

PLANEJAMENTO AUTOMÁTICO E SISTEMAS REAIS BASEADOS EM CLP: UMA APLICAÇÃO PRÁTICA EM UMA BANCADA DIDÁTICA DE AUTOMAÇÃO DA MANUFATURA

João Paulo da Silva Fonseca; José Jean Paul Zanlucchi de Souza Tavares; João Carlos Mendes Carvalho.

Abstract: Automated planning system has been developed for more than 40 years, however, their practical application in real problems is still restrict and, many times, a challenge. One of the reasons for such restriction relies on language ruptures. On one side, automated planning systems provides a plan with predefined sequences of actions represented in the Planning Domain Definition Language (PDDL); on the other side, real systems use equipment as PLC or Programming Logic Controller, which utilize languages as Ladder and centralizes sensor and actuation activities. This paper present a manufacturing automation methodology, based in automated planner results, applied at a didactic bench. First, the planning domain was modeled in Unified Definition Language (UML). This model was analyzed by automated planners about optimization specific metrics, in this case, costs minimization. The best plan was chosen for will be implemented in practical system. The sequence actions defined in PDDL was translated in a Sequential Flow Chart and thus, translated in a Ladder diagram, to implement in PLC. Finally, results are presented and compared with the classical methodology solution.

Keywords: *Automated Planning; Manufacturing Automation; Programmable Logic Control (PLC); Didactic Bench; itSIMPLE.*

Resumo: Há quase 40 anos iniciou-se o desenvolvimento de sistemas de planejamento automático, porém, sua aplicação prática ainda é restrita e, muitas vezes, um desafio. Isso se deve ao fato de que as linguagens utilizadas, tanto nas aplicações reais, como nas soluções de sistemas com planejamento automático, apresentam grandes discrepâncias. Enquanto o planejamento automático apresenta um plano com seqüências pré estabelecidas baseadas em ações definidas previamente como se os objetos fossem autônomos em uma linguagem denominada *Planning Domain Definition Language* (PDDL); os sistemas reais fazem uso de equipamentos do tipo CLP ou Controlador Lógico Programável com linguagem do tipo *Ladder* e centralizam as operações de sensoramento e acionamentos. Este trabalho apresenta uma metodologia de automação da manufatura, considerando os resultados de um planejador automático, aplicada em uma bancada didática. Inicialmente, o domínio de planejamento foi modelado em um ambiente de desenvolvimento e análise de domínios de planejamento, em uma linguagem de especificação formal denominada *Unified Definition Language* (UML). Este modelo foi analisado por planejadores automáticos em função de métricas específicas de otimização, neste caso, redução de custos. O melhor plano foi então escolhido para ser implementado no sistema prático. A sequencia de ações definidas em PDDL foi traduzida em um Gráfico Funcional de Comando Etapa-Transição e, em seguida, em um diagrama *Ladder* que foi implementado em um CLP. Finalmente, resultados são apresentados e comparados com a solução obtida através da metodologia clássica.

Palavras chave: *Planejamento Automático; Automação da Manufatura; Controlador Lógico Programável (CLP); Bancada Didática; itSIMPLE.*

Sistemas discretos baseados em CLP's são vastamente utilizados nos processos de automação da manufatura. A eficiência reconhecida internacionalmente aliada à sua robustez garante seu sucesso nas aplicações industriais. Por outro lado, há alguns anos diversos estudiosos vem apresentando técnicas de inteligência artificial, tais como redes neurais, algoritmos genéticos, sistemas especialistas, lógica fuzzy, planejamento automático etc. como propostas de melhoria dos sistemas reais existentes. Especificamente, o planejamento automático surgiu em 1971 com a apresentação do primeiro planejador automático STRIPS (Stanford Research Problem Solver) (FIKES; NILSSON). Segundo os autores o planejador automático trata-se de um sistema para resolução automática de problemas onde, a partir de um modelo que representa as ações e objetos do sistema real, um conjunto de entradas (estado inicial) e uma situação objetivo (estado final), o planejador roda uma árvore de busca de modo que se defina a melhor sequência de ações que leve o sistema da situação inicial à situação objetivo.

Contudo, o que se vê é um grande abismo entre estes desenvolvimentos, o que dificulta a integração de novas tecnologias em aplicações industriais como, por exemplo, na automação da manufatura. Em (TAVARES; FONSECA, 2011) é apresentado um estudo inicial para avaliar a distância entre planos teóricos e sistemas práticos a partir de um problema recorrente em uma cadeia de suprimentos, através de um exemplo simples.

Nos parágrafos seguintes é apresentada a fundamentação teórica deste trabalho. Inicialmente é apresentado o conceito de planejamento automático, em seguida é apresentado o sistema itSIMPLE (VAQUERO et al, 2009) e, finalmente, é feita uma breve descrição sobre o CLP e algumas de suas linguagens de programação.

$$s_0 \ a_1 \dots \ s_j \ a_j \ s_{k \dots} \ a_f \ s_f \quad (1)$$

A resolução de um problema de planejamento pode ser feita "*on the flow*", isto é, na medida que novos estados vão sendo gerados, um sistema inteligente opera no sentido de identificar novas ações; ou resolvendo inteiramente o sistema antes de aplicar o plano (usando planejadores automáticos). (TAVARES et al, 2011)

É possível pensar em um processo de fabricação na manufatura moderna como uma sucessão de ações independentes iniciadas por um acionamento e cuja finalização pode ser detectada por sensores. Estas ações podem ser realizadas por dispositivos, máquinas de comando numérico, AGV's, esteiras rolantes, robôs manipuladores, etc. Portanto estas ações são programas relativamente complexos, em diferentes linguagens, incluindo-se aí também as linguagens de programação de CLP's.

Assim, o seqüenciamento aqui referido pode ser visto como uma integração de diferentes atividades, dispositivos e máquinas, para a obtenção de um processo mais sofisticado. Mesmo que o processo de detecção e acionamento possa também ser feito por CLP's, nota-se que uma linguagem deste nível não é adequada a este tipo de integração. O próprio processo de projeto não é factível em uma linguagem de baixo nível como a linguagem *Ladder*, ainda que no final do processo, se deseje justamente ter uma forma de acionar automaticamente a entrada destas ações usando um CLP (e detectando a condição de término das ações com sensores). (TAVARES et al, 2011)

Torna-se assim bastante atraente a possibilidade de se aplicar novas ferramentas de design para sistemas de planejamento e escalonamento neste tipo de integração, com as vantagens em precisão, análise de dependências, adaptabilidade e inserção de um comportamento inteligente no sistema.

Até recentemente esta possibilidade era muito remota, dado que os problemas de planeamento automático eram tratados apenas como problemas modelo, sendo extraídos já diretamente em linguagens de especificação formal como a PDDL (FOX; LONG, 2003). Entretanto, a partir da virada do século, uma discussão passou a dominar a comunidade de planeamento, e que se refere justamente à possibilidade do uso destas técnicas da Inteligência Artificial a problemas reais entre estes os problemas de manufatura.

Problemas reais de grande interesse têm sido abordados com estas técnicas como na logística dos sistemas portuários (DAHALL et al, 2003), ou na logística de carga, descarga e tanqueamento de petróleo no porto de São Sebastião (SETTE et al, 2008), ou ainda no roteamento de óleo cru em oleodutos (LI et al, 2005).

É preciso portanto ter um processo de projeto que permita associar o plano gerado pelas técnicas de planeamento a cada um dos sub-processos que substanciam as ações. A Figura 1 a seguir mostra a proposta de processo de projeto apresentada no sistema itSIMPLE (VAQUERO et al, 2007).

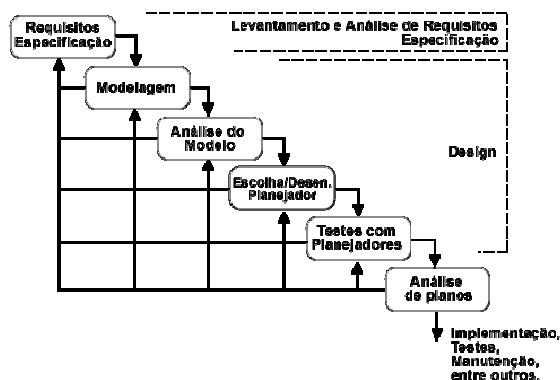


Figura 1: Processo de projeto adotado pelo ambiente itSIMPLE. (TAVARES et al, 2011)

Nota-se que nesta proposta um tratamento é feito passando pela eliciação e análise de requisitos, modelagem e análise do modelo, antes mesmo de escolher o planejador adequado para a geração do plano. Uma vez feito o plano, este é analisado para detectar incorreções que podem ser o resultado de omissões e contradições nos requisitos que só podem ser detectados após a elaboração do plano (Vaquero et. al. 2010).

O itSIMPLE foi projetado para permitir que usuários tenham um processo de design disciplinado para criar modelos de conhecimento de diferentes domínios de planeamento automático. O processo sugerido e implementado na ferramenta para projetar os modelos de domínios segue uma sequência de fases (cíclica) herdadas das Engenharias do Conhecimento e Software combinado a experiências adquiridas no design de aplicações de planeamento. (TAVARES et al, 2011)

O itSIMPLE proporciona uma abordagem diferente para a modelagem do domínio de planeamento. Sua principal característica é possibilitar que todo o processo de modelagem possa ser feito através de diagramas UML (*Unified Definition Language*) (OMG, 2003). Como o grupo de conhecedores da linguagem PDDL e seus formalismos é muito limitado, o itSIMPLE abriu as portas para que um grupo maior de pessoas seja capaz de modelar os domínios de planeamento, a partir de uma linguagem gráfica. Para isso o software é composto por uma ferramenta capaz de traduzir o modelo em UML para um correspondente em PDDL que pode ser utilizado pelos planejadores automáticos.

O ambiente itSIMPLE incorpora um conjunto de linguagens de representação e teorias capazes de lidar com requisitos e engenharia do conhecimento, associando-os ao ciclo de vida de projeto, conforme mostra a Figura 2. Entre as diversas linguagens de especificação e modelagem, o itSIMPLE propõe a linguagem semi-formal UML (OMG, 2001) (linguagem diagramática bastante conhecida e utilizada na Engenharia de Requisitos e de Software) para as etapas iniciais na construção do modelo e conhecimento do domínio. Redes de Petri clássicas (MURATA, 1989) são usadas para verificação dinâmica dos planos. E, como a comunidade de Planeamento Automático em IA utiliza PDDL como um padrão de representação de domínios e como representação de entrada para os planejadores, o itSIMPLE também integra esta linguagem (até a versão 3.0 da PDDL) no seu processo de projeto. (VAQUERO, 2009)

Na versão atual, itSIMPLE pode se comunicar com os principais planejadores disponíveis na literatura. Portanto, o processo de escolha do planejador ainda não é feito de fato e, nesta versão o sistema simplesmente prepara o domínio de aplicação (modelo do chão de fábrica e mais do problema de planejamento) para todos os planejadores incluídos. Em uma versão futura pretende-se eliminar este processo, elegendo somente os planejadores mais apropriados para o problema em questão. (TAVARES et al, 2011)

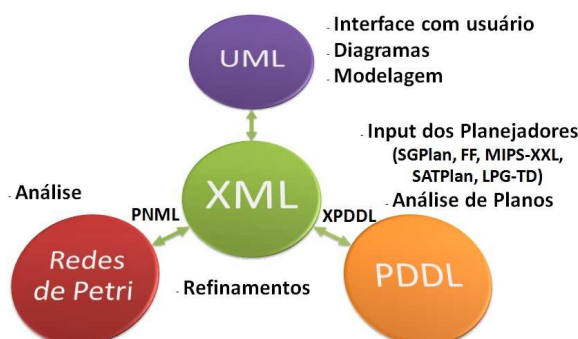


Figura 2: Estrutura de representações e linguagens do sistema itSIMPLE. (TAVARES et al, 2011)

O itSIMPLE é hoje um sistema de domínio público¹. Este sistema participou das três edições da competição *International Competition on Knowledge Engineering For Planning and Scheduling* (ICKEPS), ficando em primeiro lugar na terceira edição, realizada em 2009. (TAVARES et al, 2011)

Como o objetivo deste trabalho é apresentar uma metodologia de integração entre planejamento automático e linguagem de programação de máquina, especificamente, uma integração entre itSIMPLE e CLP, vamos agora definir o CLP e apresentar suas principais linguagens de programação, de acordo com a IEC 1131-3.

Os Controladores Lógicos Programáveis ou CLPs são equipamentos eletrônicos utilizados em sistemas de automação flexível. São ferramentas de trabalho muito úteis e versáteis para aplicações em sistemas de acionamentos e controle, e por isso são utilizados em grande escala no mercado industrial. Permitem desenvolver e alterar facilmente a lógica para acionamento das saídas em função das entradas. Desta forma, podemos associar diversos sinais de entrada para controlar diversos atuadores ligados nos pontos de saída. (TAVARES, 2011 apud OLIVEIRA, 1993)

Com a evolução dos microprocessadores ocorreu o aumento da capacidade de processamento e de memória dos CLP's que passaram a se tornar atrativos, além de todos os campos de atividade industrial, também, a área de automação predial atuando no controle de climatização, alarmes, iluminação. As atuais gerações de controladores possuem funções avançadas de controle, disponibilidade de grande número de entradas e saídas, além de uma grande facilidade de comunicação com sistemas supervisórios e sensores e atuadores inteligentes.

As linguagens utilizadas em CLP's variam basicamente entre as cinco linguagens definidas pela IEC 1131-3, sendo elas a Linguagem *Ladder*, *Diagrama de Blocos Funcionais*, *Gráfico Sequencial de Funções* (SFC – *Sequential Flow Chart*), podendo ser encontradas ainda linguagens textuais como *Lista de Instruções* e *Texto Estruturado*.

O diagrama *Ladder* é uma linguagem de programação gráfica, derivada da representação do diagrama de circuitos de controladores a relé diretamente conectados. Este diagrama é caracterizado por conter linhas de corrente à direita e à esquerda do diagrama, estas estão ligadas através de caminhos de corrente com elementos comutadores (contatos normalmente abertos, contatos normalmente fechados) e bobinas. A Figura 3 ilustra um exemplo em linguagem diagrama *Ladder*.

1 itSIMPLE. Disponível em <http://code.google.com/p/itsimple/>

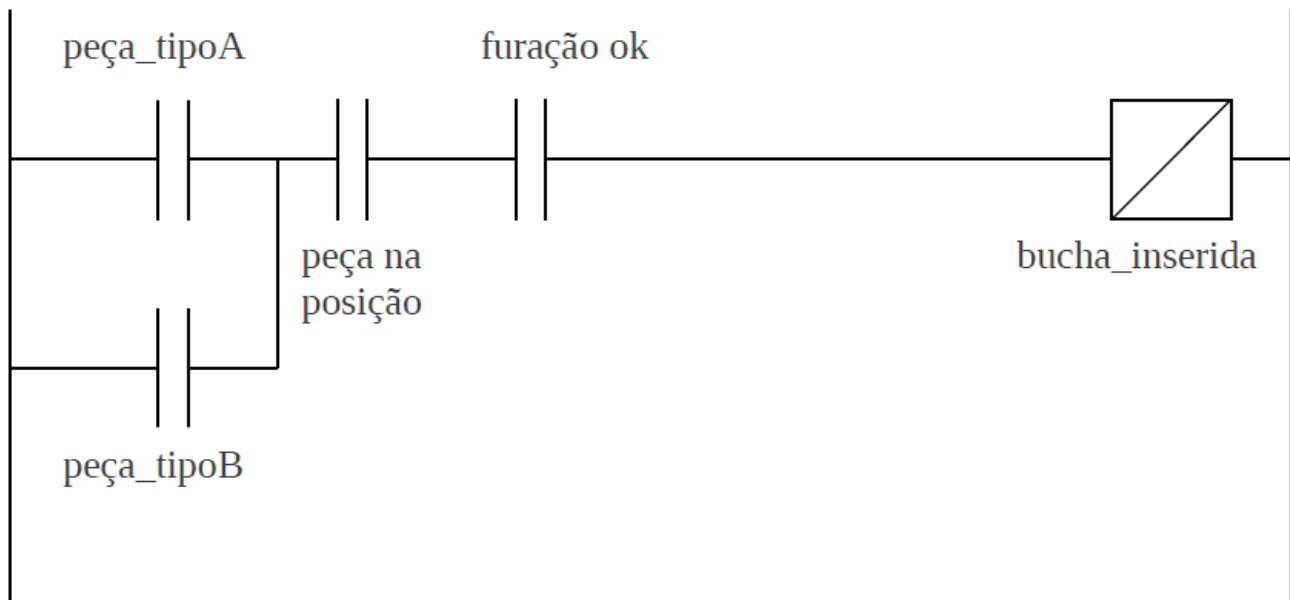


Figura 3: Exemplo de linguagem diagrama *Ladder*. (TAVARES, 2011)

Comercialmente, *SFC* e *Ladder* são as duas linguagens de maior representatividade no universo de linguagens para CLP. Segundo MIYAGI (1996), o diagrama *Ladder* corresponde a uma representação lógica baseada no diagrama de circuitos de relés, cuja utilização era ampla antes do surgimento dos CLP's, o que explica a preferência por este tipo de linguagem pela maioria dos programadores. Já o *SFC* é uma linguagem gráfica derivada da Rede de Petri. O seu desenvolvimento é motivado pelo interesse em ferramentas gráficas que representem de forma explícita as funções para descrever processos sequenciais para aplicações industriais.

Tradicionalmente, os processos que necessitam de automação através do uso de CLP's são inicialmente modelados em *Rede de Petri* ou em *SFC* e posteriormente traduzidos para diagramas *Ladder* para enfim serem implementados no CLP. Os CLP's, em sua maioria, são aptos a receber diretamente o *SFC*, sem a necessidade de tradução externa *SFC-Ladder*. No entanto, o que se vê na maioria dos casos é utilização do diagrama *Ladder* para representar o modelo desenvolvido em *SFC*. Esta representação requer uma transcrição *SFC-Ladder*. O caso apresentado aqui considera a metodologia de transcrição *Grafcet-Ladder*.

Um diagrama *Ladder* pode ser dividido em três partes distintas, isto é, uma responsável pela ocorrência das transições (representa em maior grau o fluxo do processo), outra responsável pelas etapas (sequencializa o processo) e, finalmente, uma que realizará as ações operativas (em função da etapa ativa). A Figura 4 apresenta as partes do programa *Ladder* referente ao diagrama *Ladder*.

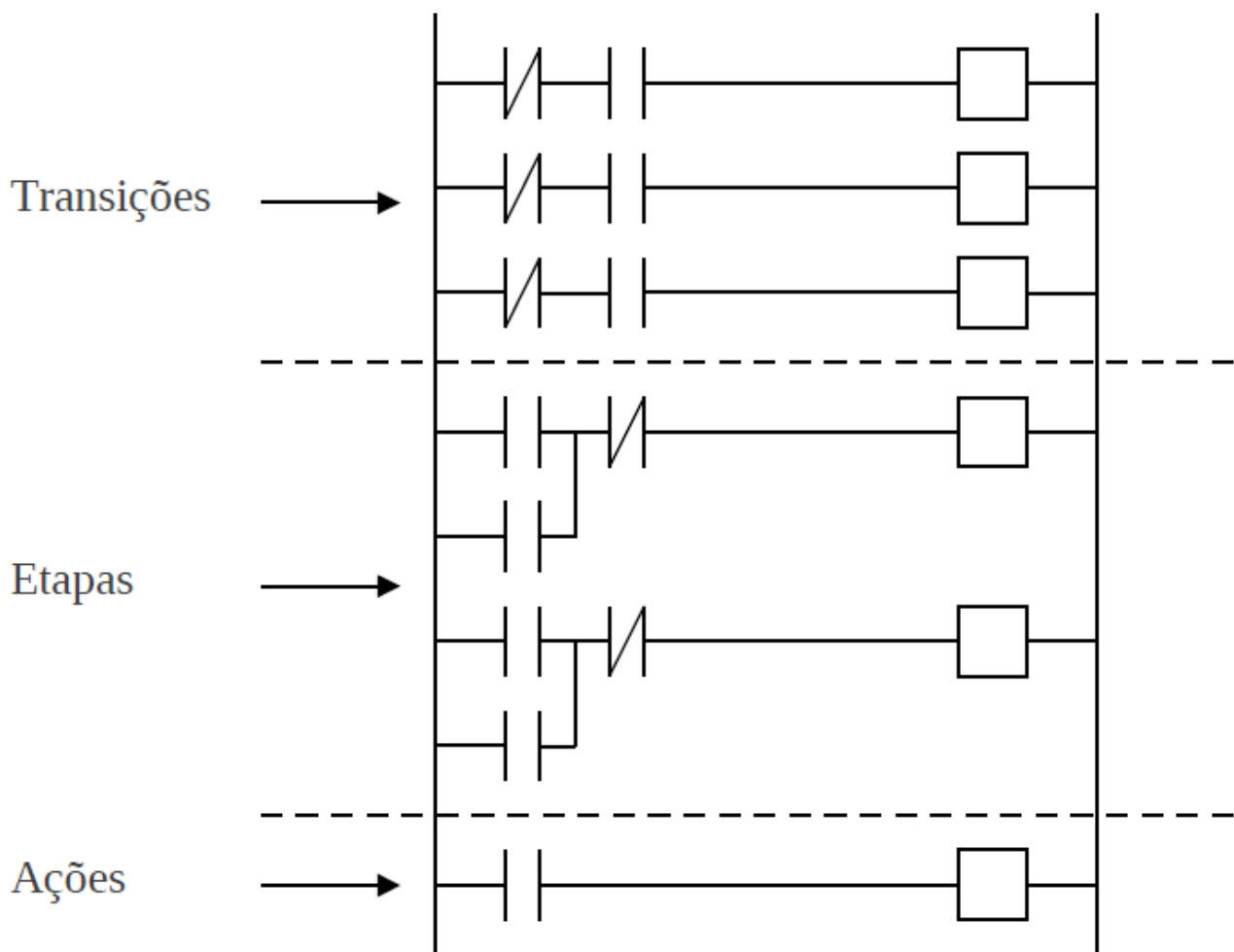


Figura 4: Diagrama de contatos implementados a partir de um *Grafcet*. (TAVARES, 2011)

Para transformar um *Grafcet* em Diagrama *Ladder*, é necessário criar, inicialmente, as tabelas de relacionamento entre sensores (sensibilidade do *Grafcet*) e entradas do CLP (I0, I1, .. In), transições e memórias internas (R0, R1, ... Rj), etapas e memórias internas (R10, R11, .. R1j), e ações com saídas do CLP (O0, O1, ... Ok). (TAVARES, 2011)

As transições serão saídas (bobinas) de entradas que são a(s) etapa(s) anterior(es) (Rx) e a sensibilidade da transição. A Figura 5 mostra como a transição R0 é acionada pela etapa anterior (R10) e da sensibilidade I0 (botão de partida S0).

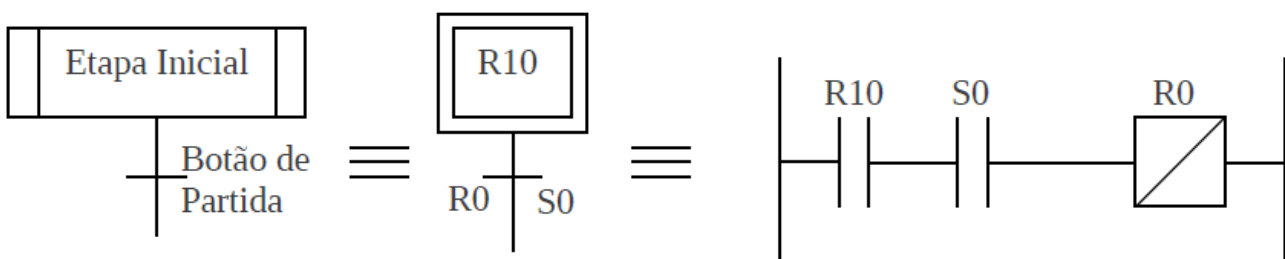


Figura 5: Diagrama de contatos referente à transição "Botão de Partida". (TAVARES, 2011)

Em seguida deve-se montar as partes referentes às etapas, ligando-as quando a(s) transição(ões) anterior(es) for(em) válida(s) e desligando-as quando a(s) transição(ões) posterior(es) for(em) válida(s). Vale ressaltar que para que as etapas se mantenham ativas, elas mesmas devem estar colocadas como entradas, recursivamente. A Figura 6 ilustra esta representação.

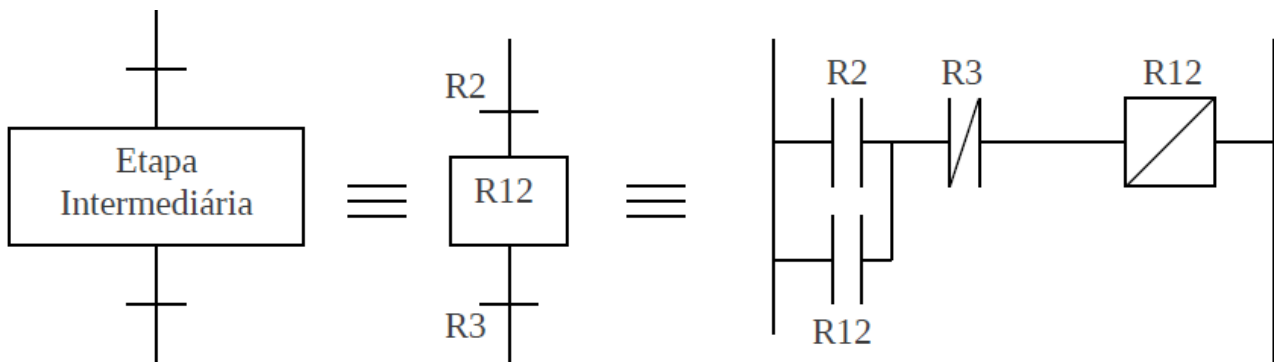


Figura 6: Diagrama de contatos referente à “Etapa Intermediária”. (TAVARES, 2011)

Finalmente, o diagrama referente às ações deverá acioná-las (bobinas) quando a etapa estiver ativa. A Figura 7 serve de referência à etapa intermediária com a ação de acionar uma válvula (Saída O2 do CLP).

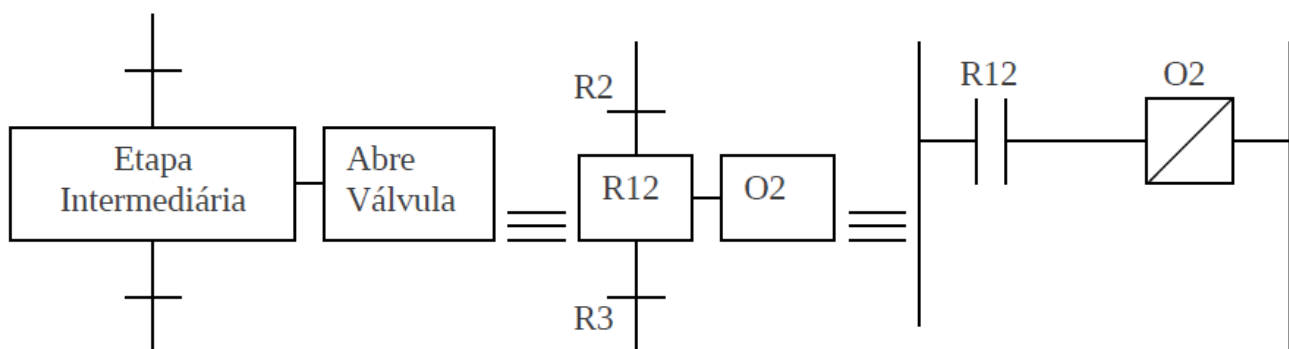


Figura 7: Diagrama de contatos referente à ação “Abre Válvula”. (TAVARES, 2011)

Apresentada a fundamentação teórica para a realização deste trabalho, será introduzida a seguir a metodologia utilizada para o seu desenvolvimento.

Inicialmente foi proposto o desenvolvimento de uma bancada didática onde fosse possível avaliar a aplicação do planejamento automático em sistemas práticos. Para isto, partiu-se da modelagem de uma 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 teria ainda de atender demandas específicas de clientes.

O processo de modelagem do domínio foi totalmente desenvolvido utilizando-se o software itSIMPLE (VAQUERO et al, 2009), seguindo a metodologia de modelagem em UML.

Em função da modelagem, foi desenvolvida uma bancada didática física que simula um sistema de logística e transporte de um fornecedor para dois distribuidores utilizando um veículo controlado por um CLP. Cada distribuidor possui uma bomba que serve a diversos clientes e o sistema de planejamento deve definir as entregas do veículo de forma a atender as demandas dos clientes dos distribuidores. O serviço de entregas do veículo é requisitado em função da variação do nível dos distribuidores. (FONSECA, 2011)

A Figura 8 apresenta uma representação esquemática e o esboço tridimensional da bancada didática.

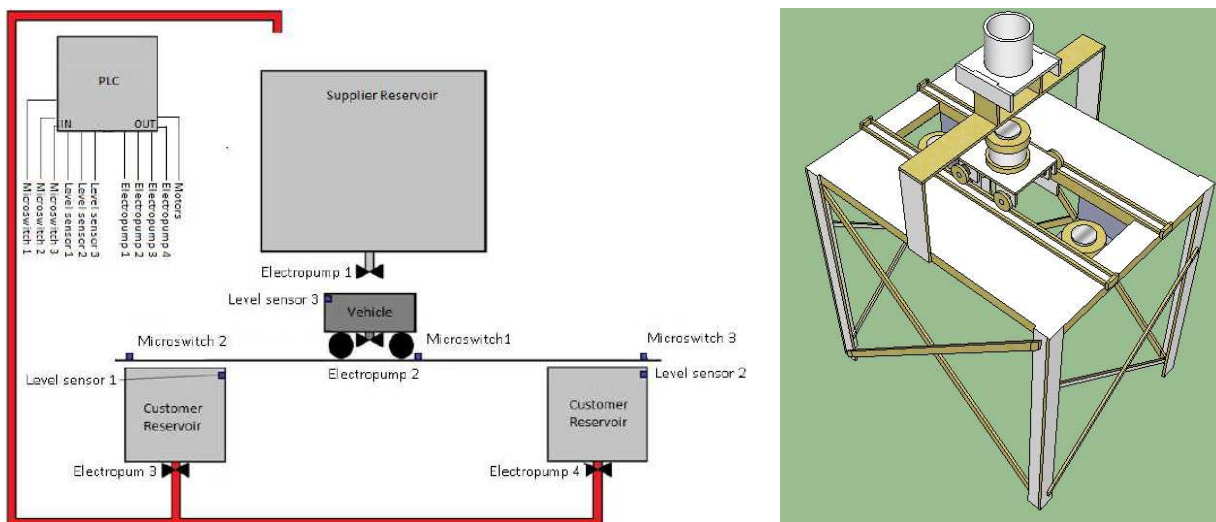


Figura 8: Apresentação da bancada didática utilizada.

A bancada didática 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. O veículo possui dois motores 12Vcc para se movimentar (um driver L293 controla o sentido de giro dos motores). 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 resistivos 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. Para facilitar os experimentos com a bancada, utilizou-se água como produto da cadeia de suprimento.

O processo de modelagem inicia-se com o diagrama de casos de uso, onde 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 (*customer*). O veículo pode se mover (*move*), carregar (*load*) e descarregar (*unload*). Já o distribuidor pode efetuar uma venda completa (*completeorder*), ou uma venda parcial (*partialorder*) e sua complementação (*finalorder*). Especificamente, o caso de uso descarregar (*unload*) está vinculado diretamente tanto com o ator veículo (*vehicle*) quanto com o ator distribuidor (*customer*). Este diagrama é a parte do modelo em que o projetista define os requisitos e restrições do sistema real. Nele ressaltam-se as características de cada ator e definem-se as pré-condições e pós-condições de cada caso de uso. É a partir destas informações que se torna possível a elaboração das regras dos objetos que compõem o modelo. A Figura 9 apresenta o diagrama de casos de uso do domínio em questão.

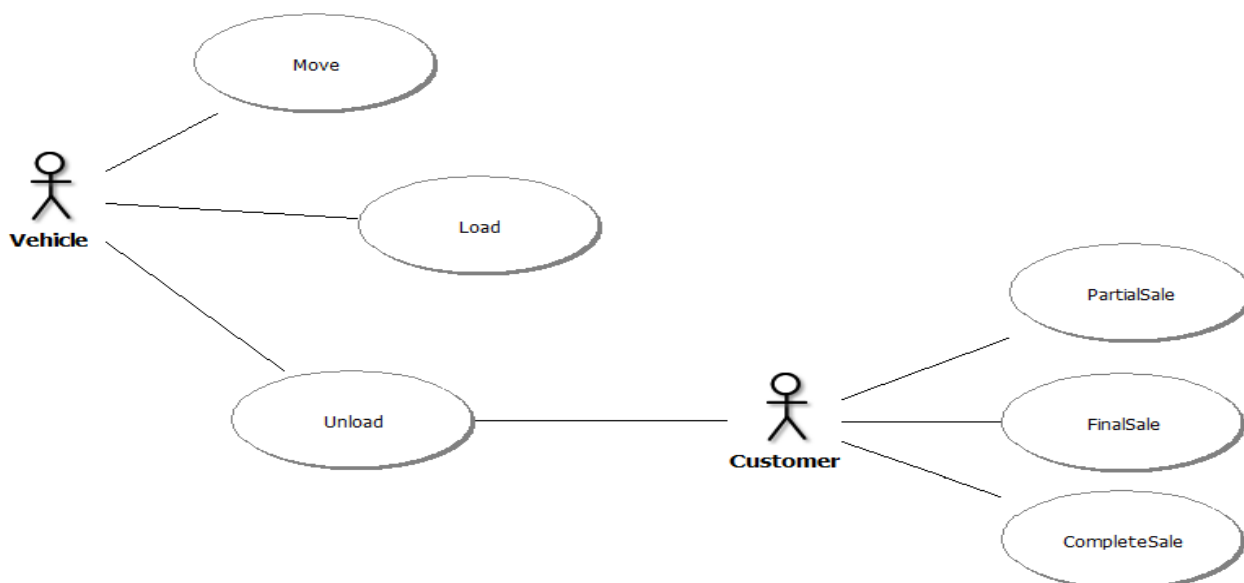


Figura 9: Diagrama de casos de uso da Bancada

Após a elaboração do diagrama de casos de uso é necessário definir as classes que irão instanciar o modelo e as relações entre estas classes. Isto é feito através do diagrama de classes. A Figura 10 apresenta o diagrama de classes da bancada.

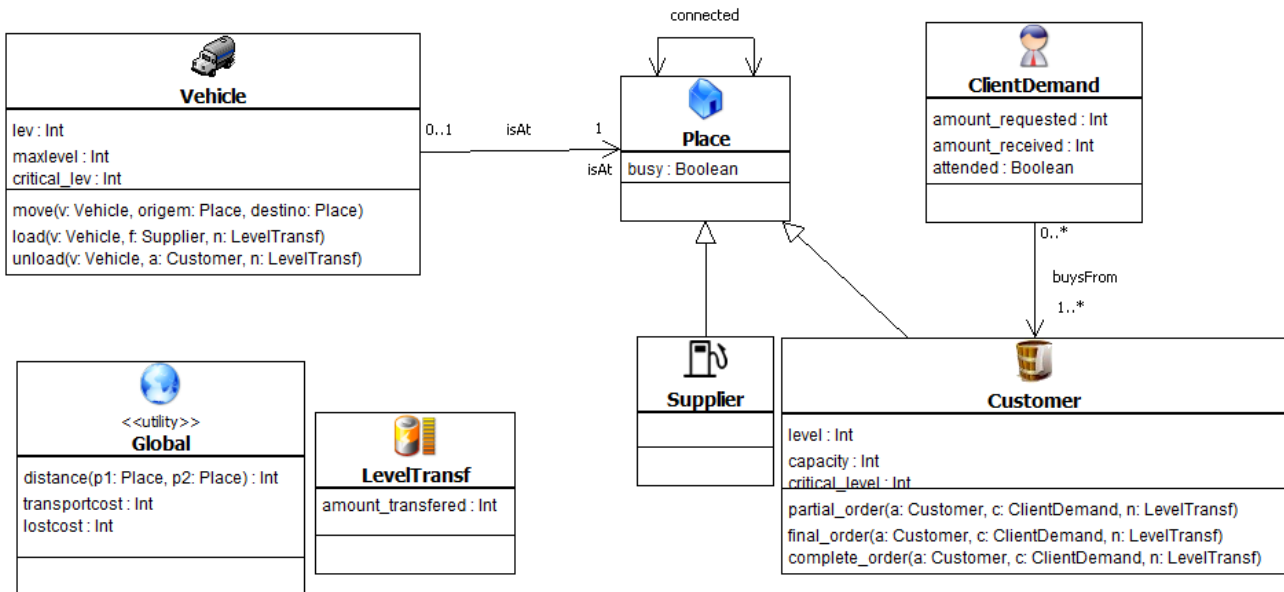


Figura 10: Diagrama de classes da Bancada

Cada classe deve possuir os atributos e métodos que representem o comportamento dos objetos no sistema real. Por exemplo, a classe veículo (*vehicle*) possui as informações necessárias para instanciar todos os objetos do tipo veículo, ou seja, um objeto do tipo veículo possui dados que representam o seu nível máximo, nível crítico e o seu nível em um determinado momento. Estes dados são tratados na classe veículo (*vehicle*) como atributos e foram definidos como: nível máximo (*maxlevel*), nível crítico (*critical_level*) e nível do objeto no momento presente (*lev*). 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 (*move*), carregar seu reservatório no momento em que estiver estacionado em um fornecedor (*load*) e descarregar o seu reservatório, abastecendo um distribuidor no qual está estacionado (*unload*). Estas ações são suportadas pelos operadores da classe veículo (*vehicle*): *move*, *load* e *unload*. A modelagem deste domínio pode ser encontrada mais bem detalhada em (FONSECA, TAVARES, 2011).

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 adéque às características do projeto, que podem ser minimizar ou maximizar tempo de processamento, tempo de uma operação, custo de uma operação, etc.

As métricas desenvolvidas neste modelo, que podem ser consideradas como custos a minimizar, são *transportcost* (custo do transporte), de acordo com a Eq.2, e *lostcost* (custo de perda por não se atender uma demanda completamente), de acordo com a Eq.3. O custo de perda naturalmente é algo concordado através de uma cláusula do contrato entre fornecedor e distribuidor e, para este caso específico, foi escolhido arbitrariamente um peso de valor 50.

$$transportcost = transportcost + distance(origem, destino)(v.lev + 1) \quad (2)$$

$$lostcost = lostcost + 50(amount_{requested} - amount_{received}) \quad (3)$$

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 (Figura 11) e do Distribuidor (Figura 12). 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 veículo (*Vehicle*), cujos objetos podem se movimentar de um lugar a outro, ou ter seu nível afetado por operações de carregar e descarregar, e outro referente à classe distribuidor (*Customer*), cujos objetos podem ter seu nível modificado por operações do tipo descarregar (*Unload*), entrega parcial (*PartialOrder*), entrega final (*FinalOrder*) e entrega completa (*CompleteOrder*).

As pré-condições e pós-condições definidas no Diagrama de Casos de Uso são agora traduzidas como as regras necessárias para a ocorrência das ações relacionadas com os Casos de Uso da Figura 9. Por exemplo, a pré-condição para que a ação *move* ocorra é que: o veículo deve estar estacionado em um 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. A pós-condição para esta 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.2.

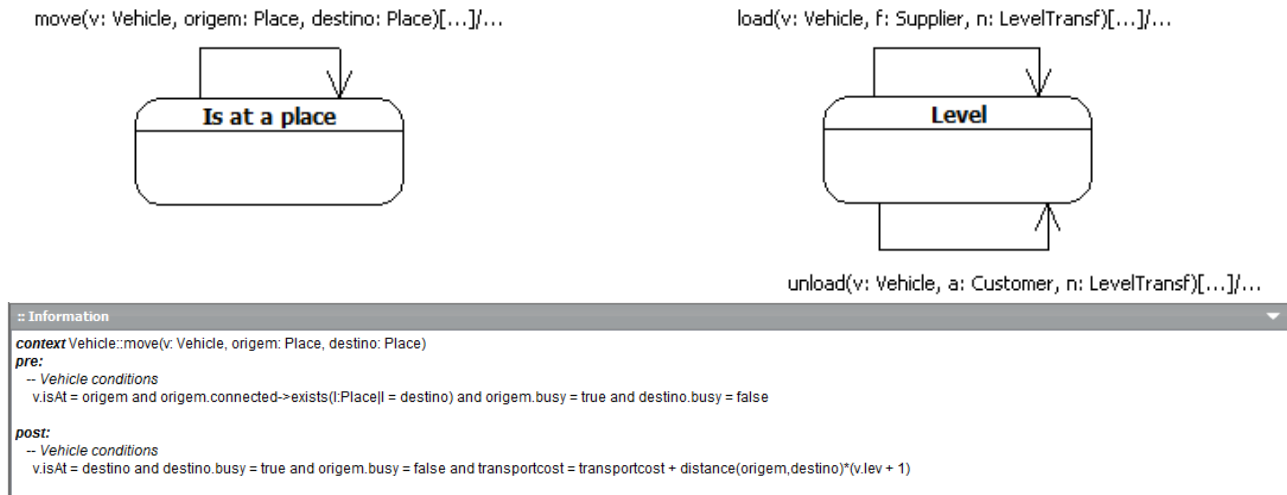


Figura11: Diagrama de Estados do Veículo com as regras para a ação *move*.

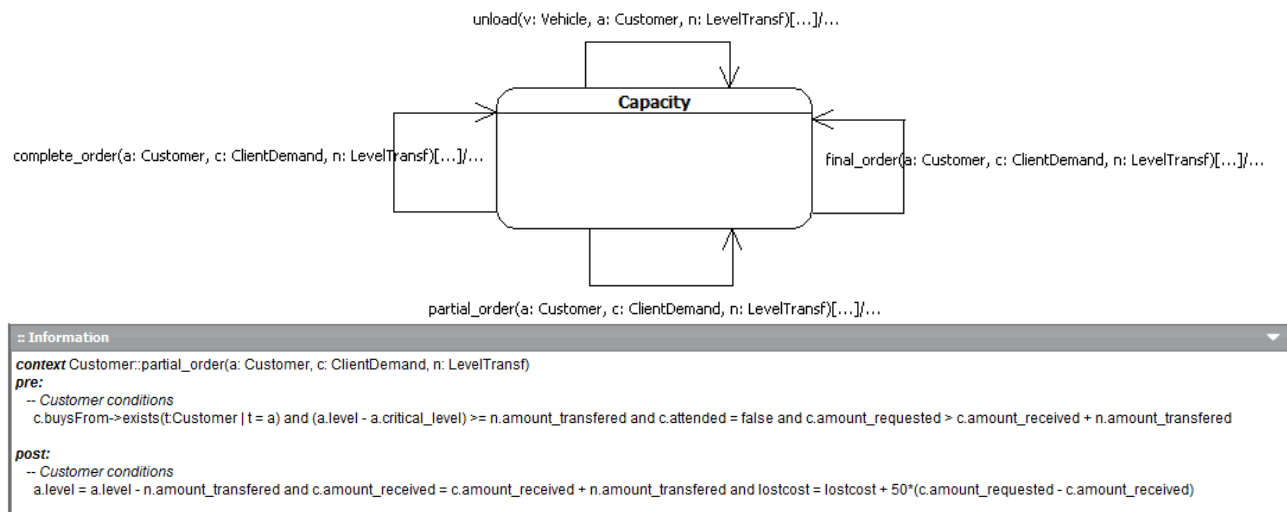


Figura 12: Diagrama de Estados do Distribuidor com as regras para a ação *partial_order*.

O problema a ser resolvido é movimentar, carregar e descarregar o veículo *v1* de forma a atender a demanda dos clientes dos distribuidores. Foi definido como problema a ser resolvido 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%. Este estoque intermediário dos distribuidores poderia estar associado a uma previsão de demanda para os clientes em um espaço de tempo subsequente. As Figuras 13 e 14 mostram o estado inicial e final correspondente a esse problema.

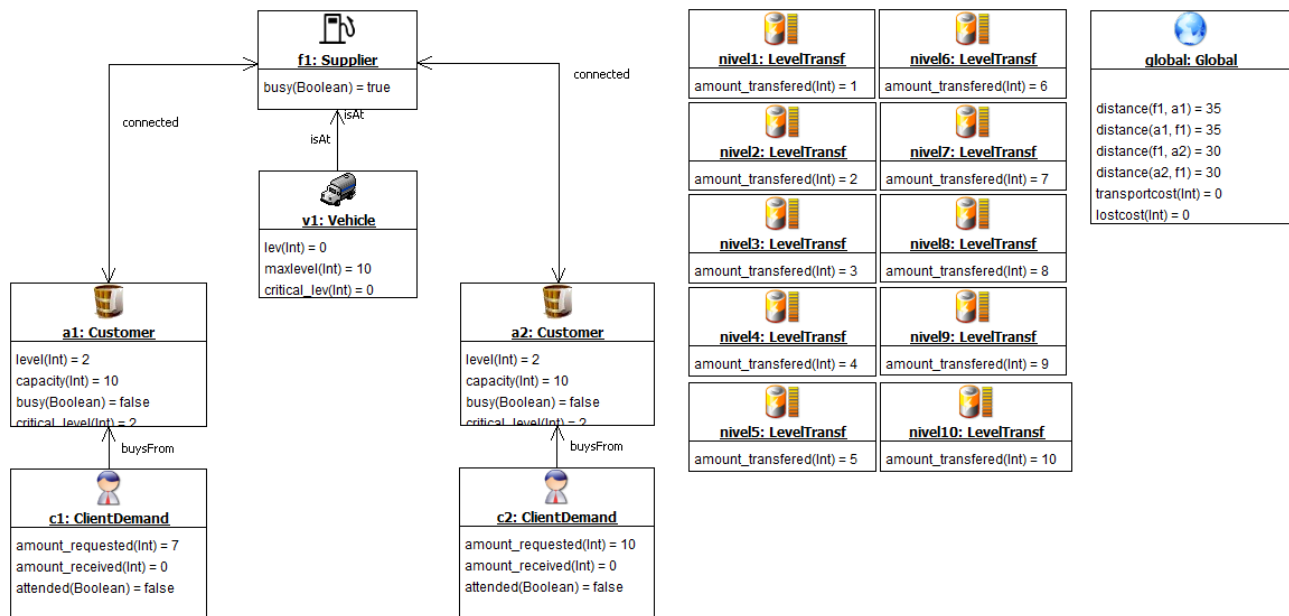


Figura 13: Situação Inicial

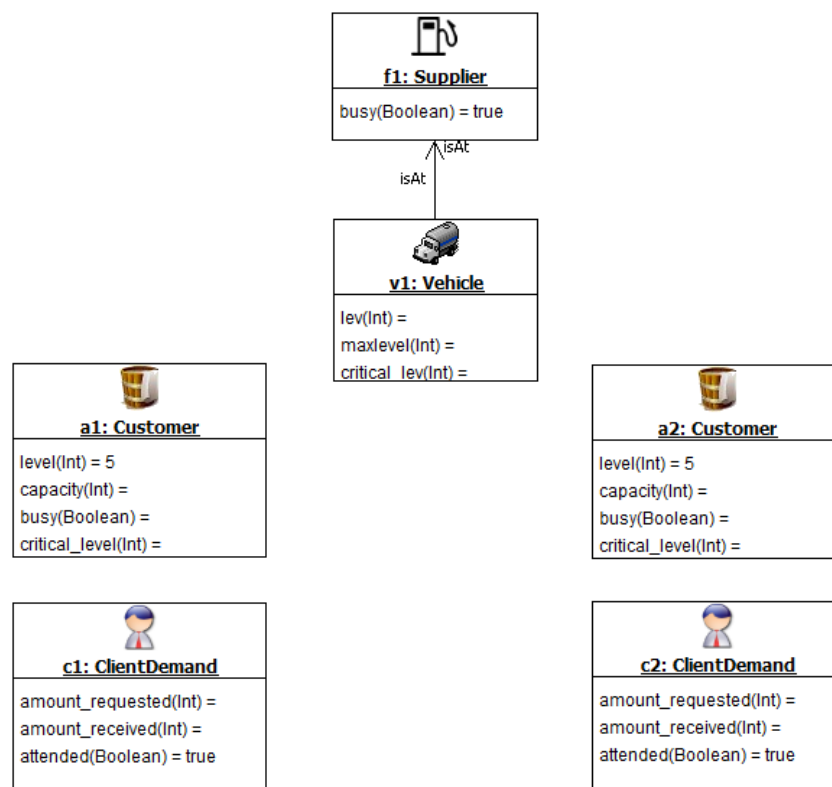


Figura 14: Situação Final

O itSIMPLE possui um número considerável de planejadores automáticos, tais como SGPlan6 (HSU; WAH, 2008), Metric-FF (HOFFMANN, 2003) etc. Neste trabalho será apresentado o resultado obtido com o planejador Metric-FF, o qual gerou o plano-solução descrito abaixo.

```

1: LOAD V1 F1 NIVEL8
2: MOVE V1 F1 A1
3: UNLOAD V1 A1 NIVEL8
4: MOVE V1 A1 F1
5: LOAD V1 F1 NIVEL8
6: MOVE V1 F1 A2
7: UNLOAD V1 A2 NIVEL8
8: MOVE V1 A2 F1
9: COMPLETE_ORDER A1 C1 NIVEL7
10: LOAD V1 F1 NIVEL2
11: MOVE V1 F1 A1
12: UNLOAD V1 A1 NIVEL2
13: MOVE V1 A1 F1
14: PARTIAL_ORDER A2 C2 NIVEL8
15: LOAD V1 F1 NIVEL5
16: MOVE V1 F1 A2
17: UNLOAD V1 A2 NIVEL5
18: MOVE V1 A2 F1
19: FINAL_ORDER A2 C2 NIVEL2

```

Esta possível solução faz com que o veículo *v1* realize as ações sequencialmente para levar o sistema da situação inicial à situação objetivo.

O plano gerado pode ser visto como uma sequência de ações necessárias, e assim, atingir a solução do problema. Antes disso, os sensores e atuadores devem ser identificados.

Para cada sensor e atuador da bancada foram atribuídas variáveis no CLP Omron (modelo CJ1M CPU13 ETN21) conforme Tabelas 1 e 2.

Tabela 1: Sensibilidade x Entrada CLP

Sensibilidade	Entrada CLP
Fim de Curso 1 (fc1) – Fornecedor	Entrada Digital 1 (I0.00)
Fim de Curso 2 (fc2) – Distribuidor 1	Entrada Digital 2 (I0.01)
Fim de Curso 3 (fc3) – Distribuidor 2	Entrada Digital 3 (I0.02)
Sensor de Nível 1 (S1) - Veículo	Entrada Analógica 1 (I1.00)
Sensor de Nível 2 (S2) – Distribuidor 1	Entrada Analógica 2 (I1.01)
Sensor de Nível 3 (S3) – Distribuidor 2	Entrada Analógica 3 (I1.02)

Tabela 2: Ação x Saída CLP

Ação	Saída CLP
Eletrobomba 1 (B1) – Fornecedor	Saída Digital 1 (Q1.00)
Eletrobomba 2 (B2) – Veículo	Saída Digital 2 (Q1.01)
Movimento para esquerda – Motores	Saída Digital 3 (Q1.02)
Movimento para a direita – Motores	Saída Digital 4 (Q1.03)
Eletrobomba 3 (B3) – Distribuidor 1	Saída Digital 5 (Q1.04)
Eletrobomba 4 (B4) – Distribuidor 2	Saída Digital 6 (Q1.05)

A relevância deste projeto está vinculada exatamente na aplicação prática do plano-solução apresentado na seção. Para isso, foi definido um método de integração entre o plano-solução e a linguagem de programação de máquina referente ao sistema real, neste caso definida como um diagrama *Ladder* em um CLP Omron CJ1M CPU13 ETN21. Inicialmente, a sequência de ações definidas pelo plano-solução é traduzida em um modelo a eventos discretos do tipo Gráfico Funcional de Comandos Etapa-Transição, conhecido também como Grafcet. O desenvolvimento do modelo segue uma lógica em que cada ação da sequência definida no plano-solução está relacionada com uma etapa no Grafcet. O modelo inicia-se no *snapshot* inicial e finaliza sua rotina no *snapshot* final. Os estados intermediários, diretamente vinculados com as pré-condições e pós-condições de cada ação possuem relação direta com as transições entre etapas do modelo em Grafcet. O modelo em Grafcet correspondente ao plano-solução sugerido pelo planejador Metric-FF é apresentado parcialmente na Figura 15.

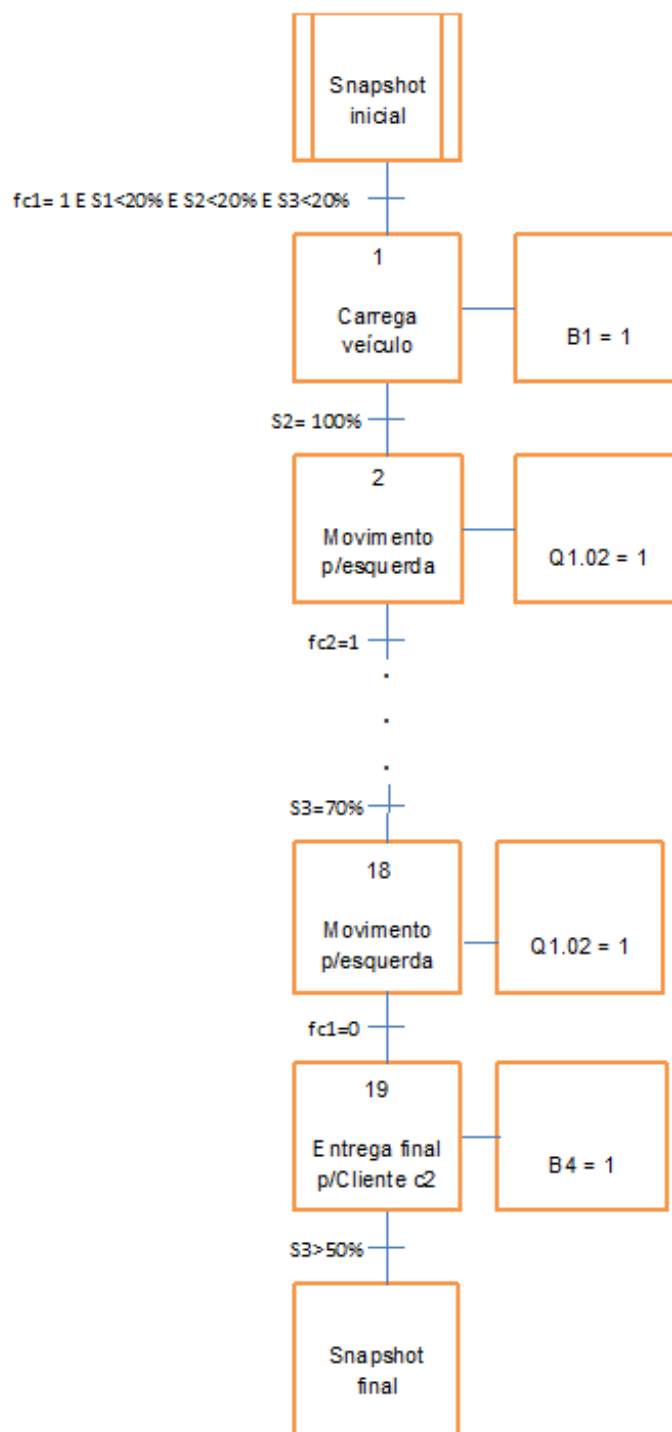


Figura 15: GRAFCET correspondente ao plano-solução.

A comparação entre os níveis percebidos pelos sensores com os níveis esperados em cada estado do plano foi realizada através da utilização de memórias auxiliares. Para isso, armazenou-se os valores percebidos pelos sensores de nível em memórias auxiliares, como descrito na Tabela 3. Os valores percebidos pela entrada analógica I0.00 (sensor de nível do veículo) foram armazenadas na posição de memória D100; os valores percebido pela entrada analógica I0.01 (sensor de nível do distribuidor 1) foram armazenados na posição de memória D101; e os valores percebidos pela entrada analógica I0.02 (sensor de nível do distribuidor 2) foram armazenados na posição de memória D102.

Tabela 3: Entrada analógica x Posição de memória auxiliar

Entrada analógica	Memória auxiliar
Entrada Analógica 1 (I1.00)	D100
Entrada Analógica 2 (I1.01)	D101
Entrada Analógica 3 (I1.02)	D102

Uma característica fundamental dos sensores de nível utilizados na bancada é a variação do sinal em função da variação do nível. No caso, os sensores utilizados são do tipo resistivo. Para que estes sensores pudessem ser utilizados na bancada, foi necessária a construção de um transmissor de corrente capaz de converter o sinal da variação de nível em um sinal correspondente (4-20mA) para a entrada analógica do CLP. Além disso, como estes dispositivos possuem um desvio considerável no seu range de variação da resistência, foi realizada uma calibração, conforme apresentado na Tabela 4.

Tabela 4: Tabela de calibração dos sensores de nível dos reservatórios.

Nível (ml)	Relação (%)	Sensor 1 (Veículo)	Sensor 2 (Distribuidor 1)	Sensor 3 (Distribuidor 2)
500	20	1322	736	1061
600	30	1610	1356	1298
700	40	1880	1620	1521
800	50	2115	1854	1773
900	60	2344	2140	1958
1000	70	2569	2389	2166
1100	80	2902	2623	2385
1200	90	3309	2938	2836
1300	100	3784	3358	3261

Para aplicar o plano gerado pelo planejador automático no CLP, tanto os atributos como os operadores de cada classe (representada no Diagrama de Classes) devem estar associados a um conjunto de estados dos sensores e atuadores da Bancada. Por exemplo, a ação *Action 2: move(v1, f1, a1)* corresponde ao movimento do veículo para o lado esquerdo, ou seja, saída digital 3 (Q1.02) habilitada e saída digital 4 (Q1.03) desabilitada. A pré-condição para a habilitação de Q1.02 é o veículo estar estacionado no fornecedor, ou seja, *fc1* acionado que é o mesmo que entrada digital 1 (I0.00) habilitada; o sensor de nível do veículo deve indicar 100 %, que corresponde à memória auxiliar D100=3784; e os sensores de nível dos distribuidores 1 e 2 devem indicar 20 %, que corresponde aos seguintes valores para as memórias auxiliares: D101=736 e D102=1061. Quando o sistema atingir o estado apresentado, a saída digital Q1.02 deverá ser habilitada, proporcionando a movimentação do veículo para o lado esquerdo até que a pós-condição para esta ação seja atingida. A pós-condição para esta ação é o veículo estacionado sobre o distribuidor *a1*, ou seja, *fc2* acionado ou entrada digital 2 (I0.01) habilitada e os níveis dos reservatórios semelhantes aos percebidos na pré-condição.

Isto deve ocorrer para todas as ações presentes no plano. Assim, a Tabela 5 representa parte da *Tabela De Para* necessária para facilitar a interface entre o plano-solução apresentado anteriormente e a construção do diagrama *Ladder* utilizado no CLP.

As ações do plano-solução são representadas no CLP através dos endereços de memória W10.xx onde xx é o valor dado a ação do plano-solução (Tabela 6).

Desse modo, partiu-se de um modelo da bancada, o qual foi utilizado por um planejador automático que gerou um plano a partir dos requisitos apresentados. Este plano foi traduzido para uma linguagem operacional, de modo que, aliada com o mapeamento de entradas e saídas do CLP, foi possível a confecção do diagrama *Ladder* correspondente ao plano-solução. A Figura 16 ilustra uma parte do diagrama *Ladder* obtido através da metodologia descrita para este artigo.

Neste caso, a ocasião da situação inicial W10.00 (*snapshot inicial*) habilita o relé W20.00 (*Trans0*). Este relé marca a transição entre a situação inicial e a Ação nº 1 (*Load v1 f1 level 8*). Esta ação está associada diretamente à energização da eletrobomba 1 através da habilitação da saída digital Q1.00. Além disso, a Ação nº 1 está condicionada com a não-habilitação da Transição 1 (entre essa ação e a seguinte). Esta, por sua vez, será habilitada assim que o sensor de nível do reservatório do veículo indicar 100%, que corresponde à memória auxiliar D100=3784. Neste momento, a Ação nº 0 é desabilitada e a Ação nº 1, habilitada. Este processo continua até o sistema atingir o *snapshot* final.

Tabela 5: Tabela De Para Parcial Plano Ladder

SNAPSHOT/AÇÃO	PLANO	CLP
Snapshot Initial	-	-
	(isAt v1 f1)	I0.00=1
	(=(lev v1)2)	D100=1322
	(=(level a1)2)	D101=736
	(=(level a2)2)	D102=1061
	(=(amount_reveived c1)0)	
	(=(amount_reveived c2)0)	
Action 1	(Load v1 f1 level 8)	Q1.00=1 até D100=3784 / W10.01
	(isAt v1 f1)	I0.00=1
	(=(lev v1)10)	D100=3784
	(=(level a1)2)	D101=736
	(=(level a2)2)	D102=1061
	(=(amount_reveived c1)0)	
	(=(amount_reveived c2)0)	
Action 2	(Move v1 f1 a1)	Q1.02=1 e Q1.03=0 até I0.01=1 / W10.02
	(isAt v1 a1)	I0.01=1
	(=(lev v1)10)	D100=3784
	(=(level a1)2)	D101=736
	(=(level a2)2)	D102=1061
	(=(amount_reveived c1)0)	
	(=(amount_reveived c2)0)	
.	.	.
.	.	.
.	.	.
Action 18	(Move v1 a2 f1)	Q1.02=1 e Q1.03=0 até I0.00=1 / W10.18
	(isAt v1 f1)	I0.00=1
	(=(lev v1)2)	D100=1322
	(=(level a1)5)	D101=1854

	(=(level a2)7)	D102=2166
	(=(amount_reveived c1)8)	
	(=(amount_reveived c2)7)	
Action 19	(Final_order a2 c2 level 2)	Q1.05=1 até D102=1958 / W10.19
Snapshot Goal	(isAt v1 f1)	I0.00=1
	(=(lev v1)2)	D100=1322
	(=(level a1)5)	D101=1854
	(=(level a2)5)	D102=1773
	(=(amount_reveived c1)10)	
	(=(amount_reveived c2)7)	

Tabela 6: Ações do Plano e Endereços de Memória do CLP

Ação	Endereço de Memória do CLP
1	W10.01
2	W10.02
3	W10.03
.	.
.	.
.	.
17	W10.17
18	W10.18
19	W10.19

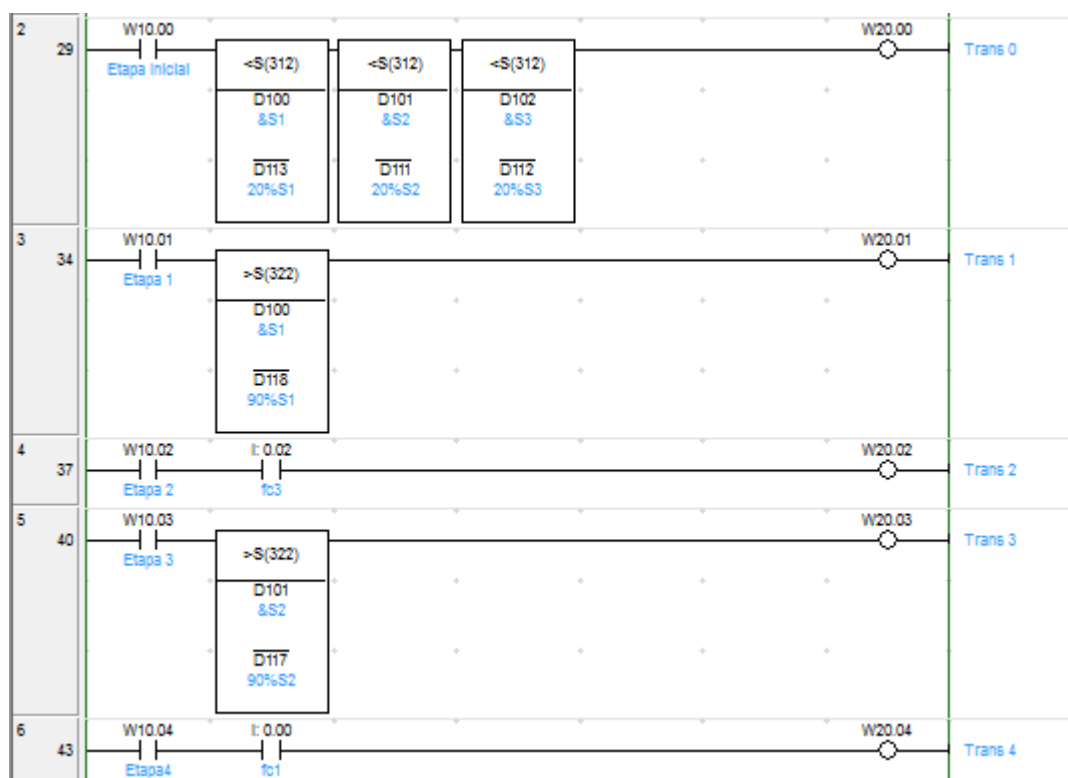


Figura 16: Diagrama Ladder parcial do domínio Bancada.

A bancada apresentada na Figura 8 pode ser modelada em um Grafcet como o apresentado na Figura 17. Neste caso o sistema atua o veículo em função do nível dos distribuidores, de modo que, assim que algum dos distribuidores indicar nível abaixo de 20%, o veículo realiza uma rotina composta de carregamento no fornecedor, movimentação, descarregamento no distribuidor correspondente e movimentação de retorno para a posição central. Caso ambos distribuidores solicitem produto, este modelo prioriza o distribuidor a1.

A tradução deste *Grafcet* para *Ladder* pode ser observada parcialmente na Figura 18.

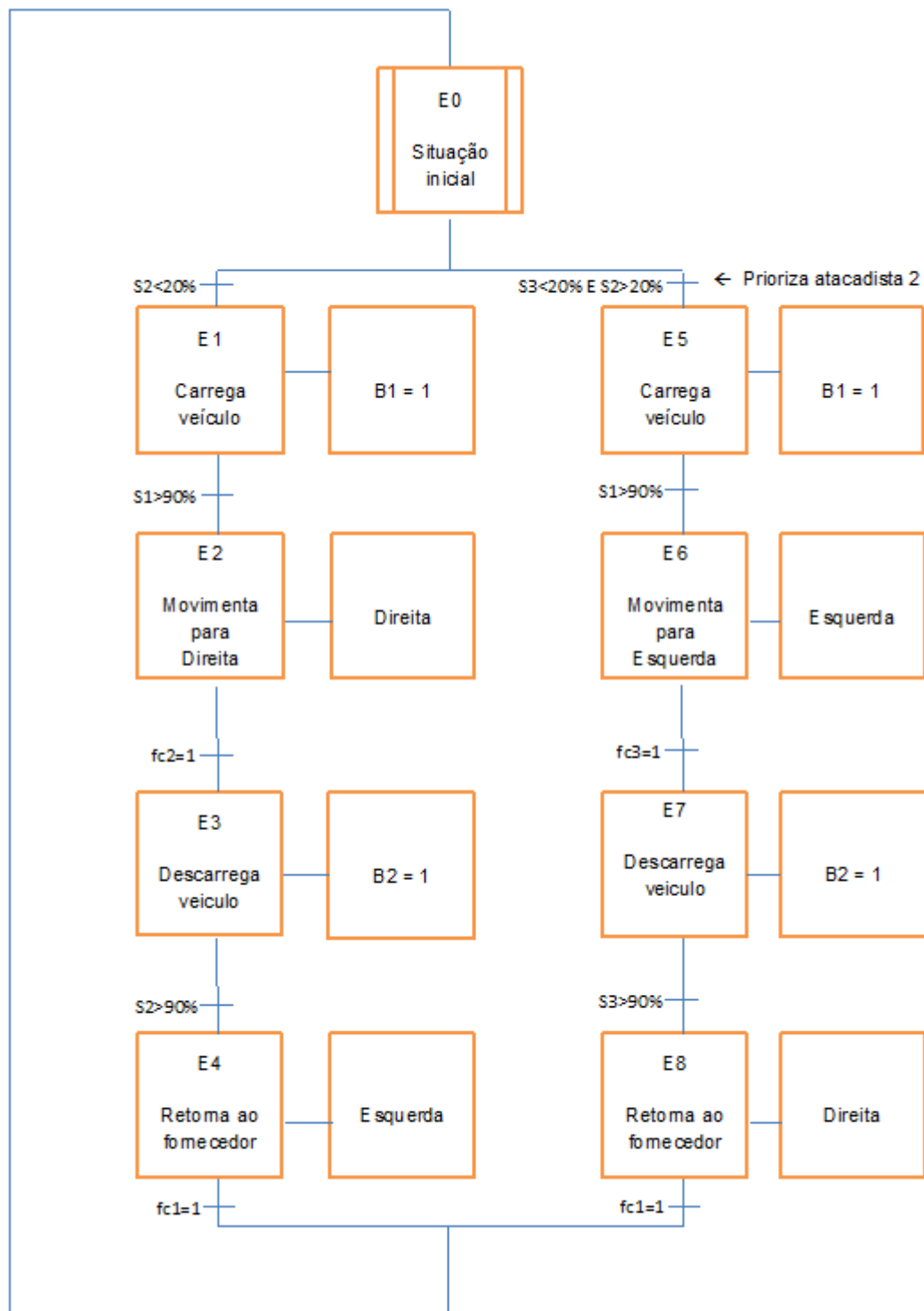


Figura 17: Modelagem da Bancada em SFC.

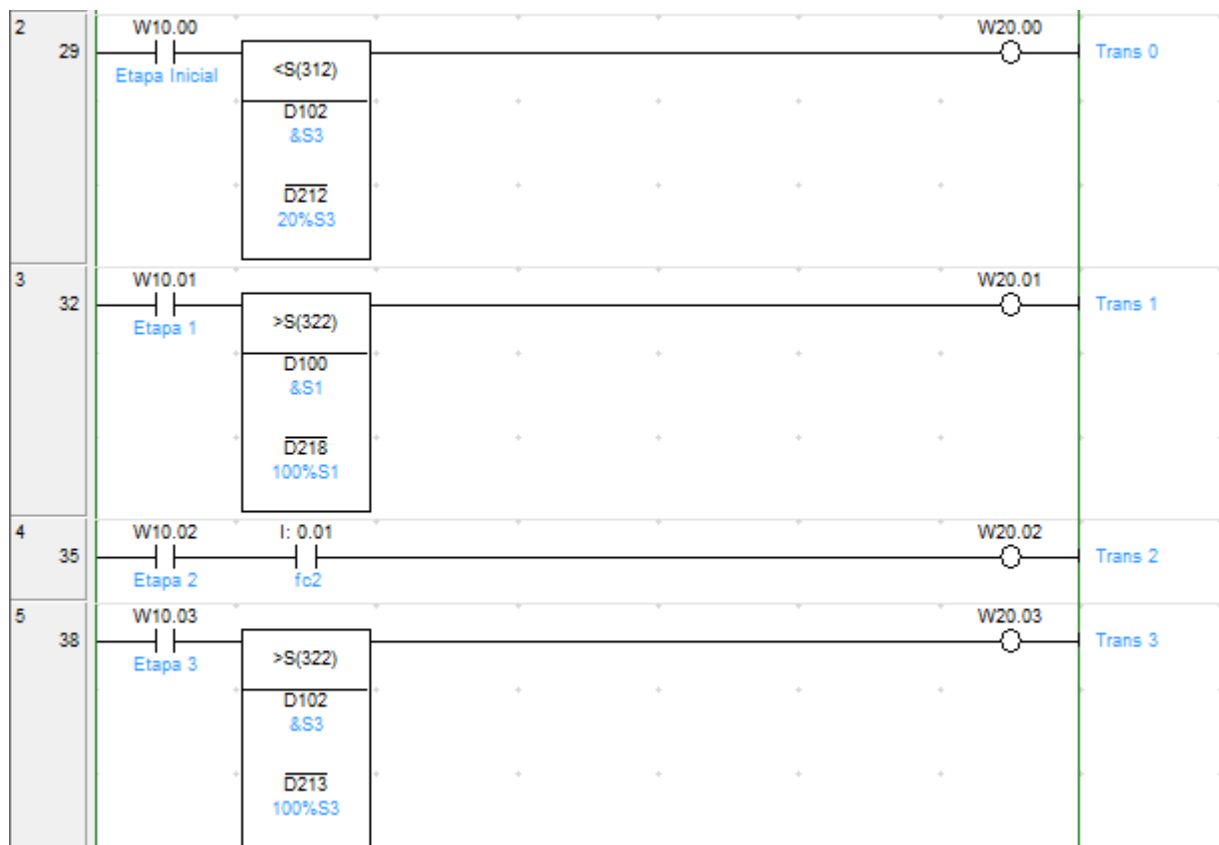


Figura 18: Diagrama *Ladder* parcial para a metodologia clássica.

Nos testes com este modelo, admitindo uma situação inicial semelhante àquela utilizada com o planejador automático e ainda, simulando demandas de clientes de acordo com o estabelecido no planejador, o sistema solucionou o problema através da sequência de ações definidas abaixo.

```

1: LOAD V1 F1 NIVEL8
2: MOVE V1 F1 A2
3: UNLOAD V1 A2 NIVEL8
4: MOVE V1 A2 F1
5: LOAD V1 F1 NIVEL8
6: MOVE V1 F1 A1
7: UNLOAD V1 A1 NIVEL8
8: MOVE V1 A1 F1
9: COMPLETE_ORDER A1 C1 NIVEL7
10: PARTIAL_ORDER A2 C2 NIVEL8
11: LOAD V1 F1 NIVEL8
12: MOVE V1 F1 A2
13: UNLOAD V1 A2 NIVEL8
14: MOVE V1 A2 F1
15: FINAL_ORDER A2 C2 NIVEL2

```

Esta sequência de ações pode ser encarada como o plano desenvolvido pelo CLP para solucionar o problema, a partir do modelo do sistema em Grafcet.

Nos parágrafos seguintes será apresentada a análise dos resultados para estas duas metodologias e a conclusão do trabalho abordará uma comparação entre os métodos.

Iniciando com a metodologia proposta, utilizando planejador automático, durante os testes foram avaliados os tempos entre ações e a mudança de sensibilidade percebida pelas entradas analógicas do CLP. A Tabela 7 apresenta o desenvolvimento do plano-solução obtido com o planejador automático Metric-FF.

Tabela 7: Desenvolvimento do plano-solução

Ação	Estado	S1	S2	S3	fc1	fc2	fc3	Cronômetro (min:seg)
	Estado inicial	1318	172	625	1	0	0	00:00
1: (Load v1 f1 level 8)								
	Estado 1	3800	172	625	1	0	0	01:30
2: (Move v1 f1 a1)								
	Estado 2	3800	172	625	0	1	0	01:42
3: (Unload v1 a1 level 8)								
	Estado 3	1305	3355	625	0	1	0	03:36
4: (Move v1 a1 f1)								
	Estado 4	1305	3355	625	1	0	0	03:47
5: (Load v1 f1 level 8)								
	Estado 5	3798	3355	625	1	0	0	05:24
6: (Move v1 f1 a2)								
	Estado 6	3798	3355	625	0	0	1	05:35
7: (Unload v1 a2 level 8)								
	Estado 7	999	3355	3433	0	0	1	07:37
8: (Move v1 a2 f1)								
	Estado 8	999	3355	3433	1	0	0	07:47
9: (CompleteOrder a1 c1 nivel 7)								
	Estado 9	999	1400	3433	1	0	0	08:24
10: (Load v1 f1 level 2)								
	Estado 10	2086	1400	3433	1	0	0	09:10
11: (Move v1 f1 a1)								
	Estado 11	2086	1400	3433	0	1	0	09:22
12: (Unload v1 a1 level 2)								
	Estado 12	1784	1825	3433	0	1	0	09:49
13: (Move v1 a1 f1)								
	Estado 13	1784	1825	3433	1	0	0	09:59
14: (PartialOrder a2 c2 nivel 8)								
	Estado 14	1784	1825	1073	1	0	0	10:45
15: (Load v1 f1 level 5)								
	Estado 15	2557	1825	1073	1	0	0	11:27
16: (Move v1 f1 a2)								
	Estado 16	2557	1825	1073	0	0	1	11:37
17: (Unload v1 a2 level 5)								
	Estado 17	724	1825	2567	0	0	1	13:13
18: (Move v1 a2 f1)								
	Estado 18	724	1825	2567	1	0	0	13:24
19: (Final_order a2 c2 level 2)								
	Estado Objetivo	724	1825	1735	1	0	0	13:46

Para validar o modelo desenvolvido no itSIMPLE, foram contabilizados os custos reais do plano-solução e comparados com os custos planejados. A Tabela 8 descreve o desenvolvimento dos custos do plano e a Figura 19 apresenta um gráfico para o desenvolvimento obtido pelo custo de transporte planejado (*Planning Transport Cost*) e o pelo custo de transporte real (*Real Transport Cost*).

Tabela 8: Desenvolvimento dos custos planejado e real.

Ação	Planning Transport Cost	Planning Lost Cost	Real Transport Cost	Real Lost Cost
1: (Load v1 f1 level 8)	0	0	0	0
2: (Move v1 f1 a1)	385	0	386,4	0
3: (Unload v1 a1 level 8)	385	0	386,4	0
4: (Move v1 a1 f1)	490	0	487,2	0
5: (Load v1 f1 level 8)	490	0	487,2	0
6: (Move v1 f1 a2)	820	0	818,1	0
7: (Unload v1 a2 level 8)	820	0	818,1	0
8: (Move v1 a2 f1)	910	0	855,3	0
9: (CompleteOrder a1 c1 nivel 7)	910	0	855,3	0
10: (Load v1 f1 level 2)	910	0	855,3	0
11: (Move v1 f1 a1)	1085	0	1061,1	0
12: (Unload v1 a1 level 2)	1085	0	1061,1	0
13: (Move v1 a1 f1)	1190	0	1223,5	0
14: (PartialOrder a2 c2 nivel 8)	1190	105	1223,5	105,7
15: (Load v1 f1 level 5)	1190	105	1223,5	105,7
16: (Move v1 f1 a2)	1430	105	1462,0	105,7
17: (Unload v1 a2 level 5)	1430	105	1462,0	105,7
18: (Move v1 a2 f1)	1520	105	1492,0	105,7
19: (Final_order a2 c2 level 2)	1520	105	1492,0	105,7

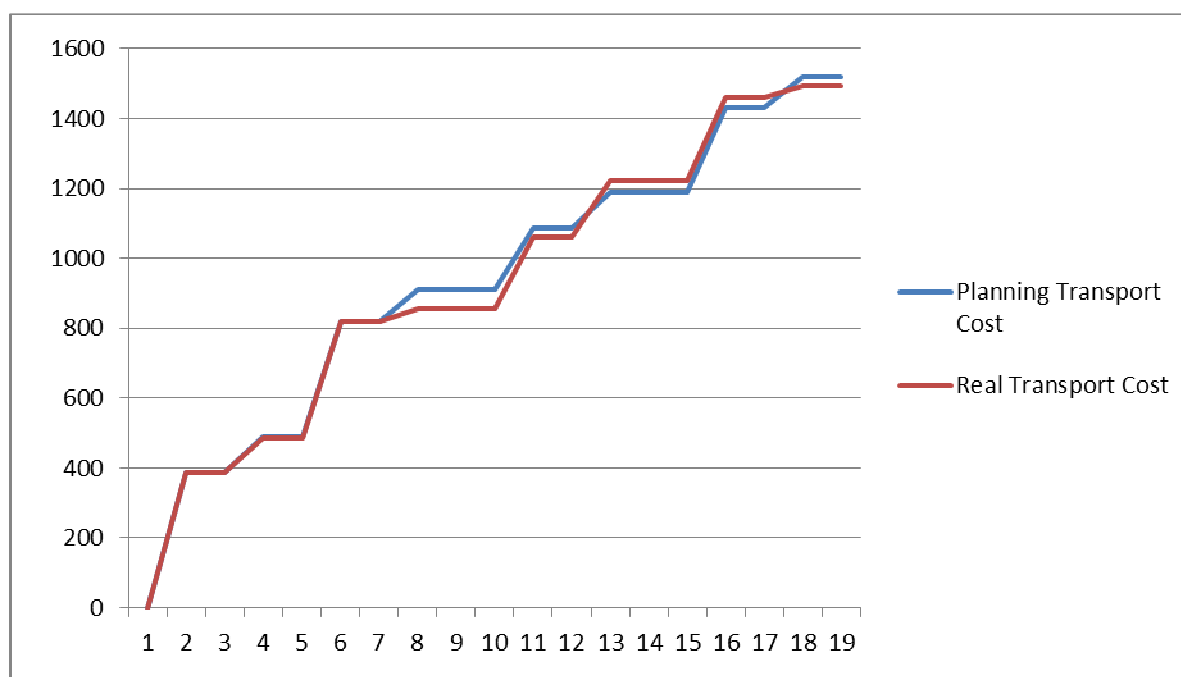


Figura 19: Gráfico do desenvolvimento dos custos de transporte real e planejado.

Para os testes com a metodologia clássica também foram colhidas informações de tempo entre ações e custos. A Tabela 9 apresenta o desenvolvimento da sequência de ações realizadas pelo sistema.

Tabela 9: Desenvolvimento das ações com a metodologia clássica.

Ação	Estado	S1	S2	S3	fc1	fc2	fc3	Cronômetro (min:seg)
	Estado inicial	1327	383	450	1	0	0	01:23
1: (Load v1 f1 level 8)								
	Estado 1	3807	383	450	1	0	0	01:34
2: (Move v1 f1 a2)								
	Estado 2	3807	383	450	0	1	0	03:21
3: (Unload v1 a2 level 8)								
	Estado 3	1237	383	3425	0	1	0	03:31
4: (Move v1 a2 f1)								
	Estado 4	1237	383	3425	1	0	0	05:00
5: (Load v1 f1 level 8)								
	Estado 5	3804	383	3425	1	0	0	05:11
6: (Move v1 f1 a1)								
	Estado 6	3804	383	3425	0	0	1	06:55
7: (Unload v1 a1 level 8)								
	Estado 7	1365	3450	3425	0	0	1	07:06
8: (Move v1 a1 f1)								
	Estado 8	1365	3450	3425	1	0	0	07:50
9: (CompleteOrder a1 c1 nivel 7)								
	Estado 9	1365	1368	3425	1	0	0	08:09
10: (PartialOrder a2 c2 nivel 8)								
	Estado 10	1365	1368	324	1	0	0	09:10
11: (Load v1 f1 level 8)								
	Estado 11	3804	1368	324	0	1	0	10:36
12: (Move v1 f1 a2)								
	Estado 12	3804	1368	324	0	1	0	10:49
13: (Unload v1 a2 level 2)								
	Estado 13	728	1368	3410	1	0	0	12:59
14: (Move v1 a2 f1)								
	Estado 14	728	1368	3410	1	0	0	13:57
15: (Final_order a2 c2 level 2)								
	Estado 15	728	1368	2070	1	0	0	14:20

Neste caso também foram contabilizados os custos reais da rotina e comparados com os custos ideais, definidos de acordo com a Eq.2 e com a Eq.3. A Tabela 10 descreve o desenvolvimento dos custos e a Figura 20 apresenta um gráfico para o desenvolvimento obtido pelo custo de transporte ideal (*Ideal Transport Cost*) e o pelo custo de transporte real (*Real Transport Cost*).

Tabela 10: Desenvolvimento dos custos ideal e real.

Ação	Ideal Transport Cost	Ideal Lost Cost	Real Transport Cost	Real Lost Cost
1: (Load v1 f1 level 8)	0	0	0	0
2: (Move v1 f1 a2)	330	0	331,5	0
3: (Unload v1 a2 level 8)	330	0	331,5	0
4: (Move v1 a2 f1)	390	0	403,5	0
5: (Load v1 f1 level 8)	390	0	403,5	0
6: (Move v1 f1 a1)	775	0	806	0
7: (Unload v1 a1 level 8)	775	0	806	0
8: (Move v1 a1 f1)	880	0	916,3	0
9: (CompleteOrder a1 c1 nivel 7)	880	0	916,3	0
10: (PartialOrder a2 c2 nivel 8)	880	105	916,3	0
11: (Load v1 f1 level 8)	880	105	916,3	0
12: (Move v1 f1 a2)	1210	105	1261,3	0
13: (Unload v1 a2 level 2)	1210	105	1261,3	0
14: (Move v1 a2 f1)	1240	105	1291,3	105,7
15: (Final_order a2 c2 level 2)	1240	105	1291,3	105,7

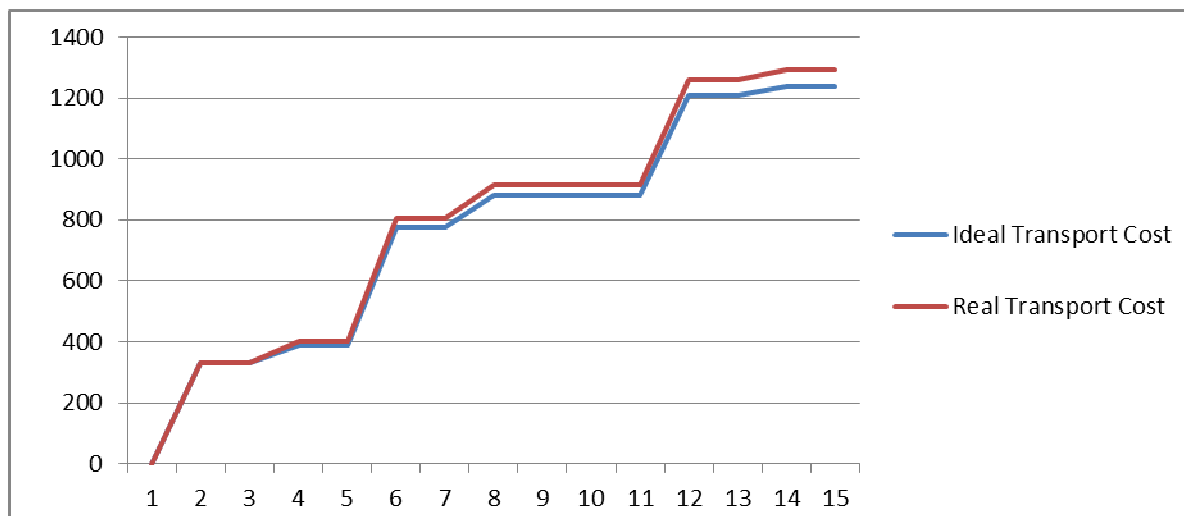


Figura 20: Gráfico do desenvolvimento dos custos de transporte real e ideal.

A principal diferença percebida entre as duas metodologias, foi o número de passos desenvolvidos para a solução do problema. Enquanto a solução utilizando o planejador automático desenvolveu 19 ações, a solução clássica desenvolveu apenas 15. Esta diferença deve-se exclusivamente ao fato de não ser possível levar a solução clássica ao mesmo estado final modelado com planejamento automático, como pode ser visto na Tabela 11. Isto levou a valores de custo de transporte muito distintos e sem possibilidade de comparação entre si. O que realmente foi analisado, neste caso, foi o desvio relativo entre o valor obtido e o desejado, para cada método.

Tabela 11: Comparação entre o Estado objetivo do modelo e após os testes

	Estado	Sensor de Nível do Veículo (%)	Sensor de Nível do Distribuidor 1 (%)	Sensor de Nível do Distribuidor 2 (%)	Posição do Veículo
Modelo	Estado objetivo		50	50	Fornecedor
Testes					
Metodologia Clássica	Estado 15	0	30,45	65,38	Fornecedor
Planejamento Automático	Estado 19	0	48,76	48,49	Fornecedor

Em ambos os casos apresentados neste trabalho o sistema conseguiu atender as demandas dos clientes. O método com planejamento automático desenvolveu 19 ações com custo de transporte de 1492 enquanto o planejado era de 1520. Já o método clássico utilizou 15 ações para resolver este problema, com custo de transporte de 1291,3 enquanto o planejado era de 1240. Percebe-se que ambos os métodos tiveram uma disparidade com o custo ideal, sendo que, para o sistema resolver o problema pelo método com planejamento automático o custo de transporte foi 1,84% menor enquanto que pelo método clássico o custo foi 4,14% maior. Já o custo com perdas de vendas, relacionado à ação de entrega parcial, foi 0,67% maior em ambos os casos. Esta análise aponta uma menor disparidade entre os custos real e planejado para o método com planejamento automático.

Outra diferença percebida foi no tempo total para execução da tarefa. Enquanto o método com planejamento automático gastou 13:46 (min:seg), o método clássico levou 14:20. Isto mostra que, apesar de ser composto por um número menor de ações, o método clássico levou mais tempo para solucionar o problema quando comparado com o método proposto. Este comportamento pode ser explicado pelo fato de, no método clássico, o veículo sempre realizar operações de carregar e descarregar o reservatório do veículo completamente, o que aumenta os tempos individuais das ações mover (*move*), carregar (*load*) e descarregar (*unload*).

Nota-se então que, através da metodologia com planejamento automático é possível: definir um estado objetivo para o sistema e, com isso, trabalhar com previsão de demanda, antecipando o estoque dos distribuidores para futuras entregas e variações de mercado; aplicar métricas de otimização de variáveis para minimização de custos; simular o comportamento do sistema real antes da implementação prática. Já através da metodologia clássica, o sistema fica travado nas informações de variação de estoque dos distribuidores e não prevê ou antecipa variações de mercado; não há métodos de otimização ou minimização de custos.

Além disso, enquanto a metodologia clássica, ao generalizar o método de controle com um modelo cíclico em Grafset, prevê apenas poucos cenários do sistema, o planejamento automático, justamente por sua característica de orientação a objetos, realiza uma varredura de todas as possibilidades possíveis e consegue prever todos os cenários possíveis do sistema. Assim, em casos de incompatibilidade entre o cenário planejado e o percebido, diferentemente da metodologia clássica, com o planejamento automático é possível redirecionar o sistema para a situação objetivo, solicitando-se um replanejamento ao planejador automático. O estudo do replanejamento encaixa-se nos trabalhos futuros deste projeto.

A análise destes resultados comprava que, neste caso abordado, o planejamento automático obteve resultados mais relevantes do que a metodologia clássica.

Contudo, existem situações mais complexas onde o tempo de processamento do plano passa a ser uma variável a se minimizar. De acordo com trabalhos recentes (HUANG et al, 2011), quando o tempo de processamento se apresenta como uma nova variável do modelo, uma forma de otimizar esta variável é através da paralelização do processo, através da distribuição de ações e processamentos em diferentes máquinas. A análise deste tipo de caso é outro trabalho futuro para este projeto.

Este trabalho ratifica a possibilidade de aplicação de planejadores automáticos em casos práticos e da sua utilização na automação da manufatura. Além disso, novos horizontes se abrem para a comunidade de automação no que diz respeito à possibilidade de se utilizar Inteligência Artificial para comandar totalmente um processo produtivo, considerando casos específicos e orientação a objetos e não apenas um caso genérico.

O melhor dos mundos para a aplicação de planejamento automático na automação da manufatura seria a integração entre as linguagens discrepantes, neste caso abordado, PDDL e *Ladder*. Esta integração seria possível através do desenvolvimento de um aplicativo capaz de realizar uma interface entre itSIMPLE e CLP. Assim, podemos citar outro trabalho futuro, que seria o desenvolvimento de um aplicativo para a integração entre o planejador automático alocado no itSIMPLE e o controlador do processo.

Outro estudo que pode ser desenvolvido é sobre a utilização de microcontroladores que substituiriam o Controlador Lógico Programável. Além disso, pretende-se avaliar esta metodologia também com sistemas para produtos do tipo discreto (como peças e componentes), integrando tecnologia de identificação por radio frequência, saindo do mundo contínuo como apresentado neste trabalho com transporte de líquidos.

REFERÊNCIAS

- CASTRUCCI, P., MORAES, C. C. de. (2001) **Engenharia de Automação Industrial**. 1.ed. São Paulo: Editora LTC, 2001
- DAHAL, K., S. GALLOWAY, G. BURT, J. MCDONALD and I. HOPKINS (2003). Port system simulation facility with an optimization capability. In: *International Journal of Computational Intelligence and Applications*, vol. 3, no. 4, pp. 395-410.
- FONSECA, J.P.S. **Desenvolvimento e Montagem de uma Bancada Didática de Planejamento Automático**. 2011. 165 f. Redação do Projeto de Fim de Curso, Universidade Federal de Uberlândia, Uberlândia.
- FONSECA, J.P.S., TAVARES, J.J.P.Z.S., (2011). Didatic Test Bench For Automated Planning. In *Proceedings of 21st Brazilian Congress of Mechanical Engineering*, Natal, Brazil
- FOX, M., and D. LONG (2003). PDDL2.1: An extension of pddl for expressing temporal planning domains. In: *Journal of Artificial Intelligence Research (JAIR)*, 20:61–124.
- HSU, C.W., WAH, B.W, 2008, Illinois, Usa. **The SGPlan Planning System in IPC-6**. Illinois, Usa: University Of Illinois, 2008. 3 p.
- HUANG, C.Y., CHEN, W.L., YEH, S.C. Supply Network Planning for Memory Module Industry by Distributed Parallel Computing. In *Proceedings of 21st International Conference on Production Research*. Stuttgart, Germany, 2011
- LI, J., L. WENKAI, I. A. KARIMI and R. SRINIVASAN (2005). Robust and Efficient Algorithm for Optimizing Crude Oil Operations, In: *American Institute of Chemical Engineers Annual Meeting*.
- MELLO, A.T.F., BARBOSA, M.C. **Transcritor de redes de Petri para linguagem de programação de máquina**. 2010. 95 f. Trabalho de Formatura, Universidade de São Paulo, São Paulo.
- MIYAGI, P. E. **Controle Programável: Fundamentos do controle de sistemas a eventos discretos**. 1. ed. São Paulo: Editora Blucher, 1996.
- MURATA, T. (1989). Petri Nets: Properties, Analysis and Applications. *IEEE Proceedings*, vol. 77, no.0 4, April.
- NGUYEN, A. (2003). Challenge ROADEF'2005: Car Sequencing Problem. Renault, France.
- OLIVEIRA, J.C.P. **Controlador Programável**. São Paulo. Makron Books, 1993.
- OMG – Object Management Group. **Unified Modeling Language specification**. Disponível em: <<http://www.omg.org/uml>>. Acesso em: 14 jul. 2010.
- ONROM Corporation (Japan) (Org.). **Programmable Controllers: Operation Manual**. Tokyo, 2001. CD-ROM.
- SETTE, F.M., VAQUERO, T.S., PARK, S.W., SILVA, J.R. (2008). Are Automated Planers up to Solve Real Problems?. *Proc. IFAC Conf.*, Seoul.
- TAVARES, J.J.P.Z.S., FONSECA, J.P.S. Supply Chain Didactic Testing Bench With Automated Planning Tool. In *Proceedings of 21st International Conference on Production Research*. Stuttgart, Germany, 2011.
- 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 *Proceedings of X Simpósio Brasileiro de Automação Inteligente*. São João Del Rey, Brazil, 2011.
- TAVARES, J.J.P.Z.S. **Elaboração de Projetos de Automação e Controle**. 2011. Apostila do curso de Automação Industrial para Engenharia Mecatrônica. Universidade Federal de Uberlândia, Uberlândia, Brasil.

VAQUERO, T.S., 2007. "ITSIMPLE: Ambiente integrado de análise de domínios de planeamento automático".2007. 316 f. Dissertação (Mestrado) - Universidade de São Paulo, São Paulo.

VAQUERO, T.S., SILVA, J.R., FERREIRA, M., TONIDANDEL, F., BECK, J.C., (2009). From Requirements and Analysis to PDDL in itSIMPLE3.0. *Proc. Int. Competition in Knowledge Engineering for Planning and Scheduling (ICKEPS 2009)*, 54-61.

VAQUERO, T.S., SILVA, J.R., BECK, J.C., (2010). Improving Planning Performance Through Post-Design Analysis. *Proc. of 19th Int. Conf. on Planning and Scheduling, Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*, 45-52.