias no sentido de generalizar a sua aplicação para mais de um tipo de CLP, bem como para outros tipos de controladores, tais como; microcontroladores PIC, *Arduino*, *Raspberry PI*. O objetivo é que o usuário defina o tipo de controlador utilizado através da própria interface do PlanPAS, bem como as especificações dos endereços de E/S do controlador, tipo de acesso (leitura/escrita ou somente leitura) às variáveis ambientes via servidor OPC, definição do endereço IP e porta para comunicação via rede ethernet, etc.

Com relação à autonomia do sistema, pretende-se promover a comunicação direta do PlanPAS com o itSIMPLE e utilizar apenas o itSIMPLE como interface do usuário. Por exemplo, tanto a tabela de relacionamentos quanto a tabela de monitoração do PlanPAS podem ser geradas diretamente pelas informações do domínio inseridas no itSIMPLE. A meta é aprimorar a comunicação do sistema como um todo, permitindo com que um usuário do itSIMPLE solicite que o sistema configure o PlanPAS para o domínio em questão e passe a acompanhar todo o processo via itSIMPLE.

Do ponto de vista operacional, uma maior autonomia do sistema pode ser garantida através de uma aplicação híbrida: o sistema operaria preferencialmente via PlanPAS porém, a qualquer iminência de queda ou congestionamento da rede, o CLP alteraria seu programa de operação, passando a operar com o método clássico.

No entanto, esta é uma implementação apenas operacional e em nada modificaria a estrutura do PlanPAS.

Como se vê, este foi apenas o primeiro passo no sentido de integrar soluções de planejamento automático e sistemas reais de automação baseados em CLPs. O que fica é o aprendizado para os próximos trabalhos, os quais poderão utilizar os conceitos apresentados aqui para ampliar os desenvolvimentos deste sistema para as suas necessidades. Quando um sistema mais autônomo for alcançado, diversas situações e

problemas específicos poderão ser analisados, diferentes estudos de casos para os mais diversos sistemas de automação em bancadas didáticas e, caso seja possível, em ambiente industrial deverão ser avaliados no sentido de levantar os critérios para determinar onde um controlador é mais indicado do que outro, seja ele um CLP ou um microcontrolador qualquer. Com isso, o papel dos controladores e das lógicas tradicionais de programação em sistemas de automação poderá ser reavaliado do ponto de vista da integração com sistemas “inteligentes”.

Enfim, o propósito do trabalho não é o de engrandecer ou menosprezar um ou outro método de abordagem, até porque é muito pouco provável que os profissionais do meio fabril abram mão de um processo clássico, consagrado pelas soluções rápidas e eficientes, por uma método “germinal”; ou mesmo que os pesquisadores da área de integração de sistemas “inteligentes” deixem de avançar seus estudos por barreiras culturais. A intenção é, na verdade, indicar que existem formas alternativas para atacar problemas de automação. Os métodos clássicos são genéricos e não preveem diversos quesitos importantes para a solução de problemas específicos. No estudo de caso avaliado, a abordagem com PA se mostrou uma alternativa interessante no ataque para a solução de problemas de natureza específica. Cabe aos engenheiros e profissionais do ramo de automação avaliar qual o melhor cenário para aplicação de cada abordagem, ou ainda a integração de ambas.

REFERÊNCIAS BIBLIOGRÁFICAS

BARATA, J.; CAMARINHA-MATOS, L.; LEITÃO, P.; BOISSIER, R.; RADDADI, M. Integrated and Distributed Manufacturing: A Multi-Agent Perspective. In: WORKSHOP ON EUROPEAN SCIENTIFIC AND INDUSTRIAL COLLABORATION, 3, 2001, Enschede. **Proceedings**....p.145-156. Disponível em: <<https://bibliotecadigital.ipb.pt/handle/10198/1535>>. Acesso em: 28 fev. 2013.

BLUM, A. L.; FURST, M. L. Fast Planning Through Planning Graph Analysis. **Artificial Intelligence**. Essex, v. 90, n.1, p. 281-300, fev. 1997. Disponível em: <<http://dl.acm.org/citation.cfm?id=249386>>. Acesso em: 28 fev. 2013.

BONET, B.; GEFNER, H. Planning as Heuristic Search: New Results. In: EUROPEAN CONFERENCE ON PLANNING, 5, 1999, Durham. **Proceedings**...p360-372. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.16.5803&type=cc>>. Acesso em: 21 mar. 2013.

BONET, B.; GEFNER, H. HSP: Heuristic Search Planner. **AI Magazine**. The AIPS-98 Planning Competition, California, v. 21, n.2, p. 13-34, jul. 2000. Disponível em: <<http://www.aaai.org/ojs/index.php/aimagazine/article/view/1505/1404>>. Acesso em: 25 abr. 2012.

BORGES, F. **Redes de Comunicação Industrial**: Documento técnico nº2. Centro de Formação da Schneider Electric Portugal, Lisboa, 2007. 44p. Documento Técnico. Disponível em: <http://www.schneiderelectric.pt/documents/product-services/training/doctecnico_redes.pdf>. Acesso em: 19 mar. 2013.

CÂNDIDO, R. V. B. **PADRÃO OPC**: Uma Alternativa de Substituição dos *Drivers* Proprietários para Acessar Dados de PLCs. 2004. 51 f. Monografia de Conclusão de Graduação –Universidade FUMEC, Belo Horizonte. Disponível em: <<http://www.aptasistemas.com.br/4.pdf>>. Acesso em: 19 mar. 2013.

CASSIOLATO, C. **Redes Industriais**. 2011. 22p. Artigo Técnico. Disponível em: <<http://www.smar.com/brasil/artigostecnicos/artigo.asp?id=48>>. Acesso em: 13 mai. 2013.

DAHAL, K.; GALLOWAY, S.; BURT, G.; MCDONALD, J.; HOPKINS, I. A Port System Simulation Facility with an Optimization Capability. **International Journal of Computational Intelligence and Applications**. London, v. 3, n.4, p. 395-410, dez. 2003. Disponível em: <<http://www.worldscientific.com/doi/pdf/10.1142/S1469026803001099>>. Acesso em: 25 abr. 2013.

DEITEL, P. J.; DEITEL, H. M. **Java How to Program**. 6.ed. Upper Saddle River: Pearson Prentice Hall, 2004. 1111 p.

DUMITRU, I.; ARGHIRA, N.; FAGARASAN, I.; ILIESCU, S. A Fuzzy PLC Control System for a Servomechanism. In: IFAC WORKSHOP ON INTELLIGENT CONTROL SYSTEMS, 1, 2010, Bucharest. **Proceedings...**p. 69-74. Disponível em: <<http://www.ifac-persononline.net/Detailed/43598.html>>. Acesso em: 25 abr. 2013.

EDELKAMP, S.; JABBAR, S. MIPS-XXL, Featuring External Shortest Path Search for Sequential Optimal Plans and External Branch-And-Bound for Optimal Net Benefit. In: INTERNATIONAL PLANNING COMPETITION BOOKLET, 6, 2008, Sydney. **Proceedings...**4p. Disponível em: <http://ipc.informatik.uni-freiburg.de/Planners?action=AttachFile&do=view&target=mips_xxl.pdf>. Acesso em: 12 abr. 2012.

ERNST, G.; NEWELL, A. **GPS: A Case Study in Generality and Problem Solving**. New York: Academic Press, 1969. 297p.

FIKES, R. E.; NILSSON N. J. STRIPS: A new approach to the application of theorem proving to problem solving. **Artificial Intelligence**. Essex, v. 2, n.3, p. 189-208, set. 1971. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0004370271900105>>. Acesso em: 03 nov. 2011.

FOGEL, L. J.; OWENS, A. J.; WALSH, M. J. **Artificial Intelligence Through Simulated Evolution**. New York: Wiley, 1966. 170p.

FONSECA, M. O. Comunicação OPC: Uma abordagem prática. In: SEMINÁRIO DE AUTOMAÇÃO DE PROCESSOS, 6, 2002, Vitória, **Anais...**12p. Disponível em: <<http://www.cpdee.ufmg.br/~seixas/PaginaSDA/Download/DownloadFiles/OPCMarcosFonseca.PDF>>. Acesso em: 12 fev. 2013.

FONSECA, J. P. S. **Desenvolvimento e Montagem de uma Bancada Didática de Planejamento Automático**. 2011. 165 f. Monografia de Conclusão de Graduação – Universidade Federal de Uberlândia, Uberlândia.

FONSECA, J. P. S.; CARDOZO, R. N.; GUIMARÃES, W. H. P.; RIBEIRO, K. S.; SOUSA, A. R.; TAVARES, J. J. P. Z. S.; CARVALHO, J. C. M. Automated Planning and Real Systems Based on PLC: A Practical Application in a Didactic Bench of Manufacturing Automation. In INTERNATIONAL CONFERENCE ON AUTOMATED PLANNING AND SCHEDULING, 22, 2012, Atibaia. **Proceedings of the Workshop on Combining Task and Motion Planning for Real-World Applications** p.37-44. Disponível em: <http://icaps12.poli.usp.br/icaps12/sites/default/files/tampra/TAMPRA_2012_Proceedings.pdf>. Acesso em: 06 dez. 2012.

FONSECA, J. P. S.; SOUSA, A. R.; FERREIRA, M. V. M.; TAVARES, J. J. P. Z. S.; CARVALHO, J. C. M. PlanPAS: Plans Parser for Automated Systems. In IFAC WORKSHOP ON INTELLIGENT MANUFACTURING SYSTEMS, 11, 2013, São Paulo. **Proceedings...** p75-80. Disponível em: <<http://www.ifac-papersonline.net/Detailed/59211.html>>. Acesso em: 13 jun. 2013.

FONSECA, J. P. S.; TAVARES, J. J. P. Z. S. Didatic Test Bench for Automated Planning. **ABCM Symposium Series in Mechatronics**, Section V - Intelligent and Distributed Manufacturing Systems, Rio de Janeiro, v. 5, n.5, p. 849 – 858, out. 2012. Disponível em: <http://www.abcm.org.br/symposiumseries/SSM_Vol5/Section_V_Intelligent_and_Distributed_Manufacturing_Systems/12691.pdf>. Acesso em: 06 dez. 2012.

GEREVINI, A.; SAETTI, A.; SERINA, I.; TONINELLI, P. LPG-TD, A Fully Automated Planner for PDDL2.2 Domains. In: INTERNATIONAL PLANNING COMPETITION BOOKLET, 4, 2004, British Columbia. **Proceedings...**2p Disponível em: <<http://130.203.133.150/viewdoc/summary?doi=10.1.1.164.6078>>. Acesso em: 13 jan. 2012.

GHALLAB, M.; NAU, D.; TRAVERSO, P. **Automated Planning**, theory and practice. San Francisco: Morgan Kaufmann Publishers, 2004. 635p.

GOUGH, J. **XPDDL 0.1b**, A XML version of PDDL. Strathclyde, 2004. Artigo Técnico. Disponível em: <<http://www.cis.strath.ac.uk/~jg/XPDDL/2004>>. Acesso em: 01 dez. 2011.

GREEN, C. Application of Theorem Proving to Problem Solving. In INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1, 1969, Menlo Park. **Proceedings...**p.219-239. Disponível em: <<http://dl.acm.org/citation.cfm?id=1624585>>. Acesso em: 13 jan. 2012.

HELMERT, M. A Planning Heuristic Based on Causal Graph Analysis. INTERNATIONAL CONFERENCE ON AUTOMATED PLANNING AND SCHEDULING, 14, 2004, Whistler. **Proceedings...**p.161–170. Disponível em <<http://www.aaai.org/Papers/ICAPS/2004/ICAPS04-021.pdf>>. Acesso em: 13 jan. 2012.

HOFFMANN, J. The Metric-FF Planning System: Translating “Ignoring Delete Lists” to Numeric State Variables. **Journal of Artificial Intelligence Research**. El Segundo, v. 20, n.1, p. 291–341, dez. 2003. Disponível em: <<http://www.jair.org/papers/paper1144.html>>. Acesso em: 13 jan. 2012.

HOFFMANN, J., NEBEL, B. The FF Planning System: fast plan generation through heuristic search. **Journal of Artificial Intelligence Research**. El Segundo, v. 14, n.1, p. 253-302, jan. 2001. Disponível em: <<http://dl.acm.org/citation.cfm?id=1622404>>. Acesso em: 13 jan. 2012.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**, an introductory analysis with applications to biology, control, and artificial intelligence. Ann Arbor: University of Michigan Press, 1975. 183p.

HONG, X.; JIANHUA, W. 2006. Using Standard Components in Automation Industry: A study on OPC Specification. **Journal of Elsevier Computer Standards & Interfaces**. North-Holland, v. 28, n.4, p.386–395, abr. 2006. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0920548905000814>>. Acesso em: 03 fev. 2012.

HSU, C. W.; WAH, B. W. The SGPlan Planning System in IPC-6. In: INTERNATIONAL PLANNING COMPETITION BOOKLET, 6, 2008, Sydney. **Proceedings...**3p. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.151.4521>>. Acesso em: 12 abr. 2012.

KAUTZ, H.; SELMAN, B.; HOFFMANN, J. SATPLAN: Planning as Satisfiability. In: INTERNATIONAL PLANNING COMPETITION BOOKLET, 5, 2006, Cumbria. **Proceedings...**2p. Disponível em: <<http://www.cs.rochester.edu/u/kautz/papers/kautz-satplan06.pdf>>. Acesso em: 12 abr. 2012.

LEITÃO, G. B.P. **Arquitetura e Implementação de um cliente OPC para aquisição de dados na indústria do petróleo**. 2006. 81f. Monografia de Conclusão de Graduação – Universidade Federal do Rio Grande do Norte, Natal.

LI, J.; WENKAI, L.; KARIMI, I. A.; SRINIVASAN, R. Robust and Efficient Algorithm for Optimizing Crude Oil Operations. In: AMERICAN INSTITUTE OF CHEMICAL ENGINEERS 2005 ANNUAL MEETING, 1, 2006, Cincinnati. **Proceedings...**15p. Disponível em: <<http://www3.aiche.org/Proceedings/ExtendedAbstract.aspx?PaperID=28112>>. Acesso em: 18 mar. 2013.

MATRIKON OPC. **MatrikonOPC Server for OMRON PLCs**. 2011. Software de produto. Disponível em: <<https://www.matrikonopc.com/downloads/types/drivers/index.aspx>>. Acesso em: 31/10/2011.

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**. Amsterdam, v. 5, n.1, p. 115-133, 1943. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0092824005800060>>. Acesso em: 18 jan. 2013.

MCDERMOTT, D. PDDL, The Planning Domain Definition Language. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE PLANNING SYSTEMS, 4, 1998, Pittsburgh. **Proceedings...**27p. Disponível em: <<http://www.inf.ed.ac.uk/teaching/courses/propm/papers/pddl.pdf>>. Acesso em: 11 fev. 2012.

MIYAGI, P. E. **Controle Programável: Fundamentos do Controle de Sistemas a Eventos Discretos**. Edgard Blucher, São Paulo, 1996. 208p.

MORAES, C. C.; CASTRUCCI, P. **Engenharia de Automação Industrial**. Rio de Janeiro: LTC, 2001. 295p.

MURATA, T. (1989). Petri Nets: Properties, Analysis and Applications. **Proceedings of the IEEE**. Piscataway Township, v. 12, n.4, p. 541-580. abr. 1989. Disponível em: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=24143&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D24143>. Acesso em: 30 jul. 2012.

NASSEHI, A.; ALLEN, R. D.; NEWMAN, S. T. A New Software Platform for STEP-NC Manufacturing Application Development. In: IFAC Symposium on Information Control Problems in Manufacturing, 12, 2006, Saint Etienne. **Proceedings...**p. 831-836. Disponível em: <<http://opus.bath.ac.uk/25489/>>. Acesso em: 09 mar. 2013.

NEWELL, A.; SIMON, H. **Human Problem Solving**. Englewood Cliffs: Prentice-Hall, 1972. 920p.

NEWELL, A. (1980), Physical Symbol Systems. **Cognitive Science**. Seoul, v. 4, n.2, p. 135-183, abr 1980. Disponível em: <http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog0402_2/abstract>. Acesso em: 02 fev. 2013.

OGATA, K. **Engenharia de Controle Moderno**. 4.ed. São Paulo: Prentice Hall, 2003. 788p.

OLIVEIRA, J. C. P. **Controlador Programável**. São Paulo: Makron Books, 1993. 200p.

OMG: OBJECT MANAGEMENT GROUP. **Unified Modeling Language specification**. Sítio. Disponível em: <<http://www.omg.org/uml>>. Acesso em: 14 jul. 2010.

OMRON CORPORATION (Japan) (Org.). Programmable Controllers: Operation Manual. Tokyo, 2001. CD-ROM.

OPC FOUNDATION. **What is OPC?**. 2013. 2p. Sítio. Disponível em: <http://www.opcfoundation.org/Default.aspx/01_about/01_what_is_opc?MID=AboutOPC>. Acesso em: 30 jul. 2013.

OPCTI. **OPC Unified Architecture**. 2013. Sítio. Disponível em: <<http://www.opcti.com/Next-Generation-of-OPC.aspx>>. Acesso em: 30 jul. 2013.

PAVEI, J. **Sistemas de Controle via Redes Industriais**: Desenvolvimento de um software para controle avançado via OPC. 2009. 54f. Monografia de Conclusão de Graduação – Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.wbezerra.com.br/prh34/site/trabahos_finais/graduacao/Jonatas%20Pavei_PRH34_UFSC_DAS_G.pdf>. Acesso em: 16 mar. 2013.

PEIXOTO, J. A. **Desenvolvimento de Sistemas de Automação da Manufatura usando Arquiteturas Orientadas a Serviço e Sistemas Multi-Agentes**. 2012. 129f. Dissertação de Mestrado – Universidade Federal do Rio Grande do Sul, Porto Alegre.

RICARTE, I. L. M. **Programação Orientada a Objetos**: Uma Abordagem com Java. Universidade Estadual de Campinas, Campinas, 2001. 117p. Material didático. Disponível em <http://www.programandojava.xpg.com.br/Apostila%20Java/2783_Java_boa.pdf>. Acesso em: 15 jul. 2012.

SAHIN, C.; BOLAT, E. D. Development of Remote Control and Monitoring of Web-Based Distributed OPC System. **Journal of Elsevier Computer Standards & Interfaces**. North-Holland, v. 31, n.5, p. 984–993, set. 2009. Disponível em <<http://www.sciencedirect.com/science/article/pii/S0920548908001487>>. Acesso em: 03 fev. 2012.

SETTE, F.M.; VAQUERO, T.S.; PARK, S.W.; SILVA, J.R.. Are Automated Planers up to Solve Real Problems?. In: WORLD CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, 17, 2008, Coex. **Proceedings...**p. 15817-15824. Disponível em <<http://www.ifac-papersonline.net/Detailed/38355.html>>. Acesso em: 07 set. 2011.

SEIXAS FILHO, C. A Automação nos Anos 2000: Uma análise das novas fronteiras da automação. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL, 4, 2000, São Paulo. **Anais...**7p. Disponível em: <<http://www.cpdee.ufmg.br/~seixas/Paginall/Download/DownloadFiles/Conai2000Automacao.PDF>>. Acesso em: 23 abr. 2011.

SILVEIRA, P. R.; SANTOS, W. E. **Automação e Controle Discreto**. 5.ed. São Paulo: Érica, 2003. 229p.

STEVENS, W. R. **TCP/IP Illustrated Volume 1: The Protocols**. Boston: Addison-Wesley, 1994, 600p.

TANENBAUM, A. S. **Redes de Computadores**. 4.ed. Rio de Janeiro: Campus, 2003. 945p.

TAVARES, J. J. P. Z. S. **Elaboração de Projetos de Automação e Controle: 3º Módulo Prático**. Universidade Federal de Uberlândia, Uberlândia, 2010. 43p. Material Didático.

TAVARES, J. J. P. Z. S.; FONSECA, J. P. S. Supply Chain Didactic Testing Bench with Automated Planning Tool. In: International Conference on Production Research, 21, 2011, Stuttgart. **Proceedings...**pp. 242.

TAVARES, J. J. P. Z. S.; FONSECA, J. P. S.; VAQUERO, T. S.; SILVA, J. R. Integração de Planejamento Automático e Sistemas Reais Baseados em CLP. In: Simpósio Brasileiro de Automação Inteligente, 10, 2011, São João Del Rey. **Proceedings...**p. 1226-1231.

TRSEK, H.; JASPERNEITE, J.; LESSMANN, G. A Concept for the System Integration of Wireless Sensor Network to Industrial Automation Systems using Profinet. In: IFAC INTERNATIONAL CONFERENCE ON FIELDBUSES AND NETWORKS IN INDUSTRIAL AND EMBEDDED SYSTEMS, 7, 2007, Toulouse. **Proceedings...**p. 183-190.

VAQUERO, T. S. **ITSIMPLE**: Ambiente integrado de análise de domínios de planejamento automático. 2007. 316 f. Dissertação de Mestrado – Universidade de São Paulo, São Paulo.

VAQUERO, T. S.; ROMERO, V.; TONIDANDEL, F.; SILVA, J. R. itSIMPLE2.0: An Integrated Tool for Designing Planning Domains. INTERNATIONAL CONFERENCE ON AUTOMATED PLANNING AND SCHEDULING, 17, 2007, Rhode Island. **Proceedings...**p.336-343. Disponível em: <<http://itsimple.googlecode.com/svn/wiki/resources/papers/2007/1.pdf>>. Acesso em: 02 jan. 2013.

VAQUERO, T. S.; SILVA, J. R.; FERREIRA, M.; TONIDANDEL, F.; BECK, J. C. From Requirements and Analysis to PDDL in itSIMPLE3.0. In: INTERNATIONAL COMPETITION IN KNOWLEDGE ENGINEERING FOR PLANNING AND SCHEDULING, 3, 2009,

THESSALONIKI. **Proceedings...**p. 54-61. Disponível em: <http://tidel.mie.utoronto.ca/pubs/ICKEPS09tvaquero.pdf>. Acesso em: 02 jan. 2013.

VAQUERO, T. S.; SILVA, J. R.; TONIDANDEL, F.; BECK, J. C. itSIMPLE: towards an integrated design system for real planning applications. **The Knowledge Engineering Review**. Cambridge, v. 28, n.2, p. 215-230, jun. 2013. Disponível em: <http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=8910412>. Acesso em: 30 jul. 2013

VRBA, P.; MARIK, V. Capabilities of Dynamic Reconfiguration of Multiagent-Based Industrial Control Systems. **Systems, Man, and Cybernetics, Part A: Systems and Humans**,. **IEEE Transactions on**. Piscataway Township, v. 40, n.2, p. 213-223, mar. 2010. Disponível em <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5353631&isnumber=5415834>. Acesso em: 15 mai. 2013.

WAH, B. W.; CHEN Y. Subgoal Partitioning and Global Search for Solving Temporal Planning Problems in Mixed Space. **International Journal of Artificial Intelligence Tools**. Hackensack, v. 13, n.4, p. 767-790, jun. 2004. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.6656&rep=rep1&type=pdf>. Acesso em: 01 mai. 2013.

WEBER, M.; KINDLER, E. The Petri Net Markup Language. **Petri Net Technology for Communication-Based Systems: Advances in Petri Nets**, Heidelberg: Springer, v. 2472, p. 124-144. jan. 2003. Disponível em: http://www.informatik.huberlin.de/top/pnml/download/about/PNML_LNCS.pdf. Acesso em: 06 dez. 2012

WORLD WIDE WEB CONSORTIUM (W3C). **Extensible Markup Language (XML) 1.0 (Fifth Edition)**. 2008. 34p. W3C Recommendation. Disponível em: <http://www.w3.org/TR/REC-xml/>>. Acesso em: 23 ago. 2010.