

Valdeci Pereira Belo

Orientador: Antônio Otávio Fernandes

Sistema para Diagnóstico Automático de Falhas em Veículos Automotores OBD-2

Dissertação apresentada ao Curso de Pós-graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte

18 de Julho de 2003

Agradecimentos

Gostaria de agradecer, em primeiro lugar, a Flaviane e Ana Luiza, minha esposa e filha queridas, pelo apoio, carinho e compreensão a mim dedicados durante todo o processo de desenvolvimento deste trabalho.

Gostaria de agradecer aos meus pais, Luis e Aparecida (in memoriam), aos meus irmãos Valteir, Wanderley e Vanderli, às minhas irmãs, Selma e Sandra e a todos os meus familiares, pelo grande apoio e compreensão.

Gostaria de agradecer a todos os meus amigos da UFMG, pelo apoio técnico e afetivo, fornecidos durante o tempo de desenvolvimento deste trabalho. Gostaria de agradecer em especial ao meu amigo Marcelo Daride Gaspar, pelo grande incentivo e apoio que me forneceram a motivação necessária para a conclusão deste trabalho.

Gostaria de agradecer a todos os meus professores. Em especial, gostaria de agradecer ao professor Antônio Otávio Fernandes, pela sua orientação e apoio, e ao professor José Monteiro da Mata, pelas sábias palavras de incentivo, dadas no momento mais difícil deste trabalho.

Abstract

This work describes the application of model-based diagnosis to automated diagnosis of vehicles equipped with second generation on-board diagnostic systems (OBD-2).

The OBD-2 system is composed by hardware and software elements embedded into the vehicle. The purpose of this system is to assure proper emission control system operation for the vehicle's lifetime by monitoring emissions-related components and systems for deterioration and malfunction. When the OBD-2 system determines that a problem exists, it informs the driver that a problem has been detected and stores the corresponding diagnostic trouble code in the system memory. This information can be retrieved by OBD-2 diagnostic tools using the OBD-2 data network.

The OBD-2 system is very efficient to fault detection, but not very efficient to fault location. To eliminate such deficiency, we present a model-based diagnosis system, denominated Autodiag, whose diagnosis process incorporates the data provided by the OBD-2 system and allows the identification of faulty components. The Autodiag system is an example of an off-board diagnosis system that aims the repair or more specifically the replacement of the faulty components. Therefore, its task is to locate faults in the level of the smallest replaceable components. This system is composed of a qualitative model description language, an inference engine based on constraints propagation and an assumption-based truth maintenance system.

Resumo

Esta dissertação de mestrado trata dos fundamentos para a aplicação de técnicas de diagnóstico baseado em modelos para automatizar o processo de diagnóstico de falhas dos veículos automotores equipados com sistemas diagnósticos embarcados de segunda geração (OBD-2).

Um sistema OBD-2 é composto por elementos de *hardware* e *software* embarcados nos veículos automotores, cuja função é melhorar o desempenho e controlar o nível de emissão de poluentes. Este sistema monitora o estado de funcionamento do veículo e detecta falhas que possam aumentar a emissão de poluentes. Após a detecção de uma falha, este sistema notifica o motorista e armazena dados que permitem identificar a falha detectada. Estes dados são recuperados através de uma rede de comunicação de dados, usando-se uma ferramenta de diagnóstico OBD-2.

O sistema OBD-2 é bastante eficiente na detecção de falhas, mas pouco eficiente na localização das mesmas. Para suprir tal deficiência, apresentamos um sistema para diagnóstico de falhas baseado em modelos, denominado Autodiag, cujo processo de diagnóstico de falhas incorpora os dados providos pelo sistema OBD-2 e permite a identificação dos componentes defeituosos. O sistema Autodiag é um exemplo de sistema diagnóstico *off-board* que visa o reparo ou mais especificamente a troca dos componentes defeituosos. Dessa forma, a sua tarefa é localizar falhas no nível dos menores componentes substituíveis. Este sistema é formado por uma linguagem de descrição de modelos qualitativos, por um mecanismo de inferência baseado em propagação de restrições e por um sistema de manutenção da verdade baseado em suposições.

Sumário

1	Introdução	1
1.1	Sistemas diagnósticos inteligentes	3
1.2	Sistemas para diagnóstico de falhas em veículos automotores	5
1.3	Estrutura do documento	8
2	Sistema diagnóstico embarcado OBD-2	9
2.1	Introdução	9
2.2	Monitoramento de sistemas e detecção de falhas	11
2.2.1	Monitor da eficiência do catalisador	12
2.2.2	Monitor de falhas de ignição	13
2.2.3	Monitor do sistema de combustível	13
2.2.4	Monitor dos sensores de oxigênio aquecido	14
2.2.5	Monitor do sistema de controle de evaporação de combustível	14
2.3	Códigos de defeito diagnóstico	15
2.4	Rede de comunicação de dados diagnósticos	17
2.4.1	Serviços de comunicação	17
2.4.2	Serviços diagnósticos	19
2.5	Ferramenta de diagnóstico OBD-2	22
2.6	Processo de manutenção de veículos automotores OBD-2	23
3	Sistemas diagnósticos baseados em modelos	27
3.1	Introdução	27
3.2	Diagnóstico baseado em consistência	29
3.2.1	Formulação teórica do diagnóstico baseado em modelos	30
3.3	General diagnostic engine (GDE)	35
3.3.1	Predição de comportamento	37
3.3.2	Detecção de conflitos	39
3.3.3	Geração de diagnósticos candidatos	41
3.3.4	Discriminação de diagnósticos candidatos	42
3.4	Diagnóstico de falhas em veículos automotores: uma abordagem baseada em modelos	45
3.4.1	Domínio de aplicação	46
3.4.2	Classes de sistemas diagnósticos	47

3.4.3	Análise de requisitos	48
3.4.4	Processo diagnóstico	50
4	Sistema para diagnóstico automático de falhas em veículos automotores	
	OBD-2	52
4.1	introdução	52
4.2	Arquitetura do sistema	53
4.2.1	Modelos de estrutura e comportamento	54
4.2.2	Mecanismo de inferência e ATMS	61
4.3	Processo diagnóstico	62
4.3.1	Predição de comportamento	62
4.3.2	Detecção de conflitos	63
4.3.3	Geração de diagnósticos candidatos	64
4.3.4	Discriminação de diagnósticos candidatos	64
4.4	Processo de manutenção de veículos automotores	65
5	Implementação do sistema e resultados	67
5.1	Introdução	67
5.2	Implementação	68
5.2.1	Modelagem UML	70
5.2.2	Descrição da modelagem em XML	72
5.3	Avaliação do sistema	74
6	Conclusões e trabalhos futuros	81
A	Noções de lógica matemática	84
A.1	Cálculo proposicional	84
A.2	Lógica de primeira ordem	86
B	Modelo AMDL	89
B.1	Modelagem do sistema de injeção em AMDL	89
C	Definição dos modelos do Autodiag em XML	95
C.1	DTD para os modelos do Autodiag	95

Lista de Figuras

2.1	Arquitetura de rede OBD-2	18
2.2	Ferramenta de diagnóstico Scannycar SX 900	23
2.3	Processo de manutenção de veículos automotores OBD-2	25
3.1	Somador completo	31
3.2	Processo diagnóstico do GDE	36
3.3	Esquemático do sistema de propulsão de um veículo	46
4.1	Arquitetura do Autodiag	55
4.2	Circuito de um sensor de temperatura	58
4.3	Arquitetura do sistema diagnóstico	61
4.4	Inserção do Autodiag ao processo de manutenção	66
5.1	Arquitetura de implementação	68
5.2	Diagrama de pacotes do Autodiag	71
5.3	Diagrama de classes do ATMS	72
5.4	Diagrama de classes do mecanismo de inferência	73
5.5	Diagrama de pacotes da modelagem	74
5.6	Construção de modelos para o Autodiag	75
5.7	Esquemático do sistema modelado	76
5.8	Interface do Autodiag antes do carregamento de um modelo	78
5.9	Interface do Autodiag após o carregamento de um modelo	79
5.10	Interface do Autodiag apresentando resultados	80

Lista de Tabelas

2.1	Código de defeito - Primeiro caracter	16
2.2	Código de defeito - Segundo caracter	16
2.3	Código de defeito - Terceiro caracter	16
5.1	Pacotes e suas funções	71
A.1	Conectivos do cálculo proposicional	85

Capítulo 1

Introdução

Esta dissertação de mestrado trata dos fundamentos para a aplicação de técnicas de diagnóstico baseado em modelos para automatizar o processo de diagnóstico de falhas dos veículos automotores equipados com a tecnologia OBD-2. O seu objetivo é a definição e implementação de um sistema para diagnóstico automático de falhas baseado em modelos capaz de simplificar o processo de manutenção dos sistemas eletrônicos dos veículos automotores. O funcionamento deste sistema baseia-se no uso dos serviços diagnósticos oferecidos pelo sistema diagnóstico embarcado OBD-2, presente na unidade de controle eletrônico do motor, para gerar informações que comparadas com as informações produzidas por modelos da estrutura e do comportamento dos componentes reais do veículo, permitam a identificação e o diagnóstico de falhas.

De um modo geral, podemos definir o processo de diagnóstico de falhas como a tarefa de gerar um diagnóstico, ou seja, de decidir se existe ou não uma falha e então identificá-la, a partir de variáveis e/ou comportamentos para os quais existe conhecimento do que é esperado ou normal. A aplicação deste processo aos sistemas eletrônicos dos atuais veículos automotores gera uma série de desafios para a comunidade técnica responsável pela sua manutenção, uma vez que o avanço da tecnologia automobilística é muito rápido. Além disso, esses desafios tendem a crescer ainda mais, devido ao crescente aumento da complexidade global dos veículos.

Grande parte das operações necessárias para se manter o funcionamento de um automóvel otimizado são realizadas por sistemas controlados eletronicamente. Estes sistemas, tais como os sistemas de injeção eletrônica de combustível e de freios anti-blocantes, possui uma estrutura comum, formada por sensores, atuadores e um dispositivo programável, responsável pela execução da função de controle desejada. Além de funções de controle, os sistemas de controle modernos possuem diversas funções relacionadas ao diagnóstico de falhas que ocorrem nos seus próprios dispositivos e que comprometem a execução dos processos por eles gerenciados. O conjunto destas funções dedicadas ao diagnóstico de falhas compõe o sistema diagnóstico embarcado (*Onboard Diagnostic* ou OBD) do veículo e atualmente representa cerca de 50% de todo o software presente nos sistemas de controle. A importância dos sistemas diagnósticos embarcados tem aumentado significativamente devido às leis ambientais que regulamentam a quantidade de poluentes emitidos pelos veículos. Um dos exemplos mais notórios desse tipo de regulamentação ambiental foi estabelecido nos Estados Unidos pela *California Air Resources Board* (CARB), no ano de 1996. Estas leis definiram limites rígidos para a quantidade de emissão gerada pelos veículos e estabeleceram novos processos de monitoramento e diagnóstico de falhas que levaram ao surgimento de uma nova geração de sistemas diagnósticos embarcados, denominada *Onboard Diagnostic Generation 2* (OBD-2).

Apesar do diagnóstico de falhas realizado por estes sistemas (OBD ou OBD-2) ser eficaz em muitos casos, o seu objetivo não é identificar o componente defeituoso para que ele possa ser reparado. O objetivo deste sistema é verificar se o desempenho do veículo está de acordo com os padrões estabelecidos e, para tal, não é necessário detectar qual componente específico apresenta uma falha, mas apenas detectar se um determinado sistema apresenta falhas. Assim, o sistema diagnóstico embarcado monitora os sistemas do veículo, detecta as falhas e notifica o motorista sobre a necessidade de reparos. Como o sistema OBD não monitora cada componente do veículo, em muitos casos as informações que ele provê são insuficientes para o diagnóstico de falhas, principalmente de falhas mais complexas, como as falhas de múltiplos componentes ou a ocorrência de informações inconsistentes ou conflitantes [1].

Por outro lado, o domínio automobilístico apresenta sistemas cujos princípios de funcionamento são bem conhecidos, uma vez que foram projetados pelo homem. Tais domínios são adequados para a aplicação de técnicas de diagnóstico baseadas em modelos [2, 3, 4], nas quais um modelo representa explicitamente a estrutura do sistema, ou seja, seus componentes constituintes e sua organização. Nestes sistemas, um problema diagnóstico surge quando o comportamento observado do sistema difere do comportamento esperado, previsto pelo modelo, e a tarefa é identificar os componentes do sistema cujos defeitos podem explicar a anomalia. Assim, para facilitar o processo de manutenção dos veículos, apresentaremos um sistema capaz de coletar e interpretar as informações providas pelo sistema diagnóstico embarcado e confrontar essas informações com o conhecimento do funcionamento dos componentes eletrônicos do veículo, sugerindo testes e incorporando os seus resultados de modo a diagnosticar todas as falhas presentes em um determinado veículo.

1.1 Sistemas diagnósticos inteligentes

De maneira bastante geral, podemos definir o processo diagnóstico como uma análise cuidadosa de fatos, buscando o entendimento ou explicação de alguma coisa [5]. Este processo pode ser aplicado a uma grande variedade de domínios: diagnóstico de doenças, avaliação e autorização de crédito, avaliação financeira, depuração de software, identificação de falhas em circuitos eletrônicos, diagnóstico de falhas em equipamentos mecânicos e eletrônicos, etc. Apesar de existirem uma grande variedade de abordagens e tecnologias para a automatização do processo diagnóstico, todas elas tratam o mesmo problema de classificação de padrões, definido como a tarefa de atribuir uma dada entrada (um programa de computador, um conjunto de observações, etc.) a alguma categoria (correto, defeituoso, etc.).

Um sistema diagnóstico é um sistema capaz de identificar a natureza de um problema através da análise dos sintomas observados. O resultado produzido por tais sistemas é denominado diagnóstico e pode vir acompanhado ou não de uma explicação ou justificativa. Tipicamente, um sistema diagnóstico recebe como entrada um conjunto de sintomas (observações ou medidas) e identifica a provável causa que pode explicar tais sintomas. Esta

identificação é feita através da análise do conhecimento adequado do domínio, incorporado ao sistema por seus projetistas ou adquirido através de experiências.

O objetivo de um sistema diagnóstico é produzir resultados precisos de maneira eficiente. Existem várias abordagens distintas usadas para atingir tal meta. Estas abordagens podem ser distinguidas pela forma como identificam os relacionamentos entre os sintomas observados e seus respectivos diagnósticos e pela forma como representam e usam tais relacionamentos para a geração de diagnósticos. Considere um cenário onde existe um modelo do domínio, um modelo de um automóvel, por exemplo. Este modelo representa explicitamente o conhecimento relacionado ao sistema, ou seja, seus componentes constituintes e sua organização. Neste contexto, um problema diagnóstico surge quando o comportamento observado do sistema difere do comportamento esperado, provido pelo modelo, e a tarefa é identificar os componentes do sistema cujos defeitos podem explicar a anomalia. Este é o princípio usado pelos sistemas diagnósticos baseados em modelos [4].

Em muitos cenários práticos, como por exemplo para a determinação da resposta do corpo humano a agentes causadores de doenças, não existe um modelo preciso do domínio disponível. Nestes cenários, usualmente são usadas abordagens heurísticas. Uma abordagem heurística muito pesquisada nos anos 70 foi a dos sistemas especialistas [6], nos quais o conhecimento de um especialista humano é usado para a construção de um sistema baseado em regras de associação entre sintomas e suas respectivas causas. Nestes sistemas, a estrutura do sistema real sendo diagnosticado é apenas fracamente representada, quando existente. O sucesso dos diagnósticos gerados depende da experiência modelada do especialista e não do conhecimento do sistema sendo diagnosticado. Um exemplo notável desta abordagem é o sistema MYCIN [7], usado para o diagnóstico de doenças do sangue. Algumas dificuldades associadas com estes sistemas são a obtenção de informações relevantes dos especialistas e a manutenção da consistência da base de conhecimento, à medida que o sistema evolui.

Outra abordagem heurística importante, denominada raciocínio baseado em casos [8], usa um grande repositório de casos de problemas diagnósticos resolvidos para solucionar outros problemas. A tarefa destes sistemas é identificar um ou mais cenários (casos) que

explicam o problema diagnóstico corrente. Diferente dos sistemas diagnósticos baseados em modelos e dos sistemas especialistas, os sistemas baseados em casos não modelam nem o conhecimento do domínio nem o raciocínio diagnóstico de um especialista. O conhecimento diagnóstico é implicitamente representado no repositório de casos. Claramente, a performance destes sistemas depende criticamente do conjunto de casos do repositório, bem como dos procedimentos usados para procurar os casos que melhor se ajustam a um dado cenário.

As dificuldades associadas com a engenharia do conhecimento têm estimulado ainda muitas pesquisas na área de sistemas de aprendizagem (*learning systems*). Um sistema de aprendizagem é basicamente um sistema capaz de melhorar o seu desempenho em uma dada tarefa (ou conjunto de tarefas) através da experiência. Existem uma grande variedade destes sistemas, entre os quais podemos destacar as redes neuronais, os sistemas de classificação de padrões estatísticos e sintáticos, etc. Uma discussão detalhada destes sistemas, bem como dos sistemas baseados em modelos, especialistas e baseados em casos, pode ser encontrada em [5].

1.2 Sistemas para diagnóstico de falhas em veículos automotores

O rápido avanço da tecnologia automobilística, aliado à contínua pressão pela redução dos custos e diminuição do tempo de lançamento de novos produtos no mercado geram uma série de desafios para a comunidade técnica responsável pela manutenção dos modernos veículos automotores. Além disso, esses desafios tendem a crescer ainda mais devido ao crescente aumento da complexidade global dos veículos, cuja geração atual contém cerca de quatro a cinco vezes o número de módulos eletrônicos que sua geração antecedente [1]. Grande parte deste aumento de complexidade é devido à incorporação de componentes eletrônicos, usados para controlar o desempenho dos motores de forma a atender às severas leis ambientais, estabelecidas por países como Estados Unidos e países da comunidade

européia.

Atualmente, grande parte das operações necessárias para se manter o funcionamento de um automóvel otimizado são realizadas por sistemas controlados eletronicamente. Além das respectivas funções de controle, estes sistemas executam uma série de funções relacionadas à detecção de falhas que ocorrem nos dispositivos que os compõem e comprometem o desempenho dos processos por eles controlados. O conjunto das funções dedicadas ao diagnóstico de falhas forma o sistema diagnóstico embarcado (OBD) do veículo e representa cerca de 50% de todo o software presente nos sistemas de controle eletrônico dos veículos atuais [9]. A importância dos sistemas de diagnóstico embarcados aumentou significativamente devido às leis ambientais que regulamentaram a quantidade de poluentes emitidos. Dentre estas regulamentações ambientais, podemos destacar a estabelecida pela *California Air Resources Board* (CARB), em 1996, a qual definiu limites rígidos para a quantidade de poluentes emitida pelos veículos automotores e estabeleceu novos processos de monitoramento e diagnóstico de falhas que levaram ao surgimento da segunda geração de sistemas diagnósticos embarcados, denominada OBD-2. Posteriormente, estas leis foram estendidas para os demais estados norte americanos e incorporadas pelos países da comunidade européia [10].

Os sistemas diagnósticos embarcados oferecem uma visão melhorada e rica em informações dos complexos sistemas de controle eletrônicos dos veículos, ainda tratados como caixas-pretas por grande parte dos técnicos de manutenção. No caso de falhas mais simples, as informações providas por estes sistemas são suficientes para o seu diagnóstico. Entretanto, o objetivo principal destes sistemas não é identificar o componente defeituoso para que ele possa ser reparado. O seu objetivo é monitorar os sistemas do veículo, detectar falhas e notificar o motorista da necessidade de execução de reparos. Além disso, ele executa rotinas de recuperação que garantem o funcionamento do veículo até que ele possa ser levado a uma oficina para manutenção. Dessa forma, nos casos de falhas mais complexas, as informações que ele provê podem ser insuficientes para o diagnóstico das falhas. Uma pesquisa da Colorado State University, mostrou que em mais de 20% das vezes que o sistema OBD detecta uma falha, ele não provê informações suficientes para

identificar o componente defeituoso [1].

Dessa forma, podemos concluir que, apesar de o sistema diagnóstico embarcado facilitar a manutenção de um veículo automotor, os recursos providos por ele não são suficientes para resolver problemas complexos, os quais necessitam de ferramentas mais poderosas. Atualmente, na ocorrência destes problemas, o mecânico deve usar o seu conhecimento relativo ao sistema sendo diagnosticado para identificar os componentes defeituosos. Como existe uma grande variedade de veículos, cada um dos quais possuindo características particulares, a quantidade de informações necessárias para o diagnóstico de falhas é muito grande, gerando uma grande demanda por ferramentas que auxiliem o processo de identificação dos componentes defeituosos, ou seja, que automatizem o processo de diagnóstico de falhas.

Como vimos na Seção 1.1, existem diversas abordagens para a construção de sistemas dedicados ao diagnóstico automático de falhas. Entretanto, o domínio automobilístico apresenta características que dificultam a aplicação de determinadas abordagens e facilitam a aplicação de outras. Uma característica marcante deste domínio é a evolução muito rápida, que gera o lançamento de novas tecnologias e produtos a cada ano. A construção de um sistema especialista eficiente para este domínio requer a existência de um mecânico experiente, capaz de criar regras de associação entre os sintomas das falhas que ocorrem em um determinado veículo e suas respectivas causas. Independentemente da capacidade do especialista, esta abordagem permite apenas a criação de regras para falhas já ocorridas e para as quais o especialista já conhece as causas. Dessa forma, esta abordagem é ineficaz para o diagnóstico de novos tipos de falhas, que ocorrem em novos veículos. Da mesma forma, a construção de um sistema diagnóstico baseado em casos eficiente necessita de uma grande biblioteca de casos resolvidos, que também é inexistente para veículos novos.

Por outro lado, a maioria dos complexos sistemas de um veículo automotor podem ser decompostos em componentes, cujos comportamentos e conexões determinam o comportamento dos respectivos sistemas. Além disso, como estes componentes foram projetados pelo homem, suas estruturas e comportamentos são bem conhecidos. Isto torna o domínio automobilístico adequado para a aplicação de técnicas de diagnóstico de falhas baseadas

em modelos.

Dessa forma, este trabalho apresenta a definição de um sistema para diagnóstico automático de falhas baseado em modelos, denominado Autodiag, capaz de simplificar o processo de manutenção dos sistemas eletrônicos dos veículos automotores equipados com a tecnologia OBD-2. O seu funcionamento baseia-se na análise das informações oferecidas pelos serviços diagnósticos do sistema OBD-2 e na comparação destas informações com as informações produzidas por modelos da estrutura e do comportamento dos componentes reais de um veículo. O sistema Autodiag é um exemplo de sistema diagnóstico *off-board* que visa o reparo ou mais especificamente a troca dos componentes defeituosos. Dessa forma, a sua tarefa é localizar falhas no nível dos menores componentes substituíveis.

1.3 Estrutura do documento

O restante deste documento está organizado da seguinte maneira: o Capítulo 2 apresenta o sistema OBD-2, definindo as suas características e os seus serviços diagnósticos que serão usados em nossa abordagem. O capítulo 3 apresenta uma revisão bibliográfica sobre a abordagem de diagnóstico baseada em modelos, apresentando uma máquina de diagnóstico geral (GDE), cujo processo diagnóstico é a base da abordagem do Autodiag. O capítulo 4 define o sistema Autodiag, apresentando a sua arquitetura, o seu processo de diagnóstico e sua integração com o sistema OBD-2. O capítulo 5 apresenta a implementação de um protótipo do sistema, feita para permitir a sua avaliação. Esta avaliação é feita com o uso de um modelo simplificado de um sistema de injeção eletrônica de combustível. Para finalizar, o Capítulo 6 apresenta as conclusões deste trabalho e apresenta algumas diretrizes para a elaboração de trabalhos futuros.

Capítulo 2

Sistema diagnóstico embarcado

OBD-2

2.1 Introdução

Os sistemas diagnósticos embarcados foram projetados para manter o nível de emissão de poluentes dos veículos automotores dentro de padrões estabelecidos. Os sistemas de primeira geração realizavam esta tarefa de maneira relativamente eficiente, mas possuíam algumas deficiências graves. A principal deficiência relacionava-se ao fato de serem capazes de detectar falhas somente quando o componente deteriorava-se ao ponto de ficar inoperante, permitindo que o veículo operasse durante muito tempo em condições que geravam maior emissão de poluentes. Outra deficiência era a dificuldade de acesso às informações providas por estes sistemas para o diagnóstico de falhas, uma vez que cada fabricante disponibilizava um conjunto de serviços diagnósticos diferente, cada qual possuindo protocolos de acesso não padronizados. Geralmente o acesso a estes serviços era feito apenas por ferramentas comercializadas pelos próprios fabricantes do sistema diagnóstico embarcado.

O sistema diagnóstico embarcado OBD-2 (*On-board Diagnostic Generation 2*) constitui a segunda geração dos sistemas diagnósticos embarcados e foi definido pela *California Air Resources Board* (CARB), sendo incorporado a todos os veículos fabricados nos EUA a

partir de 1996. Ele possui como requisitos adicionais básicos a redução do tempo entre a ocorrência de uma falha e sua detecção e a definição de um conjunto de serviços diagnósticos, acessados de maneira padronizada por uma ferramenta de diagnóstico genérica, denominada ferramenta de diagnóstico OBD-2. O objetivo deste sistema é a detecção de falhas nos veículos, cujo desempenho anormal esteja a ponto de elevar a emissão de poluentes a valores acima dos limites estabelecidos por leis ambientais da agência de proteção ambiental americana (*Environment Protection Agency* ou EPA). Assim, logo que um veículo começa a exceder os padrões de emissão definidos, o sistema OBD-2 deve detectar uma falha e iluminar a lâmpada indicadora de avarias, avisando ao motorista a existência da falha. Assim, o motorista pode providenciar o reparo o mais cedo possível, diminuindo o tempo em que o veículo funcionará fora dos padrões estabelecidos. Além disso, o reparo do sistema será feito com o auxílio de um conjunto de serviços diagnósticos comuns a todos os veículos e implementados por uma ferramenta de diagnóstico geral, fabricada por diversos fabricantes independentes.

O sistema OBD-2 é definido por um conjunto de requisitos, cujo objetivo é controlar a quantidade de poluentes emitidos pelos veículos automotores [11]. Estes requisitos estabelecem limites máximos para as quantidades de poluentes (monóxido de carbono, hidrocarbonetos, óxidos de nitrogênio, etc.) expelidos pelos veículos e rigorosos processos de monitoração e detecção de falhas para todos os sistemas relacionados com a emissão. Estes requisitos incorporam uma série de normas técnicas [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23], estabelecidas pela *Society of Automotive Engineers* (SAE) e pela *International Organization for Standardization* (ISO), definindo um conjunto de serviços diagnósticos, uma rede de comunicação de dados e uma ferramenta de diagnóstico capaz de executar tais serviços.

As próximas seções apresentam uma breve descrição dos processos de monitoração e detecção de falhas do sistema OBD-2, da sua rede de comunicação de dados, com seus respectivos serviços diagnósticos, da ferramenta de diagnóstico OBD-2 e da sua inserção no processo de manutenção de veículos automotores OBD-2.

2.2 Monitoramento de sistemas e detecção de falhas

O sistema OBD-2 adiciona novos componentes de hardware e software aos sistemas de controle eletrônicos de um veículo, cuja função é monitorar todos os componentes relacionados com a emissão de poluentes e detectar a existência de falhas que possam fazer o veículo alcançar níveis de emissão 50% maiores que o padrão permitido. Todos os componentes relacionados ao funcionamento do motor, desde a formação da mistura ar-combustível a ser queimada até a descarga dos gases através do conversor catalítico, são monitorados. Durante a ocorrência de falhas em quaisquer dos sistemas monitorados, este sistema coleta e armazena uma série de dados em memória não volátil, incluindo um código de defeito diagnóstico que identifica a falha (Seção 2.3). Estes dados podem ser recuperados posteriormente pela ferramenta de diagnóstico OBD-2 (Seção 2.5), através da rede de comunicação de dados OBD-2, usando-se uma série de serviços e protocolos de comunicação padronizados (Seção 2.4).

As principal diferença de hardware, em relação aos sistemas embarcados de primeira geração, necessária para atender tais requisitos foi o uso de dois sensores de oxigênio aquecido, montados na entrada e na saída do conversor catalítico. Estes sensores permitem avaliar a composição da mistura ar-combustível, cuja relação deve ser mantida o mais próximo possível da relação estequiométrica definida para o respectivo combustível. Esta relação define a mistura ideal, na qual a queima de combustível é completa e a emissão de poluentes é mínima. Além disso, eles permitem avaliar o nível de emissão de poluentes, medindo o desempenho do sistema de controle de emissão do veículo.

Entretanto, o maior impacto dos requisitos do sistema OBD-2 foi sentido nas estratégias operacionais da unidade de controle eletrônico do motor, onde as novas informações devem ser medidas e analisadas, de tal modo que decisões sejam tomadas e ações adequadas realizadas. O software do sistema OBD-2 usa uma estratégia operacional, denominada monitores, para testar a operação de sistemas, componentes ou funções específicas do veículo. Estes monitores formam a coluna vertebral do sistema OBD-2, estabelecendo um controle muito rígido sobre os parâmetros operacionais dos componentes do sistema, principalmente

daqueles relacionados com a emissão de poluentes. Os monitores são implementados por rotinas de software que executam tarefas estratégicas, aplicando até três tipos de testes para determinar as condições do dispositivo controlado:

- Testes passivos: correspondem aos testes diagnósticos feitos de forma contínua durante o funcionamento normal do veículo.
- Testes ativos: Quando o monitor identifica um componente fora dos seus parâmetros, um teste dinâmico é realizado naquele componente. Este teste não afeta a dirigibilidade do veículo e, para cada teste, executa-se uma ação específica para a qual um resultado é esperado.
- Testes intrusivos: Se após um teste passivo ou ativo ser executado, o resultado esperado não ocorrer, executa-se um teste intrusivo que constitui a fase final do procedimento diagnóstico. Esta fase definitivamente afeta a dirigibilidade e os padrões de emissão do veículo e o motorista notará que algo anormal está acontecendo. Porém, essa ação momentaneamente prejudicial para o sistema é necessária para a determinação conclusiva das condições do veículo, antes de acender a lâmpada indicadora de avarias. Se o resultado do teste é diferente do esperado, um código de defeito, representando a falha detectada, é armazenado e a lâmpada indicadora de avarias é iluminada.

A maior parte dos sistemas monitorados participa do controle da emissão de poluentes e são apresentados nas seções seguintes.

2.2.1 Monitor da eficiência do catalisador

O catalisador, conhecido também como conversor catalítico trivalente, é assim chamado porque reduz em cerca de 70% os três principais poluentes produzidos pelos motores à álcool e gasolina: monóxido de carbono, hidrocarbonetos e óxidos de nitrogênio. O seu perfeito funcionamento é essencial para que se atinja os padrões de emissão regulamentares.

Assim, o monitor de eficiência do catalisador avalia a sua eficiência frequentemente, usando as medidas feitas pelos dois sensores de oxigênio aquecidos, localizados na entrada e na saída do catalisador.

A estratégia do monitor baseia-se na comparação do nível de O_2 que entra no conversor catalítico, com o nível de O_2 que sai através do escapamento do veículo. O funcionamento baseia-se no fato de que a maioria das moléculas de O_2 deve ser gasta durante os processos de oxidação efetuados pelo conversor. Dessa forma, se são medidos valores mais altos do que aqueles estabelecidos para o veículo, pelos regulamentos ambientais, realiza-se os procedimentos relacionados à detecção de falha, ou seja, armazenamento de informações para diagnóstico da falha (código de defeito e etc.) e iluminação da lâmpada indicadora de avarias.

2.2.2 Monitor de falhas de ignição

Uma falha de ignição consiste na falta de combustível em um cilindro do motor, provocada por desgaste nos componentes da ignição, por uma mistura de combustível muito pobre (muito ar e pouco combustível) ou por falhas no sistema elétrico. As falhas de ignição, mesmo em pequeno número, provocam um grande aumento na emissão de poluentes pelo veículo. Assim, o monitor de falhas de ignição verifica as flutuações de velocidade do eixo da manivela do motor e determina se uma falha de ignição acontece devido a essas variações. A estratégia utilizada é bastante refinada e pode determinar até a severidade do erro.

2.2.3 Monitor do sistema de combustível

A estratégia de controle do sistema de injeção eletrônica de combustível é prover a relação de ar-combustível correta sob todas as condições de operação do veículo, exceto durante partidas a frio. Os parâmetros e componentes envolvidos neste controle são: sistema de medida de combustível, bomba de combustível, temporização da ignição, injetores de combustível, largura do pulso de injeção e controle da mistura estequiométrica (denominada

lambda). Basicamente, determina-se a largura do pulso de injeção necessário para manter a mistura ar-combustível dentro de uma janela de valores relativos ao valor lambda (0,93 a 1,07). O sistema de controle adiciona fatores de correção à largura do pulso para aumentar a injeção de combustível durante partidas a frio e aceleração total. Durante a desaceleração, o sistema de controle interrompe o fornecimento de combustível.

A monitoração do sistema de combustível é feita através do armazenamento das quantidades de combustível ótimas para cada carga e rotação do motor. O sensor de oxigênio na entrada do catalisador permite avaliar a qualidade da mistura e, desta forma, permite a execução de um controle em malha fechada para calcular as correções necessárias a serem aplicadas aos valores armazenados. Uma correção que excede um limite pré-determinado indica que o sistema de combustível apresenta falhas que podem elevar significativamente a emissão de poluentes. Em casos de detecção de falhas neste sistema, o sistema OBD-2 armazena um código de defeito e ilumina a lâmpada indicadora de avarias.

2.2.4 Monitor dos sensores de oxigênio aquecido

Os sensores de oxigênio são extremamente importantes para o controle da emissão de poluentes do veículo, sendo usados tanto no controle da mistura ar-combustível, quanto na monitoração da eficiência do catalisador. Desta forma, o monitor dos sensores de oxigênio executa as seguintes verificações: teste de continuidade de circuito, teste dos aquecedores (uma vez que os sensores operam melhor em altas temperaturas, acima de 260 graus Celsius, possuindo um sistema de aquecimento) e teste das tensões de operação com o motor parado e funcionando. Se quaisquer destes testes falharem, o sistema OBD-2 armazena um código de defeito e ilumina a lâmpada indicadora de avarias.

2.2.5 Monitor do sistema de controle de evaporação de combustível

Apesar de ser uma das principais fontes de emissão de hidrocarbonetos, a evaporação de combustível do tanque de combustível foi negligenciada durante muito tempo. Existem

duas causas principais para os vapores de combustível no tanque: o aumento da temperatura ambiente e o retorno de combustível não usado do motor. O sistema de controle de evaporação de combustível consiste de uma linha de ventilação que sai do tanque de combustível e entra em um elemento de carvão (canister) que absorve o vapor de combustível, permitindo somente a passagem de ar para a atmosfera.

Desta forma, o monitor do sistema de controle de evaporação de combustível é responsável pela verificação da integridade do fechamento hermético deste sistema. Este monitor tem a habilidade de descobrir vazamentos em qualquer ponto do sistema, graças ao uso de um sensor de pressão modificado situado no tanque de combustível. Se uma falha for detectada, o sistema OBD-2 armazena um código de defeito e ilumina a lâmpada indicadora de avarias.

2.3 Códigos de defeito diagnóstico

A estratégia usada por um sistema OBD-2 para detectar e diagnosticar falhas consiste na execução de testes em todos os sensores e atuadores que o compõem. Se uma falha é detectada por qualquer destes testes, o código de defeito diagnóstico correspondente é armazenado e a luz indicadora de avarias é iluminada. Entretanto, o componente ou sistema indicado no código de defeito não necessariamente é a causa da falha, ou seja, o objetivo do código de defeito é apenas isolar a falha a uma área funcional específica do veículo.

O uso de códigos de defeito diagnóstico comuns facilita a vida dos mecânicos, uma vez que eles podem ter acesso aos códigos de todos os fabricantes da mesma maneira. A norma SAE J2012 [14] estabelece um sistema de código alfanumérico com 5 caracteres, cada um dos quais representando um valor específico. O primeiro dígito do código de defeito, cujos possíveis valores são mostrados na Tabela 2.1, indica a localização da falha no veículo.

O segundo caracter do código indica a entidade responsável pela sua definição. O possíveis valores para este caracter são mostrados na Tabela 2.2. Este caracter é muito importante, pois indica se o código em questão é comum a todos os fabricantes (valor igual

Valor	Significado
P	Motor (<i>Powertrain</i>)
B	Corpo (<i>Body</i>)
C	Chassi (<i>chassis</i>)

Tabela 2.1: Código de defeito - Primeiro caracter

a 0) ou se é específico do fabricante.

Valor	Significado
0	SAE
1	Fabricante do veículo ou central
2 a 9	Reservados para uso futuro

Tabela 2.2: Código de defeito - Segundo caracter

O terceiro caracter do código de defeito, cujos valores são mostrados na Tabela 2.3, refere-se a um subgrupo de funções do veículo. O quarto e quinto caracteres relacionam-se a uma falha específica no subgrupo previamente indicado.

Valor	Significado
0	Sistema eletrônico completo
1	Sistema de controle de combustível e ar
2	Sistema de controle de combustível e ar
3	Sistema de ignição
4	Controles de emissão auxiliares
5	Sistema de controle de velocidade
6	Central eletrônica e entrada e saída
7	Transmissão
8	Motor (parte não eletrônica)

Tabela 2.3: Código de defeito - Terceiro caracter

A norma SAE J2012 define códigos para todos os circuitos, componentes e sistemas controlados pela SAE, ou seja para todos os códigos iniciados por “P0”. Assim, por exemplo, o código de defeito P0308 indica um problema de motor, definido pela SAE, relacionado a uma falha no sistema de ignição do cilindro número 8.

2.4 Rede de comunicação de dados diagnósticos

A definição do sistema OBD-2 apresenta uma arquitetura de rede baseada no modelo de referência OSI [24], contendo as camadas física, enlace de dados e aplicação. A camada de aplicação é responsável pelos serviços diagnósticos disponibilizados para o usuário, sendo construída sob a infra-estrutura de serviços de comunicação das camadas inferiores. A camada de enlace de dados, por sua vez, utiliza os serviços da camada física e provê serviços relacionados ao reconhecimento das mensagens e detecção de erros durante a comunicação. A camada física é a camada inferior do modelo e trata da transmissão das mensagens através de um meio físico.

Como mostra a Figura 2.1, a camada de aplicação possui um único protocolo, definido pela norma SAE J1979. Entretanto, para garantir compatibilidade com os protocolos usados nos sistemas diagnósticos de primeira geração, a camada física e a camada de enlace de dados podem possuir diferentes protocolos: ISO 9141-2, ISO 14230, SAE J1850 PWM ou SAE J1850 VPW. Durante a implementação do sistema OBD-2, o fabricante da unidade de controle eletrônica escolhe e implementa um único protocolo para cada camada. Entretanto, um dos principais requisitos da ferramenta de diagnóstico OBD-2 (Seção 2.5) é a capacidade de interação com todos os sistemas OBD-2. Assim, esta ferramenta implementa todos os protocolos referentes a cada camada e possui um mecanismo para detectar o protocolo usado em um sistema diagnóstico específico. A seguir, apresentamos os serviços providos pela arquitetura de rede OBD-2, com seus respectivos protocolos de comunicação.

2.4.1 Serviços de comunicação

Os serviços de comunicação, providos pelas camadas física e de enlace de dados, estabelecem a infra-estrutura de comunicação sob a qual são implementados os serviços diagnósticos. Eles são responsáveis pelo reconhecimento de mensagens, pela detecção de erros e pela transmissão dos dados através do meio físico. Estes serviços são implementados através de uma série de protocolos, apresentados nas subseções seguintes.

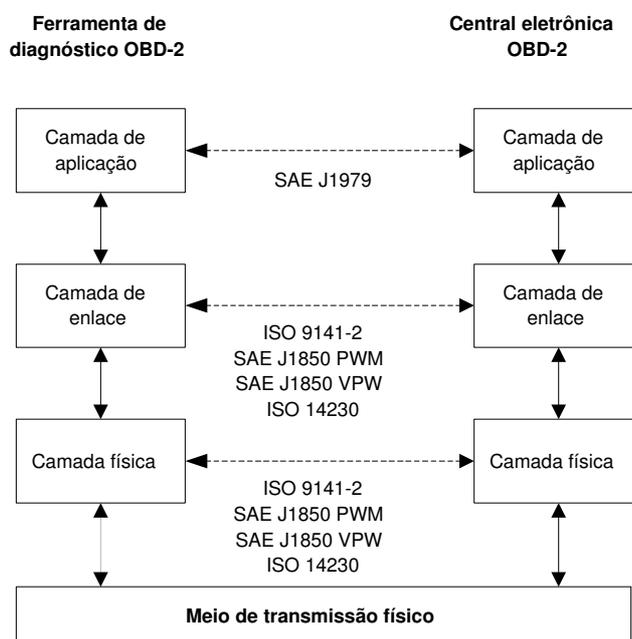


Figura 2.1: Arquitetura de rede OBD-2

2.4.1.1 Protocolo ISO 9141-2

O protocolo ISO 9141-2 [23] é uma especialização do protocolo ISO 9141 [22], definida especialmente para atender os requisitos do sistema OBD-2. Este protocolo estabelece a comunicação através de um barramento serial, composto por duas linhas de comunicação (denominadas K e L), ligadas ao conector diagnóstico SAE J1962 [16]. Além disso, ele define o formato das mensagens e procedimentos para detecção de erros (checksum) e inicialização da comunicação.

2.4.1.2 Protocolo ISO 14230

O conjunto de normas ISO 14230, ou *Key Word Protocol 2000*, foi estabelecido pela ISO com o objetivo de definir requisitos comuns para os sistemas diagnósticos implementados com o uso de uma linha de comunicação serial. Ele é baseado no modelo de referência OSI, apresentando tanto serviços diagnósticos quanto de comunicação. Os serviços diagnósticos compõem a camada de aplicação e permitem o acesso aos recursos do sistema

diagnóstico embarcado. Os serviços de comunicação fornecem a infra-estrutura necessária para a comunicação com a unidade de controle eletrônica.

Os protocolos ISO 14230 são definidos em cinco documentos, tratando respectivamente da interface dos serviços [17], da camada física [18], da camada de enlace de dados [19], da camada de aplicação [20] e dos requisitos para o seu uso com o padrão OBD-2 [21]. Apenas os protocolos das camadas física e de enlace de dados são usados no sistema OBD-2.

2.4.1.3 Protocolo SAE J1850

O protocolo SAE J1850 [15] estabelece uma rede de comunicação de dados usada para o compartilhamento de informações entre todos os módulos eletrônicos de um veículo. Os sinais digitais são comunicados utilizando-se o conceito de multiplexação por divisão de tempo, possuindo duas abordagens distintas: SAE J1850 VPW, com modulação por largura de pulso variável, e SAE J1850 PWM, com modulação por largura de pulso fixa. Na modulação por largura de pulso variável, o valor de um símbolo é determinado pela largura do pulso e pelo estado do barramento, enquanto na modulação por largura de pulso fixa o seu valor é determinado apenas pela largura ou duração do pulso. Independente da abordagem, o protocolo define o formato das mensagens e procedimentos para detecção de erros e inicialização da comunicação.

2.4.2 Serviços diagnósticos

Os serviços diagnósticos são definidos e divididos em modos de teste pelo protocolo de aplicação SAE J1979 [13]. Esses serviços são obrigatoriamente implementados por todos os sistemas OBD-2 e são acessíveis através da rede de comunicação de dados OBD-2. Os modos de teste, com seus respectivos serviços, são apresentados abaixo.

2.4.2.1 Modo de teste 1 (Obtenção de dados diagnósticos atuais)

O sistema OBD-2 gerencia e disponibiliza uma série de dados referentes ao estado atual de funcionamento de um veículo. Esses dados compreendem os valores correntes das medidas

dos sensores e de parâmetros de qualidade relacionados com a emissão de poluentes. Os dados recuperados através desse modo de teste correspondem sempre aos últimos valores determinados pelo sistema OBD-2.

2.4.2.2 Modo de teste 2 (Obtenção de dados diagnósticos armazenados)

Quando o sistema OBD-2 detecta uma falha em qualquer componente que pode provocar o aumento da quantidade de poluentes emitidos pelo veículo, ele armazena um código de defeito. Em alguns casos, além do código de defeito, ele armazena todo o estado do veículo, incluindo os valores das medidas dos sensores, no momento em que a falha foi detectada. Com esse procedimento, ele permite que o mecânico saiba qual foi a falha ocorrida e as circunstâncias na quais ela ocorreu. Assim, os serviços definidos pelo modo de teste 2, permitem obter os valores dos dados armazenados durante a ocorrência de tais falhas.

2.4.2.3 Modo de teste 3 (Obtenção dos códigos de defeitos diagnósticos)

Como foi visto anteriormente, o sistema OBD-2 verifica constantemente o funcionamento de diversos componentes do veículo e sempre que uma falha é identificada, ele armazena um código de defeito diagnóstico e indica a presença de tal irregularidade, acendendo a luz indicadora de avarias no painel frontal do veículo. Esses códigos de defeito são recuperados através dos serviços definidos pelo modo de teste 3.

2.4.2.4 Modo de teste 4 (Remoção de informações de diagnóstico armazenadas)

O modo de teste 4 provê serviços que permitem comandar o sistema OBD-2 de um veículo a remover toda a informação de diagnóstico armazenada e relacionada com a emissão de poluentes. Essa classe de serviços é especialmente importante durante o processo de teste do veículo, realizado após a sua manutenção, para verificar se o reparo foi realizado com sucesso.

2.4.2.5 Modo de teste 5 (Testes de monitoramento dos sensores O_2)

Um dos componentes estratégicos para o controle do nível de poluentes emitidos pelos veículos é o sensor de oxigênio. Através da medida da quantidade de oxigênio presente no meio ambiente e nos gases do escapamento dos veículos, esse sensor permite que seja possível determinar se o nível de emissão de poluentes do veículo está ou não dentro dos limites definidos pelas leis de proteção ambiental. Dessa forma, o modo de teste 5 possui um conjunto de serviços dedicado exclusivamente a esse componente, permitindo a recuperação de diversas informações que auxiliam o diagnóstico de defeitos nos mesmos.

2.4.2.6 Modo de teste 6 (Testes dos sistemas não continuamente monitorados)

Como vimos na Seção 2.2, o núcleo de um sistema OBD-2 é composto por rotinas executadas para testar a operação de um sistema, componente ou função específica. Essas funções são responsáveis por um rígido controle sobre os parâmetros operacionais dos componentes do veículo, principalmente daqueles relacionados com a emissão de poluentes. Existem sistemas no veículo que são constantemente monitorados e outros que são monitorados apenas sob condições especiais, tais como após a detecção de falhas. O modo de teste 6 fornece serviços que permitem o acesso aos resultados dos testes de monitoramento realizados em sistemas que não são continuamente monitorados.

2.4.2.7 Modo de teste 7 (Testes dos sistemas continuamente monitorados)

O modo de teste 7 é complementar ao modo de teste 6, provendo serviços que permitem acesso aos resultados dos testes de monitoramento diagnóstico realizados continuamente nos componentes ou sistemas relacionados com a emissão de poluentes, durante as condições normais de direção do veículo.

2.4.2.8 Modo de teste 8 (controle de componente, teste ou sistema embarcado)

Um sistema OBD-2 pode ser composto por diversos módulos ou sistemas. O modo de teste 8 oferece serviços que permitem controlar a operação de um sistema, teste ou componente

específico do veículo.

2.4.2.9 Modo de teste 9 (Informações do veículo)

O sistema OBD-2 armazena diversas informações relativas ao veículo no qual ele está instalado. O modo de teste 9 oferece serviços que permitem recuperar tais informações.

2.5 Ferramenta de diagnóstico OBD-2

A definição do sistema OBD-2, através da norma SAE J1978 [12], inclui uma série de requisitos que devem ser suportados pelos equipamentos que interagem com o sistema OBD-2 de um veículo, acessando os seus serviços diagnósticos. Esses requisitos aplicam-se a qualquer equipamento de teste, tais como *scanners* manuais, ferramentas baseadas em computadores pessoais e etc. Estes equipamentos são genericamente denominados de ferramentas de diagnóstico OBD-2 e possuem como objetivo básico permitir o acesso aos serviços diagnósticos providos pelo sistema OBD-2. Os principais requisitos desta ferramenta são:

- Implementação de todos os protocolos das camadas física e de enlace de dados, definidos na arquitetura de rede OBD-2, e de um procedimento para determinação automática da interface de comunicação usada no veículo;
- Obtenção e visualização dos parâmetros OBD-2 atuais (modo de teste 1);
- Obtenção e visualização dos parâmetros OBD-2 armazenados (modo de teste 2);
- Obtenção e visualização dos códigos de defeito diagnóstico (modo de teste 3);
- Remoção dos códigos de defeitos, parâmetros e resultados dos testes de avaliação executados pelo sistema OBD-2 (modo de teste 4);
- Obtenção e visualização dos resultados dos testes de monitoração executados pelo sistema OBD-2 (modos de testes 5, 6 e 7);



Figura 2.2: Ferramenta de diagnóstico Scannycar SX 900

- Disponibilização de um manual do usuário ou sistema de ajuda.

Como pode ser deduzido dos requisitos acima, a ferramenta de diagnóstico OBD-2 é basicamente um dispositivo que permite a leitura dos dados diagnósticos providos pelo sistema OBD-2 e a execução de funções de controle (Modo de teste 8), cujas funcionalidades são dependentes dos fabricantes.

A Figura 2.2 mostra um exemplo de ferramenta de diagnóstico OBD-2, implementada em conjunto pelo Departamento de Ciência da Computação da UFMG e pela Scanitec Equipamentos Automotivos Ltda [25]. A Seção 2.6 apresenta um processo para manutenção de veículos automotores, definindo como esta ferramenta é usada por um técnico de manutenção.

2.6 Processo de manutenção de veículos automotores OBD-2

O processo de manutenção de veículos automotores foi profundamente alterado pela crescente incorporação de sistemas eletrônicos. Apesar da parte mecânica dos veículos permanecer praticamente inalterada, a parte elétrica mudou significativamente, tornando a manutenção dos sistemas eletrônicos uma importante tarefa a ser realizada pelos técnicos

responsáveis pela manutenção dos modernos veículos automotores. De maneira geral, podemos estabelecer um método para diagnóstico e reparo de falhas composto por seis etapas [26]:

1. Coleta de evidências;
2. Análise de evidências;
3. Localização da falha;
4. Localização e solução da causa da falha;
5. Reparação do dispositivo defeituoso;
6. Teste do sistema e verificação do reparo realizado.

A coleta de evidências consiste na busca de informações relacionadas com a ocorrência de uma determinada falha do veículo. Para realizar esta coleta, é necessário conhecer quais componentes do veículo fazem parte do sistema defeituoso. Quanto maior o conhecimento a respeito de um sistema, maior é o número de informações disponíveis para a realização do diagnóstico.

Após a coleta de evidências, realiza-se uma análise crítica das informações coletadas, de forma dependente e adequada ao sistema sendo investigado, buscando uma maior compreensão do problema apresentado pelo veículo. Esta etapa é baseada no conhecimento do funcionamento dos componentes do sistema defeituoso e é indispensável para a determinação da causa da falha e do seu correto diagnóstico.

Os procedimentos de localização de falhas e de suas causas e de reparação do dispositivo defeituoso buscam identificar e reparar os componentes, cujo funcionamento incorreto explica os sintomas observados. O reparo de sistemas eletrônicos freqüentemente corresponde à substituição dos componentes defeituosos.

Para finalizar o processo de manutenção, os procedimentos de teste do veículo, para a verificação da eficácia do reparo, são realizados. Estes procedimentos constituem um importante aspecto do trabalho do técnico de manutenção, especialmente quando sistemas controlados eletronicamente são utilizados. Em caso de falhas intermitentes, por exemplo,

tais testes devem ser estendidos, uma vez que a falha pode ocorrer apenas em determinadas condições.

A execução deste processo de manutenção depende dos serviços providos pelo sistema diagnóstico embarcado. Em geral, a coleta de evidências, consiste na recuperação dos códigos de defeito e na leitura dos parâmetros atuais e armazenados, relacionados aos códigos recuperados. A localização das falhas é baseada nos códigos de defeito, uma que eles indicam uma área na qual o defeito está localizado, por exemplo “falha no circuito do sensor de temperatura da água”. Entretanto, isso não significa que o sensor esteja com defeito. A localização exata do defeito só é conseguida mediante o conhecimento dos componentes do sistema em questão, uma vez que o defeito pode estar em qualquer componente localizado na área apontada.

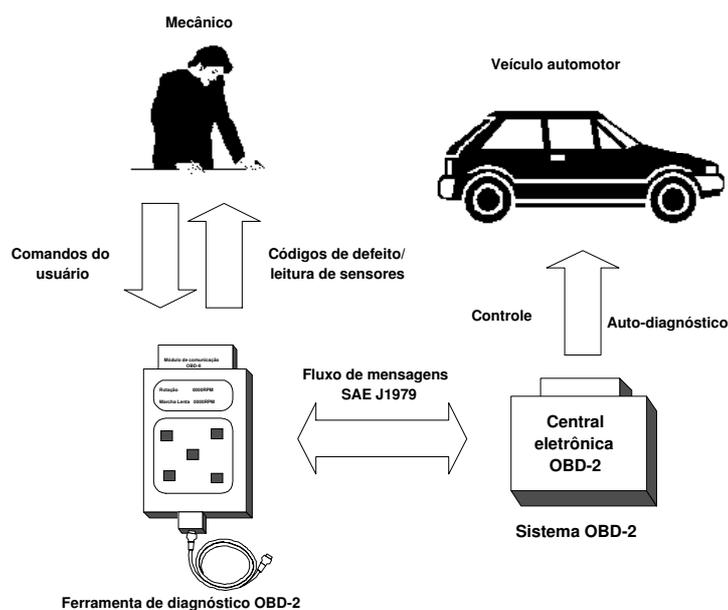


Figura 2.3: Processo de manutenção de veículos automotores OBD-2

Dessa forma, o uso da ferramenta de diagnóstico OBD-2 facilita o processo de manutenção dos veículos automotores equipados com esta tecnologia. Ela oferece uma interface padrão entre o sistema OBD-2 e o técnico responsável pela manutenção do veículo. Como mostra a Figura 2.3, a ferramenta recebe comandos do usuário e transforma tais comandos

em mensagens que são enviadas ao sistema OBD-2, requisitando os dados necessários. Ao receber a resposta da requisição, ela formata os dados retornados de tal forma que eles sejam facilmente compreendidos.

Assim, a ferramenta de diagnóstico permite que os serviços e informações disponibilizados pelo sistema OBD-2 sejam acessados pelo técnico de manutenção de uma forma bastante simples e intuitiva, facilitando o processo de manutenção. Entretanto, grande parte do processo de diagnóstico de falhas ainda depende do conhecimento técnico do mecânico responsável pela manutenção dos veículos, gerando uma grande demanda por ferramentas e sistemas que se dediquem ao diagnóstico das falhas apresentadas por tais veículos.

Capítulo 3

Sistemas diagnósticos baseados em modelos

Se o modelo é correto, então todas as discrepâncias entre observação e predição são descobertas e podem ser ligadas por caminhos causais aos defeitos no dispositivo modelado.

— Randall Davis e Walter Hamscher —

3.1 Introdução

Nos últimos anos, o desenvolvimento da área de diagnóstico automático baseado em modelos permitiu o estabelecimento de uma forte base teórica e a criação de uma tecnologia suficientemente madura para o desenvolvimento de aplicações industriais [27]. Este desenvolvimento foi possível após uma avaliação crítica da natureza e limitações dos sistemas diagnósticos especialistas, que predominaram nos anos 70, principalmente na área de diagnóstico médico. Estes sistemas capturavam as habilidades de um especialista através de um conjunto de associações, mais ou menos diretas, entre os sintomas e suas respectivas causas (por exemplo, “Se sintoma S então doença D com probabilidade P”). Como estes sistemas eram fundamentados na experiência obtida em casos passados, o processo

diagnóstico era tratado como uma coleta empírica de evidências de certas falhas, ao invés de um processo estritamente dedutivo.

Os sistemas especialistas mostraram-se muito restritivos para atender os requisitos para o diagnóstico de falhas em dispositivos técnicos. A aplicação industrial de técnicas de diagnóstico automático de falhas requer a detecção e localização de novas formas de falhas, exibidas por sistemas recém projetados e a interpretação de sintomas nunca vistos. É bastante claro que as capacidades humanas para satisfazer tais requisitos não se originam simplesmente da derivação de associações empíricas, relacionando os sintomas com suas causas. Elas originam-se do conhecimento dos princípios físicos e tecnológicos que justificam o funcionamento (correto ou defeituoso) de um artefato. Tal conhecimento permite a dedução sistemática de falhas hipotéticas a partir das observações disponíveis, mesmo que o artefato seja novo.

A idéia chave da abordagem de diagnóstico baseada em modelos é a representação explícita do conhecimento dos princípios físicos e tecnológicos de um artefato, através de um modelo da sua estrutura e do comportamento dos seus constituintes, e a organização do processo diagnóstico como um processo de inferência baseado neste modelo e no comportamento observado. Esta abordagem provê um rigoroso fundamento teórico para o processo diagnóstico, incluindo uma caracterização formal do seu objetivo e das inferências que atingem tal objetivo, dadas as predições do modelo e as observações do comportamento real do artefato.

Na seção seguinte, apresentamos os fundamentos teóricos da abordagem baseada em consistência para o diagnóstico automático de falhas. A Seção 3.3 apresenta um sistema que utiliza esta abordagem para diagnosticar falhas, usando apenas modelos de funcionamento correto de artefatos. Para finalizar, a Seção 3.4 apresenta um exemplo de aplicação desta abordagem no domínio automobilístico.

3.2 Diagnóstico baseado em consistência

Um especialista humano, tal como um mecânico automobilístico, utiliza sua experiência com falhas em dispositivos para chavear sua memória para casos diagnósticos anteriores, quando ele se depara com um conjunto similar de sintomas. Entretanto, se não existirem tais casos anteriores, ele é capaz de raciocinar sobre o comportamento esperado dos subsistemas do automóvel, baseando-se nos princípios de funcionamento dos mesmos. Através desse procedimento, ele considera que qualquer desvio do comportamento esperado é causado por falhas.

A abordagem baseada em consistência para o diagnóstico de falhas utiliza exatamente esta visão. Um modelo do dispositivo é usado para prever o seu comportamento esperado, que então é comparado com o seu comportamento real. Se discrepâncias são detectadas, existe um problema diagnóstico, uma vez que o dispositivo foi corretamente projetado e construído, mas agora exibe indicações de mau funcionamento.

A tarefa do processo diagnóstico é determinar o que está errado com o dispositivo, com o objetivo de restabelecer a sua funcionalidade original. Possíveis respostas podem ser que certo conjunto de componentes do dispositivo não está funcionando corretamente ou que tal dispositivo foi afetado em uma forma ainda mais séria que alterou a sua estrutura. Falhas estruturais, tal como a conexão entre dois fios adjacentes, estabelecem novas interações entre os componentes de um dispositivo que tornam impossível a sua identificação através de componentes defeituosos. Dessa forma, na ausência de falhas estruturais, cujo diagnóstico ainda é um problema aberto, as possíveis respostas, denominadas diagnósticos, podem ser estabelecidas em termos de componentes defeituosos.

A seguir, apresentamos como o modelo, isto é, a descrição da estrutura e comportamento de um dispositivo a ser diagnosticado, e o conceito de diagnóstico são formalmente formulados em lógica de primeira ordem.

3.2.1 Formulação teórica do diagnóstico baseado em modelos

Reiter [2] apresenta uma teoria de diagnóstico baseada nos princípios básicos que pode ser aplicada a uma grande variedade de domínios, tais como diagnóstico médico, diagnóstico de falhas em circuitos digitais e analógicos, etc. Para alcançar tal generalidade, utiliza-se lógica de primeira ordem como a linguagem de representação das informações dependentes da tarefa a ser realizada. O apêndice A apresenta uma breve introdução à lógica matemática, incluindo a lógica de primeira ordem ou cálculo de predicados. Um estudo mais profundo deste tema pode ser encontrado em [28, 29].

Definição 3.2.1 (Sistema (Reiter)) *Um sistema é um par $(DS, COMPS)$ onde:*

(1) DS , a descrição do sistema, é um conjunto de sentenças em lógica de primeira ordem, descrevendo o comportamento dos componentes e a estrutura do sistema (ou seja, as conexões entre os componentes que formam o sistema);

(2) $COMPS$, o conjunto dos componentes do sistema, é um conjunto finito de constantes.

A Definição 3.2.1 apresenta o conceito de um sistema independente de domínio, cujo objetivo é formalizar de forma tão abstrata quanto possível os conceitos de um componente e de uma coleção de componentes que interagem. A Definição 3.2.2, por sua vez, apresenta o conceito formal de uma observação. As observações representam o comportamento real de um sistema e são essenciais em qualquer aplicação de diagnóstico, baseada em modelos ou não. No caso de um sistema diagnóstico baseado em modelos, a única forma de determinarmos se algum dispositivo está funcionando corretamente é comparar as informações providas pelas observações com aquelas derivadas pelos modelos.

Definição 3.2.2 (Observação (Reiter)) *Uma observação é um conjunto finito de sentenças em lógica de primeira ordem que representa o comportamento real de um sistema e $(DS, COMPS, OBS)$ corresponde a um sistema $(DS, COMPS)$ para o qual foi fornecida a observação OBS .*

Segundo a Definição 3.2.1, um sistema é composto por um conjunto de componentes ($COMPS$), cujo comportamento é descrito em lógica de primeira ordem. Dessa forma,

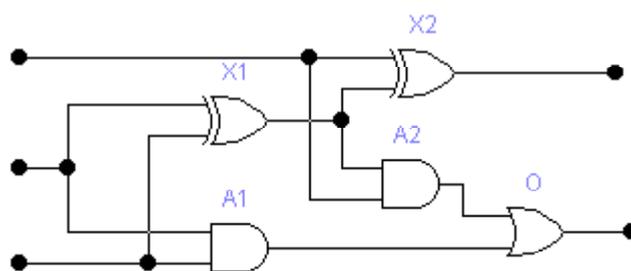


Figura 3.1: Somador completo

constantes e predicados são usados para nomear os componentes e especificar as suas interações, feitas através de portas de entrada e saída. Tipicamente, a descrição de um sistema (DS) descreve o comportamento correto dos seus componentes, usando o predicado $ok(C)$ para representar a suposição de que cada componente (C) está funcionando corretamente. O uso deste predicado permite que a conclusão $\neg ok(C)$, ou seja, a conclusão de que o componente C não está funcionando corretamente, seja derivada, quando as observações contradizerem os valores preditos pelo modelo.

A descrição formal de um sistema é composta pela definição dos axiomas do domínio do sistema modelado, por um conjunto de sentenças, relacionando os tipos de componentes e o comportamento das suas saídas em função das suas entradas, e pela definição dos componentes que formam o sistema e suas conexões. Os axiomas do domínio definem as características básicas dos componentes do sistema, enquanto a definição dos componentes e suas conexões definem a estrutura do mesmo. O comportamento do sistema como um todo é definido pelo comportamento dos seus componentes e suas conexões.

Para exemplificar o uso das definições acima, usaremos o modelo de um somador completo, definido em [27], formado por cinco portas lógicas conectadas por fios (Figura 3.1). Cada porta lógica possui dois portos de entrada ($E1$ e $E2$) e um porto de saída (S). A Descrição 3.1 apresenta parte de uma possível descrição formal deste circuito, feita em lógica de primeira ordem.

Descrição 3.1 Modelo de um somador completo

Axiomas do domínio (define axiomas para portas lógicas em circuitos digitais)

$$\forall x \in COMPS \Rightarrow [Valor(E1(x), 1) \otimes Valor(E1(x), 0) \wedge \\ Valor(E2(x), 1) \otimes Valor(E2(x), 0) \wedge \\ Valor(S(x), 1) \otimes Valor(S(x), 0)],$$

onde \otimes e \wedge denotam as operações lógicas “ou-exclusivo” e “e”, respectivamente.

Comportamento (dos tipos de componentes)

$$\begin{aligned} \forall x \text{ PortaE}(x) \wedge ok(x) \wedge Valor(E1(x), 1) \wedge Valor(E2(x), 1) &\Rightarrow Valor(S(x), 1) \\ \forall x \text{ PortaE}(x) \wedge ok(x) \wedge Valor(E1(x), 0) &\Rightarrow Valor(S(x), 0) \\ \forall x \text{ PortaE}(x) \wedge ok(x) \wedge Valor(E2(x), 0) &\Rightarrow Valor(S(x), 0) \\ \forall x \text{ PortaOU}(x) \wedge ok(x) \wedge Valor(E1(x), 0) \wedge Valor(E2(x), 0) &\Rightarrow Valor(S(x), 0) \\ \text{etc.} \end{aligned}$$

Componentes (define os componentes do sistema, com seus respectivos tipos)

$$\begin{aligned} \text{PortaE}(A1), \text{PortaE}(A2), \\ \text{PortaXOR}(X1), \text{PortaXOR}(X2), \\ \text{PortaOU}(O1) \end{aligned}$$

Estrutura (define as interações entre os componentes do sistema)

$$\begin{aligned} E1(X1) = E1(A1), E2(X1) = E2(A1), \\ E1(X2) = E2(A2), E2(X2) = E1(A2), \\ S(X1) = E2(X2), S(A1) = E2(O1), \\ S(A2) = E1(O1) \end{aligned}$$

Analisando o modelo do somador completo, verificamos que os axiomas definem que os portos de entrada e saída devem possuir valores binários. As sentenças definem o comportamento dos tipos de portas lógicas “e”, “ou” e “ou-exclusivo”, usando o predicado *Valor* para ler o valor corrente dos portos de entrada e para atribuir valores para os portos de saída. Os predicados *E1*, *E2* e *S* retornam os valores atuais das entradas e saídas de uma porta lógica, enquanto os predicados *PortaE*, *PortaOU* e *PortaXOR* identificam o tipo da porta lógica. Além disso, esta descrição mostra que não existe uma representação canônica para um dispositivo. Quando definimos a descrição de um sistema, tomamos decisões quanto ao que deve ser representado ou não. Esta descrição, por exemplo, não descreve o comportamento dos fios que ligam as portas lógicas. Dessa forma, esse modelo é insuficiente para o diagnóstico de falhas provocadas pelo rompimento destes fios. Isto demonstra o quanto o modelo escolhido para representar um dispositivo e seu comportamento é crucial para determinar a capacidade de um sistema diagnóstico baseado em modelos.

Agora que já definimos formalmente o modelo (ou descrição) de um sistema, veremos como este modelo pode ser usado para realizar o diagnóstico de falhas em seus respectivos dispositivos físicos, quando confrontados com observações que contradizem as suas predições.

3.2.1.1 Diagnósticos

Na ausência de falhas estruturais, podemos definir o diagnóstico de falhas como o processo de localização de componentes defeituosos. Da perspectiva da localização de falhas, nós precisamos distinguir apenas dois modos de comportamento para cada componente: o comportamento correto e um desvio arbitrário deste comportamento. Esta abordagem é suficiente para a tarefa de manutenção dos sistemas eletrônicos de um veículo automotor, na qual não é necessário distinguir falhas que sejam corrigidas pela substituição de um mesmo componente. Dessa forma, neste trabalho, consideraremos apenas os modos de comportamento correto e incorreto de cada componente, denominados $ok(C)$ e $\neg ok(C)$, respectivamente.

Definição 3.2.3 (Atribuição completa de modos de comportamento) *Considere $DEFEITUOSO \subseteq COMPS$. Uma atribuição de modos de comportamento, $D(DEFEITUOSO, OK)$, é dada pelo conjunto*

$$\{ok(x) \mid x \in OK\} \cup \{\neg ok(x) \mid x \in DEFEITUOSO\}$$

e denominada completa quando $DEFEITUOSO \cup OK = COMPS$.

As Definições 3.2.3 e 3.2.4 definem um diagnóstico como uma atribuição completa de modos de comportamento para todos os componentes de um sistema [27]. Dessa forma, o conjunto de componentes de um sistema ($COMPS$) é particionado em dois subconjuntos, contendo respectivamente os componentes cujo comportamento é correto (OK) e cujo comportamento é incorreto ou defeituoso ($DEFEITUOSO$).

Definição 3.2.4 (Diagnósticos) *Um diagnóstico para uma descrição de sistema DS , componentes $COMPS$ e observações OBS é uma atribuição completa de modos de comportamento $D(DEFEITUOSO, OK)$, tal que $DS \cup OBS \cup D(DEFEITUOSO, OK)$ é consistente, ou seja, não existe nenhuma discrepância entre os valores preditos pelo modelo e os respectivos valores observados.*

De acordo com as definições acima, os diagnósticos podem ser muito numerosos. O espaço de busca por potenciais diagnósticos é $2^{|COMPS|}$, onde $|COMPS|$ é o número de componentes do sistema. Dessa forma, torna-se necessário o uso de uma forma compacta de representação dos diagnósticos, obtida através do seguinte princípio: Um diagnóstico é determinado pelos menores conjuntos de componentes, cuja suposição de que estão defeituosos, juntamente com a suposição de que os demais componentes estão funcionando corretamente é consistente com a descrição do sistema e com as observações existentes. Este princípio é apresentado formalmente na Definição 3.2.5. Apesar dos diagnósticos mínimos constituírem uma forma compacta de representação dos diagnósticos, o seu número também pode crescer exponencialmente.

Definição 3.2.5 (Diagnósticos mínimos) *Um diagnóstico $D(DEFEITUOSO, OK)$ é mínimo se e somente se para todo subconjunto próprio $DEFEITUOSO' \subset DEFEITUOSO$,*

a atribuição completa de modos de comportamento $D(DEFEITUOSO', OK)$ não é um diagnóstico.

3.3 General diagnostic engine (GDE)

Em [3], de Kler e Williams descrevem um arcabouço para diagnóstico que computa diagnósticos mínimos e que subsequente influenciou quase todos os trabalhos na área de diagnóstico baseado em modelos. Este arcabouço, é denominado *General diagnostic engine* (GDE) e possui as seguintes características básicas:

- trata falhas múltiplas (múltiplos componentes defeituosos), em contraste com os sistemas previamente definidos, que trabalhavam com a hipótese de falha única;
- computa diagnósticos a partir de conflitos mínimos, isto é, a partir de conjuntos mínimos de suposições quanto ao funcionamento correto de componentes que não podem ser verdadeiras simultaneamente, derivados das inconsistências descobertas;
- explora extensivamente um sistema de manutenção da verdade baseado em suposições para identificar os conjuntos de suposições conflitantes, usando-os como base para determinar os diagnósticos candidatos e os pontos de prova ótimos.

A abordagem do GDE baseia-se na caracterização das discrepâncias entre o comportamento predito pelo modelo e o comportamento observado como violações de suposições. Um componente garante o seu comportamento de acordo com seu modelo desde que todas as suas suposições sejam válidas. Se quaisquer destas suposições forem falsas, então o dispositivo diverge do seu modelo e o mesmo não mais se aplica. Esta caracterização em termos de violações de suposições é muito geral, pois uma suposição pode receber diferentes significados, dependendo do domínio de aplicação. Por exemplo, em eletrônica, uma suposição pode representar o funcionamento correto de cada componente, enquanto em um domínio científico, pode representar a validade de uma hipótese.

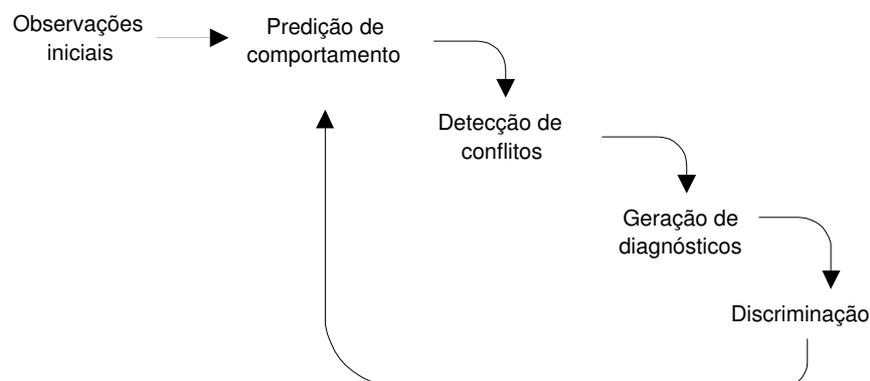


Figura 3.2: Processo diagnóstico do GDE

O GDE assume que os sintomas não são diretamente observados, sendo deduzidos indiretamente das observações. Intuitivamente, um sintoma é qualquer diferença entre uma predição feita por um procedimento de inferência e uma observação. De forma mais geral, um sintoma é qualquer inconsistência descoberta pelo procedimento de inferência, podendo ocorrer entre duas predições (inferidas de medidas distintas) ou entre uma medida e uma predição (inferida de alguma outra medida).

O processo diagnóstico do GDE é guiado pelos sintomas, onde cada sintoma nos diz que uma ou mais suposições possivelmente foram violadas. Intuitivamente, um conflito é um conjunto de suposições que dão suporte a um sintoma e, desta forma, conduz a uma inconsistência. Em domínios complexos, qualquer sintoma pode gerar um grande número de conflitos, incluindo o conflito formado por todos os componentes do sistema. Assim o conjunto de conflitos é representado concisamente pelos conflitos mínimos, ou seja, os conflitos que não possuem nenhum subconjunto formal que também seja um conflito.

Este processo pode ser dividido em quatro fases principais, organizadas em um ciclo que se repete até que o diagnóstico para as falhas do dispositivo seja encontrado (Figura 3.2). A seguir, discutimos cada uma destas fases e suas relações com os conceitos introduzidos anteriormente.

3.3.1 Predição de comportamento

O GDE assume que o modelo de um dispositivo é descrito em termos de sua topologia e do comportamento de cada um dos seus componentes e que o único tipo de diferença entre modelos e dispositivos é o funcionamento correto ou não dos seus componentes. Ele considera ainda que as medidas possuem um alto custo e assim nem todos os valores dos portos dos componentes são conhecidos. Dessa forma, alguns valores devem ser deduzidos de outros valores, a partir das sentenças de comportamento. A dedução de valores é feita através da propagação dos valores medidos (observações) e deduzidos através da estrutura do sistema.

O processo de predição de comportamento usado pelo GDE é denominado propagação de restrições e opera em portos, valores e restrições. Portos representam variáveis de estado como tensões, níveis lógicos ou fluxos de fluidos. Uma restrição define uma condição que os portos devem satisfazer. Por exemplo, a lei de Ohm, $V = IR$, é representada como uma restrição entre os portos V , I e R de um dado componente $X1$. Dado um conjunto de valores iniciais, a propagação de restrições atribui a cada porto, um valor que satisfaça as restrições definidas. O passo de inferência básico é encontrar uma restrição que permita determinar um valor para um porto previamente sem valor. Por exemplo, ao descobirmos os valores $V = 2$ e $I = 1$, usando a restrição $V = IR$, calculamos o valor $R = 2$.

De maneira geral, durante a predição de comportamento, a descrição do sistema (DS) e as observações (OBS) são usadas para derivar conclusões sobre os valores dos portos. O processo é centrado nos componentes do dispositivo e deriva conclusões sobre portos de um componente sempre que existirem informações suficientes sobre outros portos do mesmo componente. Cada conclusão é derivada a partir dos valores dos portos dos quais ela depende e de suposições sobre o comportamento correto dos componentes.

O mecanismo de predição do GDE usa um sistema de manutenção da verdade baseado em suposições (*Assumption-Based Truth Maintenance System* ou ATMS) [30, 31, 32] como repositório de inferências. O ATMS armazena cláusulas proposicionais, denominadas

justificativas, da forma

$$\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \dots \wedge \beta_n \Rightarrow A.$$

Um subconjunto distinto, *SUPS*, do conjunto de átomos proposicionais, *PROP*, é denominado de suposições. Os modos de comportamento de um componente, por exemplo *ok(X1)*, são tratados como suposições. O conjunto de átomos derivados a partir de um conjunto de suposições *E*, denominado ambiente, é denominado o contexto de *E* e denotado por *ctx(E)*. Considerando o exemplo da lei de Ohm, a conclusão $R = 2$ seria comunicada ao ATMS através da seguinte justificativa:

$$ok(X1) \wedge (V = 2) \wedge (I = 1) \wedge (V = IR) \Rightarrow (R = 2).$$

O ATMS é uma ferramenta geral de Inteligência Artificial que permite o raciocínio em múltiplos contextos. Este tipo de raciocínio é caracterizado por considerar simultaneamente todos os contextos consistentes (*ctx(E)*), referentes a todos os ambientes (*E*) formados por subconjuntos das suposições do modelo de um sistema (*SUPS*). No ATMS, todas as proposições são rotuladas com o conjunto completo de ambientes mínimos consistentes, ou seja, conjuntos de suposições que devem ser verdadeiras para que a proposição seja verdadeira, dos quais elas são deriváveis. Dessa forma, o rótulo de uma proposição é definido como

$$Rótulo(p) = \{E \subset SUPS \mid (E \text{ é consistente} \wedge p \in ctx(E) \wedge \forall E' \subset E, p \notin ctx(E'))\}.$$

As justificativas são usadas para registrar as conclusões realizadas pelo mecanismo de inferência. O rótulo de uma proposição é calculado pela propagação e combinação dos ambientes na rede de justificativas, usando-se operações básicas de conjuntos. Devido ao fato de as inferências serem armazenadas como justificativas, uma inferência é feita uma vez para o primeiro contexto em que ela é derivada e torna-se válida automaticamente em cada contexto caracterizado como superconjunto dos seus ambientes. Quando os antecedentes de uma justificativa contêm outros contextos, isto causa apenas uma atualização do rótulo

da proposição conseqüente, evitando-se o recálculo do rótulo.

Entretanto, os rótulos calculados por um ATMS podem crescer muito, impedindo o seu uso em aplicações de grande porte. A focalização nos contextos interessantes [33] evita este problema, mantendo as propriedades essenciais do ATMS.

3.3.2 Detecção de conflitos

Sempre que um porto P assume valores contraditórios, como por exemplo α e β , em um mesmo contexto, o conjunto de suposições que o suporta deve ser identificado como inconsistente, ou seja, as suas suposições não podem ser todas verdadeiras simultaneamente. Se uma suposição indica o funcionamento correto de um componente, então um ambiente inconsistente indica que pelo menos um dos componentes representados está defeituoso. Um ambiente é definido como inconsistente através da criação de justificativas da forma

$$Valor(V, \alpha) \wedge Valor(V, \beta) \Rightarrow \perp, \text{ onde a constante } \perp \text{ representa uma contradição.}$$

Ao inserirmos tais justificativas no ATMS, ele calcula e mantém um banco de dados de ambientes inconsistentes mínimos, denominados “nogoods”. Um ambiente inconsistente é considerado mínimo se o seu conjunto de suposições não possui nenhum subconjunto próprio que forme outro ambiente inconsistente. Dessa forma, a identificação de conflitos mínimos é trivial para o GDE, sendo determinada simplesmente pelo conjunto de ambientes inconsistentes mínimos, mantidos pelo ATMS.

Considere novamente o circuito somador completo da Figura 3.1 e suponha que o seguinte conjunto de observações seja informado ao GDE.

$$OBS = \{Valor(S(X2), 1), Valor(E1(X2), 1), Valor(E1(X1), 1), \\ Valor(E2(A1), 0), Valor(S(O), 1)\}$$

O GDE considera as observações como verdades universais, que não dependem de qualquer suposição, sendo válidas em todos os ambientes lógicos (representados pelo ambiente vazio, ϕ). Isto é registrado no ATMS através da inserção dessas observações como premissas, ou seja, proposições cujo rótulo contém apenas o ambiente vazio:

$$\begin{aligned}
Rótulo(Valor(S(X2), 1)) &= \{\phi\}, & Rótulo(Valor(E1(X2), 1)) &= \{\phi\}, \\
Rótulo(Valor(E1(X1), 1)) &= \{\phi\}, & Rótulo(Valor(E2(A1), 0)) &= \{\phi\}, \\
Rótulo(Valor(S(O), 1)) &= \{\phi\}.
\end{aligned}$$

Com a inserção das observações, a fase de predição de comportamento do GDE usa a regra de comportamento $Valor(E1(X2), 1) \wedge Valor(E2(X2), 1) \wedge ok(X2) \Rightarrow Valor(S(X2), 0)$ para calcular $Valor(O(X2), 0)$. O rótulo de $Valor(O(X2), 0)$ é calculado pelo ATMS a partir dos rótulos das proposições antecedentes:

$$\begin{aligned}
Rótulo(Valor(E1(X2), 1)) &= \{\phi\}, \\
Rótulo(Valor(E2(X2), 1)) &= \{\{ok(X1)\}, \{ok(A1), ok(O), ok(A2)\}\}, \\
Rótulo(ok(X2)) &= \{ok(X2)\} \\
\Rightarrow Rótulo(Valor(O(X2), 0)) &= \{\{ok(X1), ok(X2)\}, \{ok(A1), ok(O), ok(A2), ok(X2)\}\}
\end{aligned}$$

A proposição $ok(X2)$ depende somente da suposição de que $X2$ está correto, enquanto o rótulo de $Valor(E(X2), 1)$ reflete os dois diferentes caminhos pelos quais este valor pode ser derivado do modelo. Como $Valor(S(X2), 1)$ é uma observação, qualquer conjunto de suposições que suportem $Valor(S(X2), 0)$ é um conflito. Conseqüentemente, o ATMS detecta dois ambientes inconsistentes mínimos: $\{ok(X1), ok(X2)\}$ e $\{ok(A1), ok(O), ok(A2), ok(X2)\}$. Cada um destes ambientes estabelece que pelo menos um dos componentes citados nas suas suposições deve estar defeituoso, ou seja,

$$\neg ok(X1) \vee \neg ok(X2) \text{ e } \neg ok(A1) \vee \neg ok(O) \vee \neg ok(A2) \vee \neg ok(X2).$$

O GDE denomina estas cláusulas disjuntivas de conflitos, considerando-os conjuntos de suposições que não podem ser todas verdadeiras simultaneamente. Um conflito é mínimo quando não possui nenhum conflito como subconjunto próprio.

3.3.3 Geração de diagnósticos candidatos

Um diagnóstico candidato, ou simplesmente candidato, é uma hipótese particular que explica as diferenças entre um dispositivo e seu respectivo modelo. Por exemplo, “ $X1$ e $A1$ estão defeituosos” é um diagnóstico candidato para o exemplo da seção anterior. No GDE, a meta do processo diagnóstico é identificar e refinar o conjunto de candidatos consistentes com o conjunto de observações. Um candidato é representado por um conjunto de suposições, onde as suposições explicitamente mencionadas são falsas e as demais são verdadeiras. Este conjunto é construído de tal maneira que se todas as suas suposições forem falsas, então todos os sintomas conhecidos são explicados. Assim, cada conjunto de suposições correspondente a um candidato possui uma interseção não vazia com todos os conflitos mínimos identificados.

No caso do domínio de circuitos eletrônicos, que pode ser estendido para o domínio dos componentes eletrônicos de um automóvel, um candidato pode representar um conjunto de componentes defeituosos, onde qualquer componente não mencionado é garantido estar operando corretamente. Antes de qualquer medida ser feita, ou seja, quando o conjunto de observações é vazio, qualquer componente pode estar operando de maneira normal ou anormal. Considerando o circuito da Figura 3.1, o tamanho do espaço de candidatos inicial, cujo crescimento é exponencial com o número de componentes, seria de $2^5 = 32$ candidatos. Dessa forma, é essencial que os candidatos sejam representados concisamente. Assim, o GDE representa o espaço de candidatos através de candidatos mínimos. A representação e manipulação do espaço de candidatos em termos de candidatos mínimos é crucial para a abordagem do GDE, pois embora o espaço de candidatos cresça exponencialmente com o número de componentes potencialmente defeituosos, normalmente este espaço pode ser representado por poucos candidatos mínimos que explicam todos sintomas identificados.

O processo diagnóstico do GDE é incremental, ou seja à medida que observações são inseridas e novos conflitos mínimos identificados, o GDE refina continuamente o espaço de candidatos, que por sua vez são usados para guiar a seleção de medidas adicionais (veja Seção 3.3.4). Dessa forma, os candidatos são gerados incrementalmente, usando-se o novo

conflito mínimo e o antigo conjunto de candidatos mínimos. A geração do novo conjunto é feita da seguinte maneira: todo candidato mínimo que não explica o novo conflito mínimo é substituído por um ou mais dos seus superconjuntos que sejam candidatos mínimos. Isto é feito através da geração de uma série de tentativas de candidatos mínimos, cada um dos quais contendo o antigo candidato e uma suposição do novo conflito. Todo novo candidato duplicado ou que possua um subconjunto candidato é eliminado e os candidatos restantes são adicionados ao novo conjunto de candidatos mínimos.

A geração de candidatos possui várias propriedades interessantes. Primeiro, o tamanho do conjunto de candidatos mínimos pode aumentar ou diminuir como resultado de uma nova observação. Entretanto, um candidato eliminado, ou seja, substituído por candidatos maiores, nunca reaparece. Segundo, se uma mesma suposição aparece em todos os candidatos mínimos (e por conseqüência, em todos os candidatos), então esta suposição é necessariamente falsa, ou seja, o componente que ela representa está defeituoso. Terceiro, a suposição de que existe apenas uma única falha equivale a assumirmos que todos os candidatos devem conter apenas um elemento, podendo ser obtidos através do cruzamento de todos os conflitos.

3.3.4 Discriminação de diagnósticos candidatos

Para reduzir o conjunto de diagnósticos candidatos o GDE propõe a execução de medidas, cujo objetivo é a geração de novas observações. O seu método de seleção da próxima medida busca escolher a medida que melhor discrimine os candidatos, ou seja, que na média nos leve à descoberta do diagnóstico atual em um número mínimo de medidas.

As estratégias de predição de comportamento e detecção de conflitos do GDE identificam todas as possíveis predições de valores para cada ambiente (conjunto de suposições), fornecendo a base para um procedimento diagnóstico diferencial que permite ao GDE identificar as possíveis medidas e as suas respectivas conseqüências. O banco de dados do GDE representa explicitamente os valores de X_i e os ambientes que os suportam: $\langle X_i = v_{ik}, e_{ik1}, \dots, e_{ikm} \rangle$. Se X_i é medido ser v_{ik} , então os ambientes apoiando qualquer

valor distinto do valor medido são necessariamente conflitos. Se v_{ik} não é igual a nenhum dos valores preditos para X_i , então todos os ambientes que apóiam cada um dos valores preditos de X_i são conflitos. Assim, dado o banco de dados do GDE, torna-se simples identificar medidas úteis e seus respectivos resultados e conflitos resultantes.

Dado um estágio particular do processo diagnóstico, o GDE analisa as conseqüências de cada medida para determinar qual a próxima a ser executada. Para isso, ele usa uma função de custo que determina para cada possível resultado de uma medida, o quão difícil será, em termos do número de medidas adicionais, identificar o diagnóstico atual. Esta função de custo é a entropia (H) das probabilidades do candidato, dada por:

$$H = - \sum p_i \log p_i, \quad (3.1)$$

onde p_i é a probabilidade do candidato C_i ser o diagnóstico atual, dado o resultado da medida hipotética.

A função entropia (H) calcula o custo esperado para identificar o diagnóstico atual da seguinte maneira: O custo para localizar um candidato de probabilidade p_i é proporcional a $\log p_i^{-1}$ (busca binária por p_i^{-1} objetos) e, dessa forma, o custo esperado para identificar o diagnóstico atual é proporcional à soma do produto da probabilidade de cada candidato ser o diagnóstico atual e o custo de identificar tal candidato, ou seja:

$$\sum p_i \log p_i^{-1} = - \sum p_i \log p_i, \quad (3.2)$$

Considerando que toda medida possui um custo igual, o objetivo do processo diagnóstico é identificar o diagnóstico atual realizando um número mínimo de medidas. A melhor medida é aquela que provoca a mínima entropia esperada das probabilidades dos candidatos. Assumindo que o processo de realização de uma medida não influencia o valor medido, a entropia esperada $H_e(x_i)$ depois de medir a quantidade x_i é dada por:

$$H_e(x_i) = \sum_{k=1}^m p(x_i = v_{ik}) H(x_i = v_{ik}) \quad (3.3)$$

onde v_{i1}, \dots, v_{im} são todos os possíveis valores para x_i e $H(x_i = v_{ik})$ é a entropia resultante, se x_i é medido ser v_{ik} .

A entropia esperada é calculada a partir da informação disponível, ou seja, a cada passo calcula-se $H(x_i = v_{ik})$, determinando as novas probabilidades dos candidatos a partir das probabilidades atuais e do resultado hipotético de $x_i = v_{ik}$. As probabilidades iniciais dos candidatos são calculadas das probabilidades iniciais de falha dos componentes, obtidas dos seus fabricantes ou por dados empíricos. Assumindo que os componentes falham de maneira independente¹, a probabilidade inicial de um candidato qualquer ser o diagnóstico atual, C_a , é dada por:

$$p_i = \prod_{c \in C_i} p(c \in C_a) \prod_{c \notin C_i} (1 - p(c \in C_a)). \quad (3.4)$$

Dado o resultado da medida $x_i = v_{ik}$, esta probabilidade é atualizada de acordo com a regra de Bayes:

$$p(C_l | x_i = v_{ik}) = \frac{p(x_i = v_{ik} | C_l)p(C_l)}{p(x_i = v_{ik})}. \quad (3.5)$$

A avaliação de $p(x_i = v_{ik} | C_l)$ pode ser dividida em três casos. Se C_l prediz $x_i = w_{ik}$, onde $w_{ik} \neq v_{ik}$, então a probabilidade condicional é 0, ou seja,

$$p(x_i = v_{ik} | C_l) = 0. \quad (3.6)$$

Se $w_{ik} = v_{ik}$, então a probabilidade condicional é 1, ou seja,

$$p(x_i = v_{ik} | C_l) = 1. \quad (3.7)$$

Para finalizar, se C_l não prediz nenhum valor para x_i , o GDE assume que todos os possíveis valores de x_i (existindo m valores) são igualmente prováveis, ou seja,

¹Esta abordagem pode ser estendida para falhas dependentes, mas requerendo um grande volume de dados

$$p(x_i = v_{ik} | C_l) = 1/m. \quad (3.8)$$

Substituindo estas probabilidades na regra de Bayes, obtemos as seguintes regras para a atualização das probabilidades condicionais dos candidatos:

$$P(C_l | x_i = v_{ik}) = \begin{cases} 0, & \text{se o candidato prediz } x_i \neq v_{ik} \\ \frac{p(C_l)}{p(x_i=v_{ik})}, & \text{se o candidato prediz } x_i = v_{ik} \\ \frac{p(C_l)/m}{p(x_i=v_{ik})}, & \text{se o candidato não prediz valores para } x_i \end{cases} \quad (3.9)$$

3.4 Diagnóstico de falhas em veículos automotores: uma abordagem baseada em modelos

Para demonstrar a utilidade das técnicas de diagnóstico baseadas em modelos no domínio automobilístico, a *European Commission*, juntamente com diversas empresas do setor automobilístico europeu² e instituições de pesquisa, criaram o projeto *Vehicle Model-based Diagnosis* (VMBD) [34, 35]. Durante este projeto, foi feita uma análise das características do domínio automobilístico e dos requisitos a serem suportados por sistemas diagnósticos baseados em modelos, usados para o diagnóstico de falhas neste domínio. Os resultados deste projeto incluem a definição de um formalismo geral para a construção de uma biblioteca de modelos qualitativos de componentes, a definição de uma arquitetura para a solução do problema diagnóstico, tanto para sistemas *on-board* quanto *off-board*, e da experimentação em três aplicações reais.

Como o projeto VMBD trata da aplicação de técnicas de diagnóstico baseado em mo-

²DaimlerChrysler, GenRad, Volvo, Fiat, Bosch e Magneti Marelli

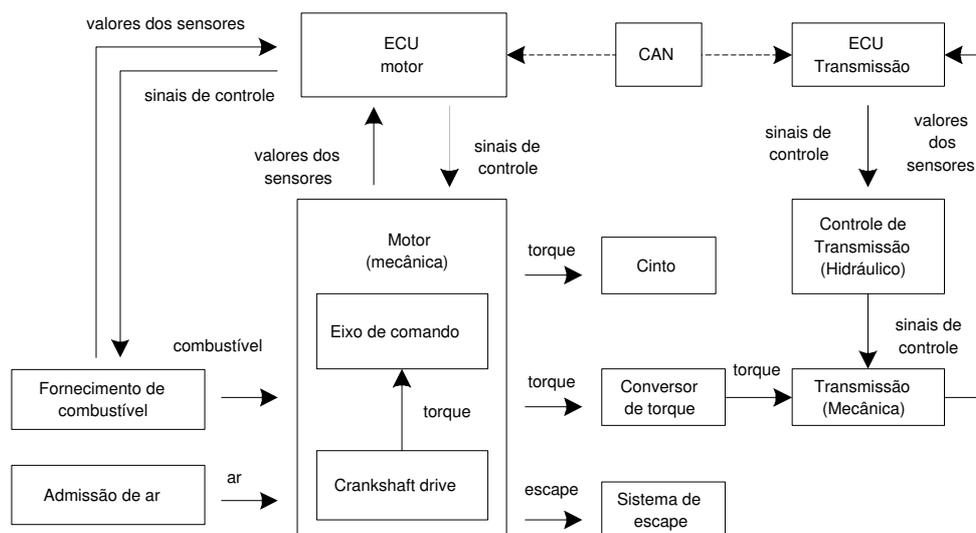


Figura 3.3: Esquemático do sistema de propulsão de um veículo

delos no domínio automobilístico, as próximas seções apresentam as suas principais características, definições e resultados.

3.4.1 Domínio de aplicação

O sistema de propulsão é certamente o núcleo de um veículo e, dessa forma, foi um dos primeiros candidatos para a aplicação da tecnologia de diagnóstico baseado em modelos pelo VMBD. O sistema de propulsão, mostrado na Figura 3.3, é formado basicamente pelos sistemas de admissão de ar, de fornecimento de combustível, de escapamento de gases, de controle eletrônico do motor e de transmissão automática (quando existente). Entretanto, devido à grande complexidade deste sistema, seria uma tarefa extremamente grande modelar tudo de uma vez. Assim, foram escolhidos dois sistemas de injeção de diesel e um sistema de transmissão automática como as aplicações nas quais a tecnologia seria testada. Todos os três sistemas são complexos e modulares, consistindo de componentes eletrônicos, elétricos, hidráulicos e pneumáticos. A descrição destes sistemas foge do escopo deste trabalho e pode ser encontrada em [34].

3.4.2 Classes de sistemas diagnósticos

A distinção dos sistemas diagnósticos entre embarcados (*on-board*, ou seja, executando no próprio veículo) e *off-board* (executando em uma estação de serviços) é usual, à medida que estes sistemas possuem diferentes requisitos para a localização e identificação de falhas e diferem significativamente em termos de medidas disponíveis e reprodução de condições.

Sistemas diagnósticos embarcados visam a seleção apropriada de ações de recuperação. O sistema de injeção eletrônica, por exemplo, monitora continuamente parte dos sinais dos seus sensores, permitindo que o sistema diagnóstico embarcado possa detectar falhas baseando-se na verificação de limites e plausibilidade de sinais. Quando uma falha é detectada, ele executa ações de recuperação que estendem-se desde pequenas reduções de desempenho até a parada total do motor, dependendo da provável falha detectada e dos seus efeitos. Entretanto, devido à escassez de sensores, em muitos casos a unidade de controle erra na discriminação entre as diferentes possíveis causas que levam a uma falha. Consequentemente, o sistema frequentemente aplica uma ação de recuperação mais restritiva do que a necessária. Assim, o principal objetivo de um sistema diagnóstico embarcado é a redução do número de situações em que ações de recuperação muito severas são tomadas.

Os sistemas diagnósticos *off-board*, por outro lado, visam o reparo ou, mais especificamente, a troca de componentes defeituosos. A tarefa é localizar a falha ao nível do menor componente substituível. Assim, sistemas *off-board* requerem diferentes níveis de precisão (granularidade) para a localização de falhas. Estes sistemas podem acessar mais observações, providas pela execução de testes mais avançados e por observações feitas pelos motoristas e mecânicos. Ele deve ainda propor a execução de testes, com o objetivo de particionar o sistema e discriminar entre as possíveis causas das falhas. Em contraste com os sistemas embarcados, onde as medidas são obtidas facilmente através das leituras dos sensores, a obtenção de observações apropriadas em uma oficina mecânica tem um preço nem sempre desprezível. A redução dos custos para execução de testes, por exemplo, desmontagem de sistemas, justifica o uso de algoritmos diagnósticos mais elaborados.

3.4.3 Análise de requisitos

Os componentes de um sistema de propulsão de um veículo automotor estabelecem um grande número de desafios para a aplicação de técnicas de diagnóstico automático assistidas por computadores. A base para a escolha de tecnologias de diagnóstico baseadas em modelos, para enfrentar tais desafios, foi a constatação de que a maioria dos complexos sistemas em um veículo compartilham as seguintes características em relação a suas funções:

- existe uma decomposição natural em subsistemas com poucos componentes;
- na maioria dos casos, os defeitos de um carro são provocados pela falha de algum componente;
- o comportamento dos componentes pode ser descrito por relações entre variáveis locais e parâmetros;
- o comportamento do sistema é determinado pelo comportamento dos seus componentes e suas conexões.

O domínio automobilístico possui uma série de requisitos que devem ser tratados durante a aplicação das tecnologias de diagnóstico baseadas em modelos. Estes requisitos geram uma série de dificuldades para a aplicação desta tecnologia e foram analisados durante o decorrer do projeto VMBD.

O problema da variação No domínio automobilístico, existe uma grande variedade de sistemas a serem modelados, muitos dos quais desempenhando funções semelhantes e usando componentes comuns. Segundo o VMBD, este problema pode ser resolvido através de uma modelagem composicional, na qual o modelo de um sistema é construído através do arranjo de modelos de componentes genéricos.

Fenômenos de diferentes domínios físicos Os sistemas a serem modelados requerem a interação de componentes de diferentes domínios físicos (elétrico, eletrônico,

hidráulico, pneumático e mecânico). Estes fenômenos determinam o comportamento não apenas dos componentes, mas do sistema como um todo. Esta interação de fenômenos cria a necessidade de se criar modelos de componentes que contemplem todos estes diferentes domínios físicos e que possam ser usados em um arcabouço comum. Desta forma, o VMBD adotou uma modelagem orientada nos componentes e uma biblioteca de modelos com componentes reutilizáveis.

Subsistemas dinâmicos Todos os sistemas considerados no projeto possuem estados internos dependentes das entradas anteriores, sendo exemplos de sistemas dinâmicos. Os efeitos de falhas podem ser compensados pelo controle e, assim, as falhas podem ser visíveis somente em um subconjunto dos modos de operação do veículo (durante a partida, aceleração máxima, etc.) ou durante transições de modos de operação.

Uma questão crucial é se o diagnóstico de falhas relevantes demanda a predição extensiva de comportamento sobre o tempo, isto é, algum tipo de simulação (quantitativa ou qualitativa). Para responder a esta questão, o VMBD cita trabalhos teóricos e experimentais recentes demonstrando que, sob certas condições, resultados diagnósticos úteis podem ser obtidos baseados em *snapshots* temporais, sem a realização de simulação [36].

Medidas limitadas Os sistemas de injeção de diesel e de transmissão automática, usados no projeto VMBD, apresentam pouquíssimos sensores. Uma forma de compensar esta capacidade de observação limitada, especialmente nos sistemas embarcados, é o uso de modelos de falhas, nos quais modela-se o comportamento correto de um dispositivo e as falhas que ele pode exibir. A idéia por trás desta modelagem é melhorar a localização de falhas por meio da identificação das falhas, o que significa exonerar certos componentes através da refutação de todas as falhas que ele possa exibir.

Necessidades de tempo real dos sistemas embarcados Os requisitos computacionais e de memória das funções para diagnóstico embarcadas devem ser relativamente baixos para que estas possam ser usadas nas atuais centrais de controle eletrônicas. Uma possível

solução para garantir o atendimento a tais requisitos é a pré-compilação dos resultados do raciocínio baseado em modelos, especialmente nos casos onde uma computação intensiva seja necessária, para serem usados em sistemas embarcados.

Modelos adequados para diagnóstico Existem muitas formas de modelos para projeto, simulação e fabricação em uso, mas estes modelos não são necessariamente apropriados para o diagnóstico de falhas. O fato de que o modelo do comportamento de um dispositivo sendo diagnosticado deve capturar explicitamente o conhecimento sobre o comportamento do dispositivo é o ponto chave para o sucesso de um sistema para diagnóstico automático de falhas. Desde que qualquer nova informação tem um custo (um sensor extra para sistemas embarcados ou uma nova medida nos sistemas *off-board*), informações redundantes devem ser evitadas tanto quanto informações incompletas. Por exemplo, mesmo que as falhas na unidade de controle não sejam consideradas, o seu comportamento deve ser modelado em alguns casos, desde que ela faz parte do circuito de realimentação dos sensores e atuadores.

Vários formalismos para representação de comportamento foram investigados no projeto VMDB. Modelos de comportamento quantitativos foram usados no sistema de transmissão automática, com o objetivo de obter-se uma percepção profunda dos casos de falhas e de suas manifestações dinâmicas. Por outro lado, descrições qualitativas refletem a natureza das observações disponíveis (por exemplo, “fumaça preta”) para os sistemas de injeção de diesel. Este tipo de descrição cobre classes de comportamento e ajudam a manter a biblioteca de fragmentos de modelos em um tamanho gerenciável.

3.4.4 Processo diagnóstico

O projeto VMDB definiu uma arquitetura de máquina de diagnóstico para o domínio automobilístico usando a abordagem baseada em consistência, definida nas seções anteriores. Esta máquina de diagnóstico executa um processo diagnóstico composto pelos seguintes passos:

1. Entra-se com as observações do comportamento atual do sistema;

2. Conclusões sobre parâmetros e variáveis do sistema são calculadas a partir do modelo do dispositivo;
3. Se uma contradição é detectada, isto é, conclusões e observações conflitantes para um parâmetro ou variável, o conjunto de componentes envolvidos no conflito indicam os possíveis componentes com desvio de comportamento. Isto pode ser determinado pelo sistema diagnóstico, pois o modelo do dispositivo contém a estrutura dos constituintes do dispositivo.
4. Diagnósticos hipotéticos, ou seja, conjuntos de componentes defeituosos que explicam todas as contradições encontradas, são gerados. Estes diagnósticos correspondem aos diagnósticos candidatos do GDE;
5. Sugestões de testes discriminatórios são geradas. Isto é possível porque o modelo de comportamento revela onde os diagnósticos hipotéticos requerem características distintas de comportamento. Esta função pode ser usada para reduzir os custos, nos casos onde observações são caras, ou agir como um filtro onde a quantidade de informação é muito grande.
6. Nos casos onde modelos de comportamento defeituoso são providos, a mesma abordagem (verificação da consistência de um modelo com as observações) pode ser usada para descartar falhas particulares e concluir a correção de certos componentes, se o conjunto de falhas modeladas é considerado completo.

Capítulo 4

Sistema para diagnóstico automático de falhas em veículos automotores OBD-2

4.1 introdução

Neste capítulo, apresentamos a definição de um sistema para diagnóstico automático de falhas baseado em modelos, denominado Autodiag, cujo objetivo é facilitar o processo de manutenção dos sistemas eletrônicos dos veículos automotores equipados com a tecnologia OBD-2. O funcionamento deste sistema baseia-se no uso dos serviços diagnósticos do sistema OBD-2 para a geração de informações que comparadas com as informações produzidas por modelos da estrutura e do comportamento dos componentes reais do veículo, permitam a identificação e o diagnóstico de falhas.

O sistema Autodiag é um exemplo de sistema diagnóstico *off-board* que visa o reparo ou mais especificamente a troca dos componentes defeituosos. Dessa forma, a sua tarefa é localizar falhas no nível dos menores componentes substituíveis. Como vimos nos capítulos anteriores, os sistemas diagnósticos embarcados são eficientes na detecção de falhas, mas pouco eficientes na localização das mesmas, provendo informações insuficientes em mais de

20% dos casos, segundo pesquisa da Colorado State University [1]. Assim, o principal objetivo do sistema Autodiag é oferecer um processo para diagnóstico de falhas que incorpore as informações providas pelo sistema OBD-2 e que elimine as suas deficiências relacionadas à identificação dos componentes defeituosos.

A maioria dos complexos sistemas de um veículo automotor podem ser decompostos em componentes, cujos comportamentos e conexões determinam o comportamento dos respectivos sistemas. Além disso, os comportamentos destes componentes são bem conhecidos, uma vez que são artefatos técnicos projetados e construídos pelo homem. Isto torna o domínio automobilístico adequado para a aplicação de técnicas de diagnóstico de falhas baseadas em modelos, tal como o GDE. Dessa forma, o Autodiag é um sistema diagnóstico baseado em modelos, cuja abordagem baseia-se na aplicação do processo diagnóstico do GDE para o diagnóstico de falhas em veículos automotores e na incorporação das informações do sistema OBD-2 aos modelos dos componentes. A principal diferença do processo diagnóstico do Autodiag, em relação ao do GDE, é a possibilidade de inserção direta de sintomas (que no GDE são sempre inferidos das observações e predições). Esta alteração foi necessária para permitir a inserção dos sintomas detectados pelo sistema OBD-2 (códigos de defeito) e pelos motoristas e mecânicos.

O restante deste capítulo está organizado da seguinte maneira: a Seção 4.2 apresenta a arquitetura do sistema Autodiag, apresentando os seus principais componentes, as técnicas de modelagem a serem usadas na construção dos modelos e as características do mecanismo de inferência usado na propagação de restrições. A Seção 4.3 mostra o processo diagnóstico do Autodiag e a Seção 4.4 mostra como o processo de manutenção de um veículo automotor é influenciado pela introdução do Autodiag.

4.2 Arquitetura do sistema

O sistema Autodiag é um sistema diagnóstico baseado em modelos, cuja abordagem baseia-se na aplicação do processo diagnóstico do GDE para o diagnóstico de falhas em componentes eletrônicos dos veículos automotores. O Autodiag incorpora as informações providas

pelo sistema OBD-2 aos modelos dos componentes e caracteriza os sintomas como violações de suposições. No Autodiag, um sintoma é qualquer inconsistência entre o comportamento esperado de um dispositivo e o seu comportamento real. Estas inconsistências podem ser inferidas das diferenças entre as predições dos modelos e as observações, detectadas pelo mecanismo de inferência, ou inseridas diretamente ao processo diagnóstico pelo usuário. A inserção direta de sintomas permite o tratamento dos códigos de defeito detectados pelo sistema OBD-2, além de permitir a incorporação de sintomas percebidos por motoristas e mecânicos.

O processo de modelagem dos sintomas é semelhante ao processo de criação de regras em um sistema especialista. Entretanto, o Autodiag não relaciona sintomas com causas, mas sim com os componentes que possivelmente podem estar apresentando falhas. Assim, este processo permite apenas focalizar o processo diagnóstico nos possíveis componentes defeituosos. O diagnóstico continua sendo feito a partir dos modelos de estrutura e comportamento.

A arquitetura do sistema Autodiag (Figura 4.1) é formada pelos modelos que encapsulam o conhecimento diagnóstico do sistema a ser diagnosticado, por um mecanismo de inferência, responsável pela dedução de novas informações a partir das observações e dos modelos, e por um sistema de manutenção da verdade baseado em suposições (ATMS), usado para armazenar as deduções feitas pelo mecanismo de inferência e para auxiliar o processo diagnóstico. As subseções seguintes apresentam cada um destes elementos em detalhes.

4.2.1 Modelos de estrutura e comportamento

Os veículos automotores são um exemplo clássico de uma classe de sistemas técnicos que são montados a partir de um repositório de componentes básicos. Isto significa que o modelo de um dispositivo para este domínio pode ser montado a partir de modelos de comportamento de componentes, tal como o dispositivo real é montado a partir de um conjunto de componentes.

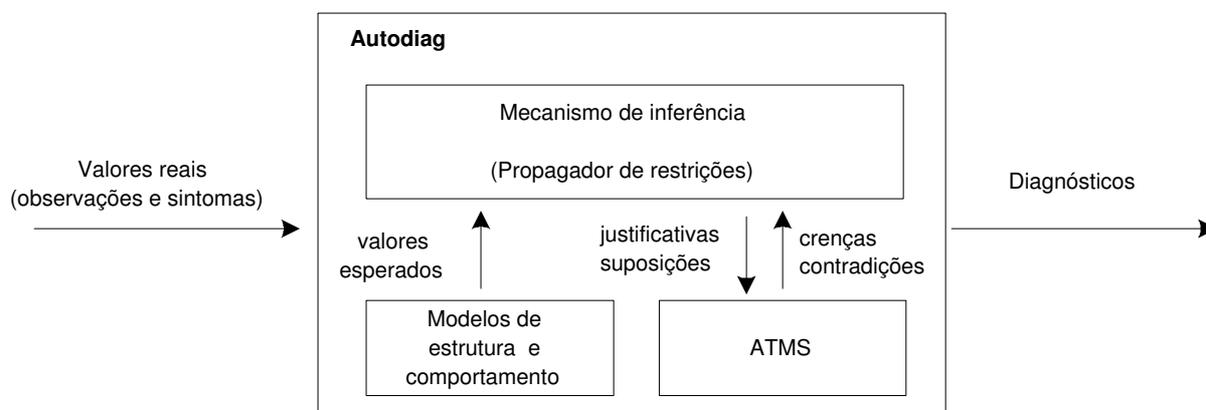


Figura 4.1: Arquitetura do Autodiag

Existem muitas formas de modelos para projeto, simulação e fabricação em uso. Entretanto tais modelos não são necessariamente apropriados para o diagnóstico de falhas. O requisito básico para que um modelo de comportamento seja usado com sucesso para o diagnóstico automático de falhas é a sua capacidade de capturar explicitamente o conhecimento sobre o comportamento do dispositivo real. Podemos dividir os tipos de modelos em duas classes principais: modelos quantitativos e modelos qualitativos. Modelos de comportamento quantitativos lidam com um conjunto de variáveis cujos valores são medidos precisamente (por exemplo, sinal de rotação da roda esquerda da frente igual a $12,5s^{-1}$), enquanto modelos qualitativos lidam com variáveis com valores qualitativos (por exemplo, sinal de rotação da roda esquerda da frente muito alto).

Como os sistemas de um veículo automotor são exemplos de sistemas dinâmicos que requerem a interação de componentes de diferentes domínios físicos (elétrico, eletrônico, hidráulico, pneumático e mecânico) [34], a construção de modelos quantitativos e a posterior predição de comportamento usando tais modelos são extremamente complicadas. Além disso, em muitos casos não é relevante (ou possível) raciocinar em termos de valores quantitativos. Outra desvantagem destes modelos é a incapacidade de tratar observações feitas pelos motoristas, tais como: “o veículo desvia para esquerda quando o freio é acionado” ou “fumaça preta”.

Por outro lado, uma boa parte das informações providas pelos fabricantes dos veículos

para auxiliar o diagnóstico de falhas são qualitativas. As informações encapsuladas pelos códigos de defeito e fornecidas nos manuais de diagnóstico de falhas são exemplos de informações qualitativas disponibilizadas. Além disso, modelos qualitativos têm sido usados com sucesso no diagnóstico de falhas de sistemas eletrônicos de automóveis [35, 36, 37]. Dessa forma, como o objetivo do Autodiag é o reparo dos veículos em oficinas mecânicas e as observações disponíveis neste ambiente são em sua maioria qualitativas, os modelos dos componentes eletrônicos a serem diagnosticados pelo Autodiag são qualitativos e as únicas variáveis e parâmetros dos componentes que precisam ser modeladas são aquelas que podem ter os seus valores medidos ou observados por um técnico de manutenção. Como algumas dessas medidas são valores quantitativos, o Autodiag oferece um mecanismo que permite o mapeamento destes valores quantitativos para valores qualitativos equivalentes.

4.2.1.1 Descrição dos sistemas a serem diagnosticados

Em [38], Havelka, Stumptner e Wotawa definem uma linguagem para descrição de modelos de sistemas a serem diagnosticados. Esta linguagem é denominada AD2L e foi projetada seguindo uma série de princípios, tais como: reusabilidade, legibilidade, independência de mecanismos de inferência, uso de modelos de diferentes níveis, etc. Segundo esta linguagem, um sistema é descrito por seus componentes e suas conexões, existindo construções para descrever os componentes, seus modos de comportamento e suas conexões. A linguagem AD2L é bastante genérica, permitindo inclusive a definição de vários modos de comportamento para um componente, além do seu comportamento correto. Entretanto, o Autodiag aceita apenas modelos qualitativos de comportamento correto dos componentes e possui requisitos, tais como a modelagem de sintomas, que não são possíveis em AD2L. Dessa forma, definimos uma linguagem para descrição dos modelos dos sistemas a serem diagnosticados pelo Autodiag, baseada na linguagem AD2L, denominada AMDL (*Autodiag Model Description Language*).

De acordo com a linguagem AMDL, o modelo de um sistema a ser diagnosticado pode ser dividido em três partes principais: a definição de domínios qualitativos, a definição de

protótipos de componentes e a definição de um único sistema, formado pela composição de instâncias de componentes cujos protótipos são conhecidos. A definição de domínios qualitativos define e nomeia um conjunto de valores qualitativos que posteriormente serão usados nas definições dos protótipos de componentes. Um exemplo de definição de um domínio qualitativo em AMDL é mostrado no Modelo 4.1, onde os termos em negrito são palavras chaves da linguagem.

Modelo 4.1 Definição de um domínio qualitativo em AMDL

```
domain min_max
  values abaixo_minimo, normal, acima_maximo;
end domain
```

A definição de protótipos de componentes define os tipos de componentes que poderão ser usados na definição do sistema. Basicamente um protótipo de componente é definido por um conjunto de portos, representando os parâmetros cujos valores definirão o comportamento do componente e serão usados nas conexões com outros componentes. Os portos são os elementos básicos do componente e os seus valores serão inferidos pelo mecanismo de inferência ou medidos (observados) por um técnico de manutenção. O comportamento do componente é definido em função dos valores de seus portos, por um conjunto de regras. Cada regra especifica uma condição na parte esquerda e uma ação na parte direita e sempre que as condições são satisfeitas as ações são executadas. Tanto a condição quanto a ação são formadas por sentenças conjuntivas envolvendo o predicado **val**, um porto e um valor. Quando estas sentenças aparecem em condições elas avaliam se o porto possui um determinado valor e quando aparecem na ação determinam que um dado valor deve ser atribuído ao porto. Além disso, a condição de uma regra contém uma sentença implícita (*ok(componente)*), representando a suposição de que o componente está funcionando corretamente, e a ação pode conter a constante **nok**, que determina que a suposição de que o componente está funcionando corretamente é falsa.

Considere o trecho de definição de um protótipo de sensor de temperatura em AMDL, mostrado no Modelo 4.2. Esta definição modela parte do comportamento de um sensor

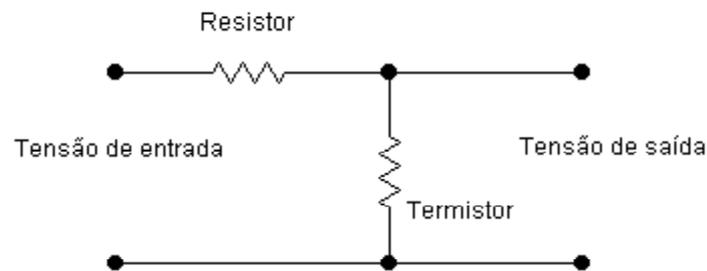


Figura 4.2: Circuito de um sensor de temperatura

de temperatura em função de parâmetros qualitativos. De maneira geral, um sensor de temperatura é composto por um dispositivo cuja resistência varia com a temperatura, usado em um circuito divisor de tensão conforme mostra a Figura 4.2. O comportamento básico deste circuito é definido pela variação da tensão de saída em função da tensão de entrada e da resistência variável do termistor. Na modelagem qualitativa deste componente, são definidos três portos, correspondentes às suas tensões de entrada e saída e à sua resistência, cujo domínio é *min_max*. A primeira regra de comportamento estabelece que se uma tensão normal for aplicada à entrada do circuito e a sua resistência variável está normal, então uma tensão normal será lida na sua saída, supondo que o sensor esteja funcionando corretamente. As duas outras regras estabelecem que se a resistência variável for observada abaixo do seu valor mínimo (em curto, por exemplo) ou acima do seu valor máximo (circuito aberto, por exemplo), a suposição de que o componente está funcionando corretamente é falsa.

Um sistema em AMDL é uma coleção de componentes interconectados, cujo comportamento é definido pelo comportamento dos componentes. Um sistema pode conter ainda as definições dos sintomas que poderão ser inseridos diretamente ao processo de diagnóstico de falhas e mapeamentos de valores quantitativos para valores qualitativos equivalentes. Na definição dos sintomas reconhecidos pelo sistema, os códigos de defeito obtidos do sistema OBD-2 e demais sintomas percebidos pelo motorista ou mecânico são descritos. Na definição dos componentes do sistema é feita a instanciação dos protótipos de componen-

Modelo 4.2 Protótipo de um sensor de temperatura em AMDL

```
prototype p_sensor_temperatura
  ports
    tensao_entrada, tensao_saida, resistencia: min_max;
  end ports
  rules
    val(tensao_entrada, normal), val(resistencia, normal) =: val(tensao_saida, normal);
    val(resistencia, abaixo_minimo) =: nok;
    val(resistencia, acima_maximo) =: nok;
  end rules
end prototype
```

tes definidos anteriormente, sendo possível ainda a atribuição de mapeamentos de valores quantitativos para valores qualitativos para os portos do componente e a declaração dos sintomas para os quais o componente deve ser considerado como suspeito. Atualmente, o único mapeamento possível é o mapeamento de valor qualitativo para uma faixa de valores quantitativos. Para finalizar, a descrição do sistema contém a definição das conexões entre os seus componentes. Cada conexão conecta portos de dois componentes distintos, indicando o caminho a ser seguido pelos valores durante a propagação. O Modelo 4.3 apresenta um trecho da descrição de um sistema bastante simples, formado por uma bateria de 5 Volts e dois sensores de temperatura.

4.2.1.2 Inclusão das informações providas pelo sistema OBD-2

As informações providas pelo sistema OBD-2 são inseridas nos modelos dos componentes do sistema a ser diagnosticado pelo Autodiag. Os códigos de defeito são inseridos diretamente nas declarações dos sintomas reconhecidos pelo sistema e nas declarações dos sintomas dos componentes. Assim, sempre que um sintoma é inserido, as suposições de que os componentes suspeitos estão funcionando corretamente formam um novo conflito.

As demais informações providas pelo sistema OBD-2, tais como a leitura dos sensores, são mapeadas através dos portos dos componentes, tal como as tensões dos sensores de temperatura no Modelo 4.2.

Modelo 4.3 Modelo de sistema simples em AMDL

```
domain min_max
  values abaixo_minimo, normal, acima_maximo;
end domain
prototype p_sensor_temperatura
  ports
    tensao_entrada, tensao_saida, resistencia: min_max;
  end ports
  rules
    val(tensao_entrada, normal), val(resistencia, normal) =: val(tensao_saida, normal);
    val(resistencia, abaixo_minimo) =: nok;
    val(resistencia, acima_maximo) =: nok;
  end rules
end prototype
prototype p_bateria
  ports
    carga, tensao_saida: min_max;
  end ports
  rules
    val(carga, normal) =: val(tensao_saida, normal);
    val(carga, abaixo_minimo) =: nok;
  end rules
end prototype
system sistema_simples
  symptom P0110 “Sensor de temperatura do ar irregular”;
  symptom P0115 “Sensor de temperatura da água irregular”;
  component sensor_temperatura_agua: p_sensor_temperatura
    name “Sensor de temperatura da água”;
    symptoms P0115;
    map resistencia normal 25 100;
  end component
  component sensor_temperatura_ar: p_sensor_temperatura
    name “Sensor de temperatura do ar”;
    symptoms P0110;
    map resistencia normal 50 120;
  end component
  component bateria: p_bateria
    name “Bateria 12V”;
  end component
  connections
    bateria(tensao_saida) -> sensor_temperatura_agua(tensao_entrada);
    bateria(tensao_saida) -> sensor_temperatura_ar(tensao_entrada);
  end connections
end system
```

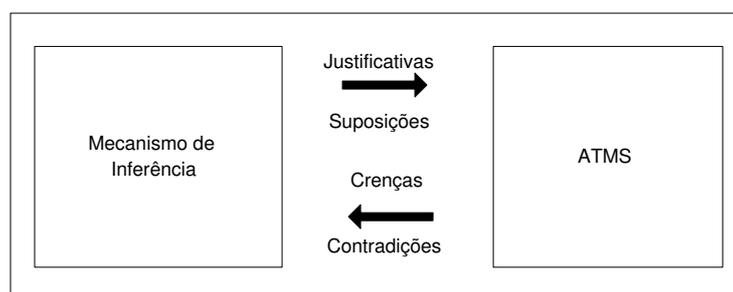


Figura 4.3: Arquitetura do sistema diagnóstico

4.2.2 Mecanismo de inferência e ATMS

O mecanismo de inferência do Autodiag é formado por um propagador de restrições, tal como no GDE. A propagação de restrições opera através da avaliação das regras de comportamento definidas nos modelos dos componentes, atuando sobre os valores dos seus portos. A operação básica deste mecanismo é a avaliação das condições de todas as regras definidas pelos componentes de um sistema e a execução das respectivas ações, quando as condições forem satisfeitas. A execução de uma regra gera novos valores para os portos de um componente. Em seguida, estes novos valores são propagados através da estrutura do sistema e podem ocasionar a execução de outras regras e a dedução de outros valores.

O mecanismo de inferência atua em conjunto com um sistema de manutenção da verdade baseado em suposições (ATMS)[31]. Um ATMS é uma facilidade geral de Inteligência Artificial para a solução de problemas que ajuda o mecanismo de inferência a manipular conveniente e eficientemente suposições. O uso de um ATMS capacita o Autodiag a adicionar fatos hipotéticos (suposições) ao seu banco de dados e removê-los futuramente, caso seja verificado que não são verdadeiros. A observação fundamental no uso de um ATMS é que a arquitetura de um sistema diagnóstico qualquer pode ser decomposta em duas partes: um mecanismo de inferência e um ATMS.

Esta divisão de preocupações, mostrada na Figura 4.3, permite que o mecanismo de inferência seja focado na derivação de inferências no domínio desejado, enquanto o ATMS focaliza nas justificativas, nas suposições e nos contextos. O resultado é um sistema di-

agnóstico muito mais simples e eficiente. Durante uma sessão de diagnóstico, o mecanismo de inferência e o ATMS interagem continuamente em um protocolo bem definido. Cada inferência feita pelo mecanismo de inferência é comunicada ao ATMS como uma justificativa. As justificativas registradas pelo ATMS permitem que o sistema diagnóstico possua as seguintes capacidades:

1. Geração de explicações, através do rastreamento das justificativas de uma informação;
2. Recuperação de inconsistências, através da manipulação de contextos e suposições.
3. Armazenamento de uma cache de inferências;
4. Registro das suposições usadas em cada informação no banco de dados, através da manipulação das justificativas para a determinação das suposições.

4.3 Processo diagnóstico

O processo diagnóstico do Autodiag é derivado do processo diagnóstico do GDE (Seção 3.3) e baseia-se na caracterização das diferenças entre modelos e dispositivos (inconsistências) como violações de suposições. Este processo contém quatro fases principais, organizadas em um ciclo (Figura 3.2) que se repete até que o diagnóstico para as falhas seja encontrado. Desta forma, o processo diagnóstico do Autodiag também é dividido em quatro etapas e as seções seguintes apresentam estas etapas, destacando os pontos que foram alterados.

4.3.1 Predição de comportamento

Nesta fase, o mecanismo de inferência utiliza os modelos de funcionamento correto dos componentes (Seção 4.2.1) para inferir novos valores para os portos dos mesmos. As observações e os valores inferidos são propagados através da estrutura do modelo, passando de um componente para outro, de acordo com as conexões do sistema. A cada passo da propagação, as condições das regras de comportamento dos componentes são avaliadas

e se forem satisfeitas, suas respectivas ações são executadas e geram novos valores. A propagação continua até que não seja possível inferir mais valores.

Para cada valor gerado, para cada porto dos componentes, o mecanismo de inferência adiciona uma proposição no ATMS, registrando as suas dependências e os ambientes (conjuntos de suposições) nos quais os valores são consistentes. A cada proposição inserida, o ATMS atualiza as informações das demais proposições afetadas pela nova informação.

4.3.2 Detecção de conflitos

Ao contrário do GDE, o Autodiag permite a inserção direta de sintomas durante uma sessão de diagnóstico. Os códigos de defeito do sistema OBD-2 e outros sintomas visíveis aos motoristas e mecânicos podem ser modelados através dos sintomas do modelo do respectivo sistema. Como o modelo define os componentes suspeitos de um determinado sintoma, a informação de um sintoma é tratada como uma contradição e o Autodiag cria um novo conflito formado pelas suposições de que cada um dos componentes suspeitos estão funcionando corretamente. Assim, um sintoma indica que pelo menos um dos componentes suspeitos está defeituoso.

Considerando o Modelo 4.3, ao inserirmos o sintoma “P0110”, o Autodiag adiciona ao ATMS, a contradição $\text{sintoma}(\text{“P0110”}) \Rightarrow \perp$, onde a constante \perp representa uma contradição. O rótulo desta contradição é composto pelo ambiente formado pela suposição de que o sensor de temperatura do ar está funcionando corretamente. Após a inserção desta contradição, o ATMS identifica um novo conflito, formado pelos ambientes do rótulo da contradição. Neste caso, o conflito seria formado apenas pela suposição de que o sensor de temperatura do ar está funcionando corretamente.

Os sintomas também são detectadas durante a propagação das observações através da estrutura do modelo do sistema. Tal como no GDE, sempre que um porto assume valores contraditórios em um mesmo contexto, o conjunto de suposições que constitui tal contexto deve ser identificado como inconsistente e formar um novo conflito.

Como vimos na Seção 3.3.2, o ATMS calcula e mantém um banco de dados de ambientes

inconsistentes mínimos. Dessa forma, a identificação de conflitos mínimos é trivial para o Autodiag, sendo determinada simplesmente pelo conjunto de ambientes inconsistentes mínimos mantido pelo ATMS.

4.3.3 Geração de diagnósticos candidatos

No Autodiag, um diagnóstico candidato é uma hipótese particular que explica as diferenças entre os dispositivos e seus respectivos modelos e os sintomas inseridos diretamente. Cada candidato é representado por um conjunto de suposições, onde as suposições explicitamente mencionadas são falsas e as demais são verdadeiras, construído de tal maneira que se todas as suas suposições forem falsas, então todos os sintomas conhecidos são explicados. No caso do domínio dos componentes eletrônicos de um automóvel, um candidato representa um conjunto de componentes defeituosos. Além disso, o espaço de candidatos é representado através dos candidatos mínimos, ou seja, dos menores conjuntos de componentes defeituosos que conseguem explicar todos os sintomas existentes.

O processo de geração de candidatos mínimos é incremental, ou seja, à medida que as observações (medidas ou sintomas) são inseridas e os novos conflitos mínimos são identificados, o espaço de candidatos mínimos é refinado de acordo com o processo definido na Seção 3.3.2.

4.3.4 Discriminação de diagnósticos candidatos

Para reduzir o conjunto de diagnósticos candidatos o Autodiag propõe a execução de medidas, cujo objetivo é a geração de novas observações que permitam discriminar os candidatos. A cada passo de execução desta fase busca-se identificar o porto que ainda não tenha sido medido, cuja medida mais reduza o número de candidatos, ou seja, que na média nos leve à descoberta do diagnóstico atual em um número mínimo de passos.

O Autodiag utiliza a estratégia baseada em entropia mínima, usada pelo GDE, para discriminar os diagnósticos candidatos. A cada passo a entropia esperada é calculada a partir das novas probabilidades dos candidatos, calculadas a partir das suas probabilidades

atuais e do resultado hipotético de $x_i = v_{ik}$. As probabilidades iniciais dos candidatos são calculadas de probabilidades iniciais de falha dos componentes, obtidas dos seus fabricantes ou através de dados empíricos.

Estratégias baseadas na entropia mínima podem ser usadas mesmo que as probabilidades de falha dos componentes não sejam conhecidas. Em [39, 30], Forbus e de Kleer apresentam um método baseado na entropia mínima que considera uma probabilidade de falha igual para todos os componentes e simplifica muito os cálculos para se encontrar a melhor medida. Segundo este método, o melhor porto a ser medido é aquele que minimiza

$$\sum c_i \log c_i, \quad (4.1)$$

onde c_i é o número de diagnósticos candidatos que predizem um valor para o i -ésimo porto.

4.4 Processo de manutenção de veículos automotores

O processo de manutenção de veículos automotores apresentado na Seção 2.6 estabelece um método para o diagnóstico e reparo de falhas composto por seis etapas [26]:

1. Coleta de evidências;
2. Análise de evidências;
3. Localização da falha;
4. Localização e solução da causa da falha;
5. Reparação do dispositivo defeituoso;
6. Teste do sistema e verificação do reparo realizado.

O sistema Autodiag é inserido neste processo conforme mostra a Figura 4.4. Inicialmente o sistema Autodiag é configurado de tal maneira que carregue os modelos do sistema a ser diagnosticado e inicie uma sessão de diagnóstico. Em seguida, a coleta de evidências é feita através da ferramenta de diagnóstico OBD-2, que coleta os códigos de defeito e demais parâmetros providos pelo sistema OBD-2, presentes no modelo. O mecânico pode

ainda inserir como evidência qualquer dos sintomas aceitos pelo modelo e que não são providos pelo sistema OBD-2 (os sintomas detectados pelo motorista, por exemplo). Todas as evidências são fornecidas ao Autodiag.

O Autodiag executa o seu processo de diagnóstico de falhas, que corresponde às etapas de análise de evidências, localização das falhas e de suas causas. Durante este processo, o Autodiag pode solicitar a realização da medida de parâmetros, que é feita pelo mecânico. Ao final deste processo, os componentes defeituosos são identificados.

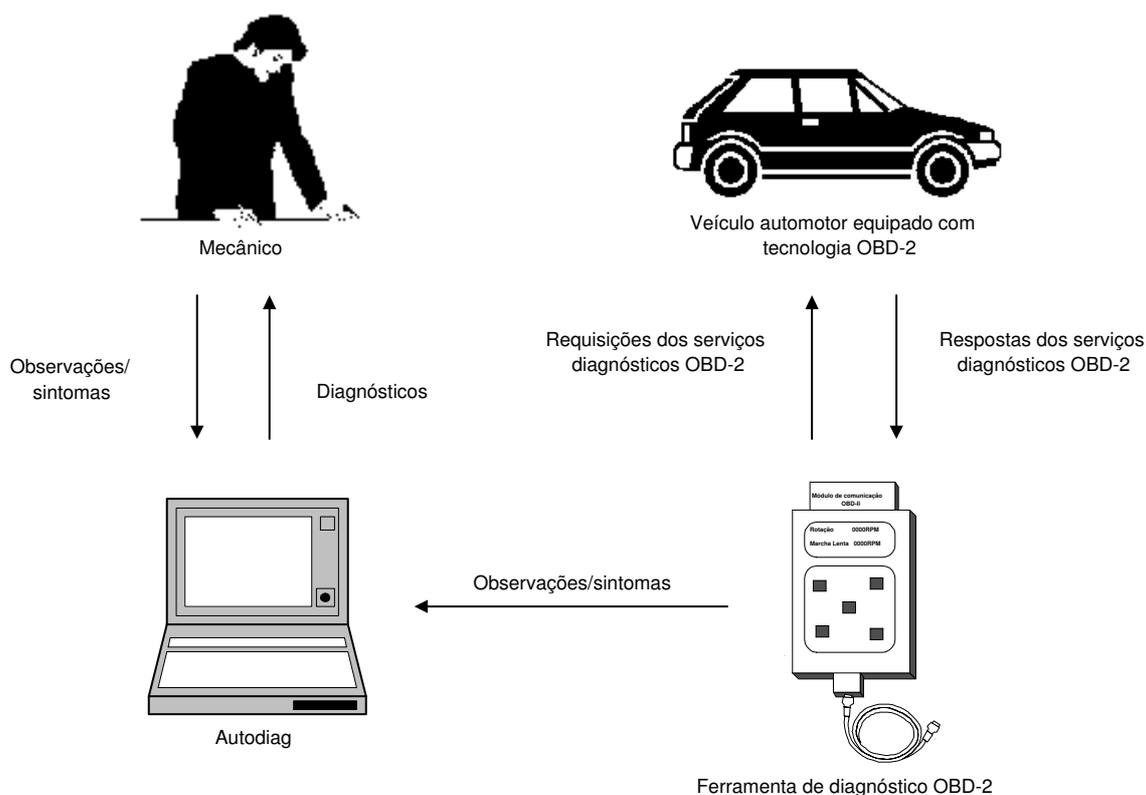


Figura 4.4: Inserção do Autodiag ao processo de manutenção

Para finalizar o processo de manutenção, o reparo dos componentes eletrônicos defeituosos e o teste do veículo são feitos pelo mecânico. Caso o veículo ainda apresente algum defeito, inicializa-se uma nova sessão de diagnóstico.

Capítulo 5

Implementação do sistema e resultados

5.1 Introdução

Neste capítulo, apresentamos o projeto e implementação do sistema para diagnóstico automático de falhas em veículos automotores OBD-2 (Autodiag). O projeto do sistema foi feito de acordo com as recomendações do processo de desenvolvimento de software PRAXIS e implementado na linguagem Java. Além disso, o sistema foi projetado de tal maneira que os modelos dos sistemas a serem diagnosticados são carregados a partir de uma biblioteca de modelos armazenados em arquivos XML.

O restante deste capítulo está organizado da seguinte maneira: a Seção 5.2 apresenta o projeto e implementação do Autodiag, descrevendo a sua arquitetura e seus componentes. a Seção 5.3 apresenta um modelo simplificado do sistema de injeção eletrônica de combustível, feito para a avaliação do sistema.

5.2 Implementação

Esta seção apresenta o desenho do sistema para diagnóstico de falhas em veículos automotores equipados com a tecnologia OBD-2, denominado Autodiag. O projeto foi feito de acordo com as recomendações do processo de desenvolvimento de software PRAXIS [40] e implementado na linguagem Java [41].

Em linhas gerais, o Autodiag foi projetado e implementado de tal maneira que seu funcionamento fosse independente do processo de criação dos modelos dos sistemas a serem diagnosticados. Dessa forma, ele implementa classes correspondentes ao mecanismo de inferência, ao ATMS e a um sistema genérico a ser diagnosticado (composto por componentes, portos, regras, códigos de defeito, conexões e etc.). Além disso, o Autodiag apresenta classes que personalizam este sistema genérico a ser diagnosticado a partir das descrições de modelos de sistemas armazenados em arquivos XML. A arquitetura da implementação, mostrada na Figura 5.1, pode ser dividida em seis módulos principais:

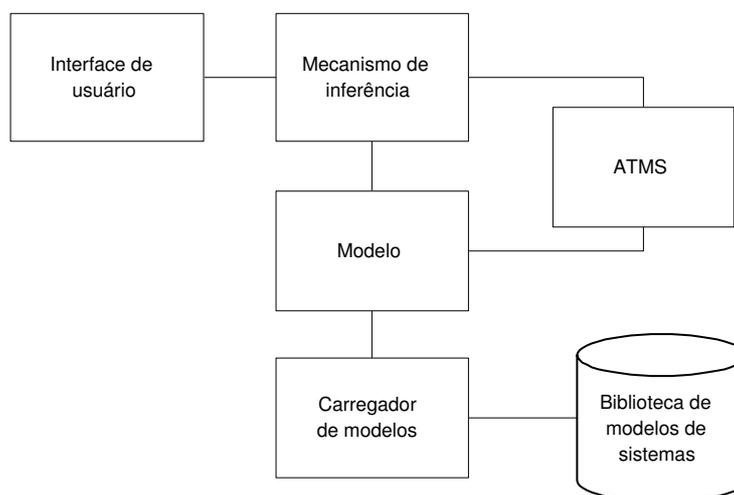


Figura 5.1: Arquitetura de implementação

Interface de usuário A interface de usuário permite o carregamento do modelo do sistema a ser diagnosticado e a execução de sessões de diagnóstico. Nestas sessões, basicamente o usuário insere observações (sintomas, códigos de defeito, medidas, etc.)

e recebe os diagnósticos, compostos por um conjunto de componentes defeituosos. A interface de usuário implementada é bastante simples, tendo sido feita apenas para permitir a avaliação do Autodiag. Entretanto, a interface de usuário é um elemento crítico para a utilização do Autodiag no ambiente de uma oficina mecânica, uma vez que grande parte dos mecânicos ainda não é muito familiarizada com o uso de computadores.

Mecanismo de inferência O mecanismo de inferência é responsável pela dedução de novos valores a partir das observações e dos modelos dos componentes do sistema sendo diagnosticado. De maneira geral este módulo, juntamente com o modelo do sistema e o ATMS, é responsável pelas quatro fases do processo diagnóstico do Autodiag: predição de comportamento, detecção de conflitos, geração de diagnósticos candidatos e busca da medida que melhor discrimine os candidatos.

Modelo As classes deste módulo são responsáveis pelo armazenamento das informações de comportamento e estrutura, presentes no modelo do sistema a ser diagnosticado. Estas informações incluem os dados do sistema, dos seus componentes (incluindo regras, portos e seus respectivos valores), dos sintomas e das conexões entre os componentes. A implementação deste módulo é composta por um conjunto de classes (Figura 5.5) que implementam elementos genéricos, tal como a classe *autodiag.model.Component*, que são personalizados a partir das descrições de sistemas armazenadas em arquivos XML.

ATMS O módulo ATMS implementa um sistema de manutenção da verdade baseado em suposições que auxilia o mecanismo de inferência durante o processo diagnóstico. As classes deste módulo (Figura 5.3) encapsulam os elementos tratados por um ATMS da seguinte maneira: premissas (observações), contradições, suposições e proposições são encapsulados pela classe *autodiag.atms.Node*; os ambientes são implementados pela classe *autodiag.atms.Environment*; as justificativas são implementadas pela classe *autodiag.atms.Justification*; a classe *autodiag.atms.Atms* implementa o ATMS pro-

priamente dito. As demais classes implementam listas e tabelas destes elementos básicos. O projeto e implementação destas classes foram baseados no código Lisp disponibilizado em [30].

Carregador de modelos Este módulo é responsável pelo carregamento do modelo do sistema a ser diagnosticado, presente na biblioteca de modelos do Autodiag. Basicamente, este módulo realiza a leitura e interpretação de um modelo descrito em XML, cria instâncias de elementos genéricos (componentes, portos, etc.) e personaliza tais instâncias com os dados do modelo.

Biblioteca de modelos A implementação do Autodiag assume a existência de uma biblioteca de modelos descritos em XML. Todo sistema a ser diagnosticado pelo Autodiag deve possuir uma descrição do seu modelo em XML, conforme definido na Seção 5.2.2. Além disso, a construção desta biblioteca de modelos pode ser feita através da implementação de ferramentas de modelagem de alto nível.

5.2.1 Modelagem UML

O processo de desenvolvimento de software PRAXIS é orientado a objetos e utiliza a linguagem de modelagem UML. Desta forma, apresentamos a seguir os diagramas de pacotes e de classes correspondentes ao desenho do Autodiag.

Em UML, os subsistemas e outros componentes são representados por pacotes lógicos de desenho. Estes pacotes lógicos são grupos de classes e outros elementos de modelagem, que apresentam fortes relacionamentos entre si (alta coesão interna) e poucos relacionamentos com elementos de outros pacotes lógicos (baixo acoplamento externo). Os pacotes lógicos do Autodiag foram definidos seguindo critérios definidos em [40] e mostrados abaixo:

- encapsular classes correlatas;
- facilitar o arquivamento e recuperação das classes;
- agrupar-se em torno das principais funções arquitetônicas.

Pacote	Conteúdo
model	Contém as classes referentes ao modelo de um sistema genérico a ser diagnosticado pelo Autodiag e à definição do “carregador” de modelos, responsável pela personalização deste sistema genérico
engine	Contém as classes referentes ao mecanismo de inferência, sendo responsável por definir a interface de acesso ao processo diagnóstico do Autodiag
atms	Contém as classes referentes ao sistema de manutenção da verdade baseado em suposições (ATMS)
ui	Contém as classes referentes à interface do usuário
xml	Contém as classes relacionadas com a carga dos modelos em XML
util	Contém classes utilitárias usadas pelos demais pacotes

Tabela 5.1: Pacotes e suas funções

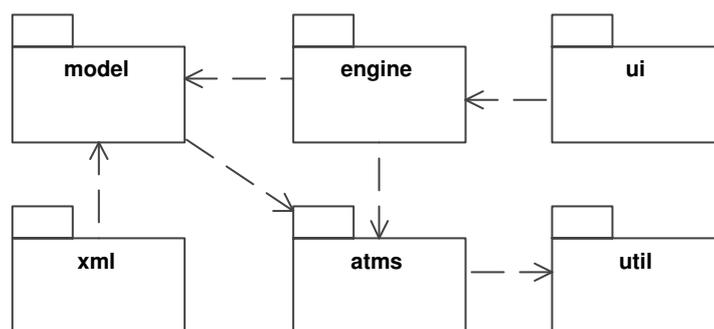


Figura 5.2: Diagrama de pacotes do Autodiag

A Figura 5.2 apresenta o diagrama de pacotes correspondente ao desenho do Autodiag. O sistema contém seis pacotes, cada um dos quais possuindo as atribuições mostradas na Tabela 5.1.

Em UML, os diagramas de classes são usados para apresentar as classes e suas associações. As associações entre as classes representam dependências semânticas entre as classes e indicam que um objeto de uma classe possui conhecimento de objetos da outra classe. As Figuras 5.3, 5.4 e 5.5 apresentam os diagramas de classes do desenho do Autodiag. A Figura 5.3 apresenta as classes e associações referentes ao pacote *atms*. A Figura 5.4 apresenta as classes referente ao pacote *engine* e suas associações com as classes dos

pacotes *atms* e *model*. A Figura 5.5 apresenta o diagrama de classes referente ao pacote *model*.

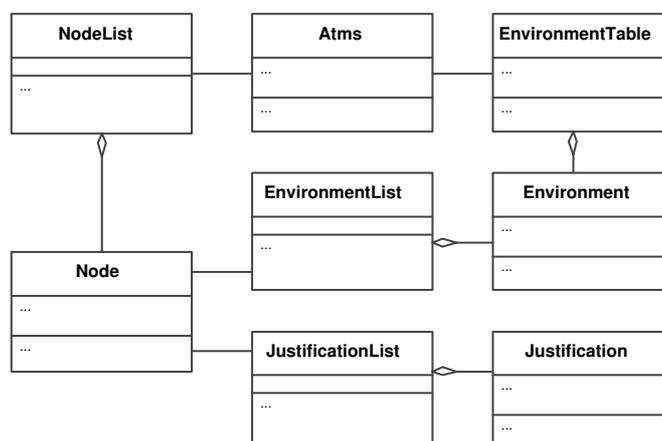


Figura 5.3: Diagrama de classes do ATMS

5.2.2 Descrição da modelagem em XML

A linguagem AMDL é adequada para a descrição dos modelos dos sistemas cujas falhas devem ser diagnosticados pelo Autodiag. Entretanto, esta descrição não é adequada para o armazenamento dos modelos na biblioteca de modelos do sistema. O armazenamento de modelos pode ser feito através de um banco de dados, de arquivos, etc. A abordagem adotada pelo Autodiag foi o armazenamento dos modelos em arquivos, usando a linguagem XML para descrever os dados armazenados.

A linguagem XML (*Extensible Markup Language*) é uma forma de representar dados padronizada e independente de sistema [42]. XML delimita os dados por *tags* que indicam o conteúdo e a estrutura dos dados, tornando possível o seu armazenamento, a execução de pesquisas, etc. Como o seu nome indica, XML é extensível, permitindo que sejam definidas novas *tags* para descrever os dados desejados. Além disso, XML permite a definição de um esquema definindo a estrutura de um tipo de documento XML. O esquema permite definir as *tags* aceitas em um documento XML, bem como a estrutura hierárquica, incluindo a

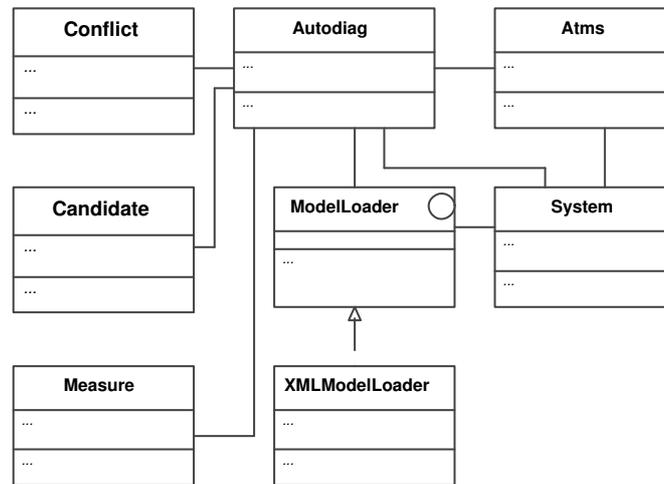


Figura 5.4: Diagrama de classes do mecanismo de inferência

ordem de aparição das *tags* e etc. A linguagem mais comum para definição de esquemas XML é *Document Type Definition* (DTD). Além disso, a linguagem XML é muito usada para o compartilhamento de dados entre aplicações, existindo bibliotecas de programação para manipulação de documentos XML, tanto para a criação, quanto para a leitura de documentos, disponíveis para várias linguagens de programação.

Estas características da linguagem XML permitem a definição de uma arquitetura onde a implementação do sistema de diagnóstico é feita de maneira completamente independente da criação dos modelos dos sistemas a serem diagnosticados. Esta arquitetura permite a implementação de ferramentas para a construção e manipulação de uma biblioteca de componentes básicos e para a geração dos modelos dos sistemas a serem diagnosticados pelo Autodiag. Dessa forma, a construção dos modelos dos sistemas pode ser feita usando-se ferramentas de modelagem especializadas, tanto de modelagem visual, quanto através de compiladores para a linguagem AMDL, conforme mostra a Figura 5.6.

Segundo esta arquitetura, as ferramentas podem compartilhar uma biblioteca de modelos de componentes (descritos em XML, ou não) para permitir a construção dos modelos de sistemas a partir dos modelos dos seus respectivos componentes. A saída destas ferramentas é um documento XML que armazena todas as informações do modelo do sistema

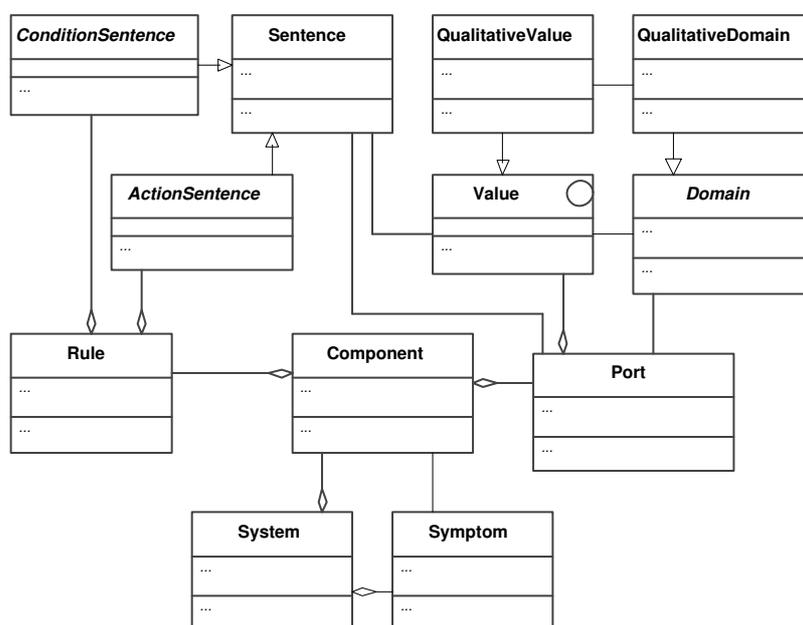


Figura 5.5: Diagrama de pacotes da modelagem

a ser diagnosticado e que será incluído à biblioteca de modelos do Autodiag. Esta abordagem possui a grande vantagem de viabilizar a reutilização dos modelos dos componentes em vários sistemas. Para descrever e armazenar os modelos em XML, definimos um tipo de documento XML, através de uma descrição *DTD* (*Document type definition*). O *DTD* define as *tags* permitidas no documento e as suas posições, definindo uma gramática que permite que o *parser* XML valide os documentos. Foram definidas tags correspondentes a todas as construções de modelagem descritas na Seção 4.2.1. O Apêndice C apresenta a descrição do tipo de documento XML definido para o Autodiag.

5.3 Avaliação do sistema

Para avaliarmos a implementação do sistema Autodiag, foi feita a modelagem qualitativa de parte do sistema de injeção e ignição eletrônica FIC EEC-V SFI de 60 pinos, presente em veículos Ford, tais como Fiesta e KA 1.0. Em linhas gerais, um sistema de injeção eletrônica

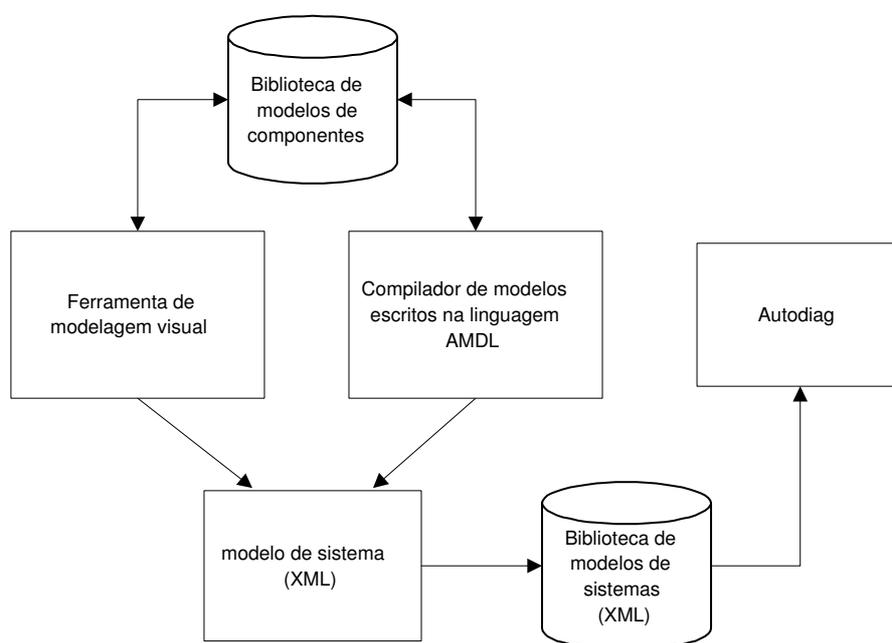


Figura 5.6: Construção de modelos para o Autodiag

de combustível é um sistema não acionado pelo motor, comandado eletronicamente, que dosa o combustível, controlando a mistura ar-combustível em função das necessidades imediatas do motor. De modo semelhante, a ignição eletrônica permite que o motor trabalhe com o seu ponto de ignição (momento da geração da fagulha que provoca a queima da mistura ar-combustível) sincronizado com as diversas condições de funcionamento do motor. A finalidade destes sistemas é dar equilíbrio de funcionamento para o motor, através de um controle rígido da mistura ar-combustível e do avanço de ignição em qualquer regime de trabalho, proporcionando maior desempenho, menor consumo, facilidade de partida a frio e a quente e, principalmente, menor emissão de gases poluentes.

Em particular, o sistema FIC EEC-V [43] é um sistema de injeção multiponto seqüencial, ou seja, ele possui um injetor de combustível para cada um dos quatro cilindros do motor e aciona um injetor de cada vez seqüencialmente. A sua estratégia de funcionamento baseia-se no cálculo do tempo de injeção (ou seja, do tempo de acionamento do injetor, que indiretamente define a quantidade de combustível injetada, uma vez que a vazão é conhecida) em função da massa de ar admitida. A massa de ar é determinada por um

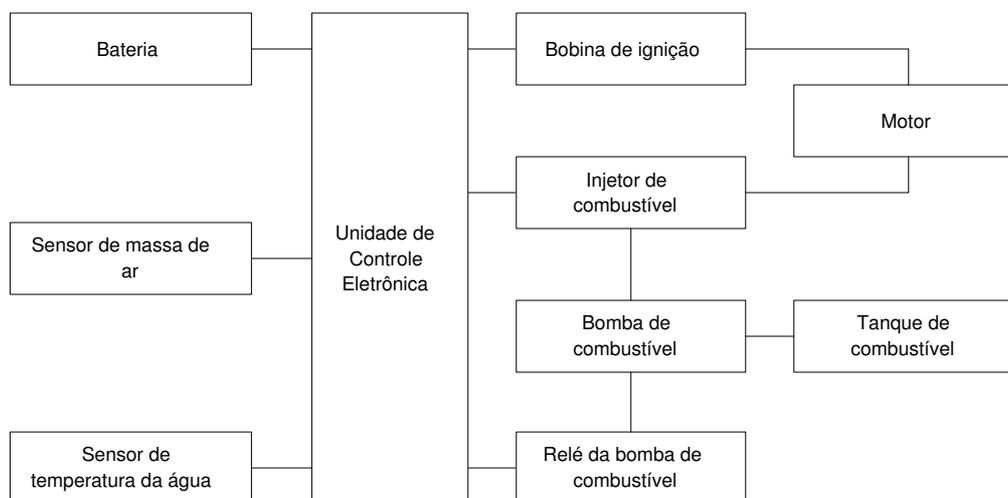


Figura 5.7: Esquemático do sistema modelado

sensor de massa de ar, que pelo seu princípio de funcionamento, corrige automaticamente as variações da pressão atmosférica, da temperatura ambiente e até da umidade relativa do ar. Este sistema utiliza ainda um sensor de temperatura da água para ajustar a injeção durante o funcionamento do motor a frio, para acionar os ventiladores de resfriamento e etc. Este sistema possui ainda uma série de sensores e atuadores que não serão considerados em nosso exemplo. A Figura 5.7 apresenta um esquemático dos componentes considerados.

A modelagem dos componentes foi feita em relação aos parâmetros operacionais definidos em [43] e limitou-se à modelagem dos códigos de defeito, das tensões e resistências esperadas e da estrutura do sistema. Além disso, os cabos que interligam os componentes também não foram modelados. O Apêndice B apresenta a descrição do sistema na linguagem AMDL. Como pode ser visto na Figura 5.7, alguns componentes não eletrônicos do veículo também foram modelados. Isto foi necessário para facilitar a modelagem dos componentes e também para permitir que o Autodiag concluísse que uma falha não está na parte eletrônica do veículo. Entretanto, isto mostra que os demais componentes não eletrônicos do veículo também podem ser modelados.

A avaliação do Autodiag foi feita através da simulação de diversas sessões de diagnóstico, nas quais eram inseridos sintomas e observações aleatórios. Após a inserção

destes dados, executava-se o processo diagnóstico do Autodiag e avaliava-se os resultados gerados. A eficiência dos diagnósticos gerados por qualquer sistema diagnóstico baseado em modelos é essencialmente uma função da qualidade dos modelos usados. No caso do nosso exemplo, o modelo é bastante simplificado, fazendo com que o Autodiag fosse capaz de diagnosticar falhas apenas nos componentes modelados, não sendo capaz de diagnosticar falhas nos cabos de conexão, por exemplo. De qualquer forma, para que seja possível uma melhor avaliação do Autodiag será necessário a modelagem de um sistema real completo, feita com a ajuda de um especialista da área (um engenheiro, técnico, etc.), e o seu uso para o diagnóstico de falhas em sistemas reais.

As Figuras 5.8, 5.9 e 5.10 apresentam as interfaces de usuário do Autodiag, durante a execução de uma sessão de diagnóstico. A Figura 5.8 mostra a interface do Autodiag antes do carregamento do modelo do nosso exemplo. Basicamente, a interface pode ser dividida em quatro regiões principais, destinadas para as informações referentes ao modelo do sistema, para as observações e sintomas inseridos pelo usuário e para os resultados (conflitos, diagnósticos e seleção de medidas) gerados pelo Autodiag. A Figura 5.9 apresenta a interface do Autodiag após o carregamento do modelo. Nesta interface, podemos ver os componentes e sintomas modelados para o sistema e os mecanismos para inserção de observações e sintomas, formado por um “menu pop-up”. A Figura 5.10 apresenta a interface do Autodiag após a execução de um passo do seu processo diagnóstico. A cada passo do seu processo diagnóstico, o Autodiag gera como resultado uma lista de conflitos mínimos, uma lista de diagnósticos candidatos e uma lista das próximas medidas que melhor distinguem os candidatos. Após a execução de cada passo, o usuário insere novas observações e sintomas e executa um novo passo, repetindo este procedimento até que o diagnóstico das falhas do sistema seja encontrado.

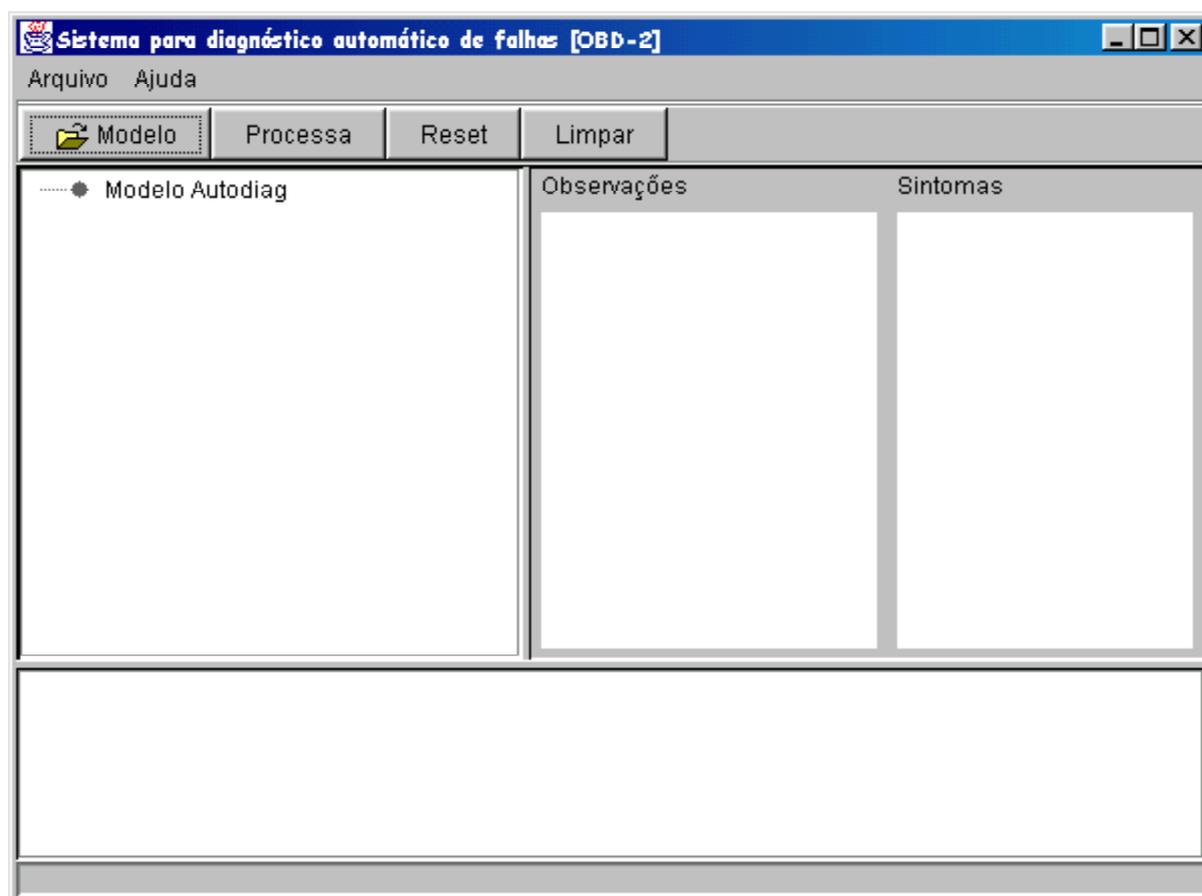


Figura 5.8: Interface do Autodiag antes do carregamento de um modelo

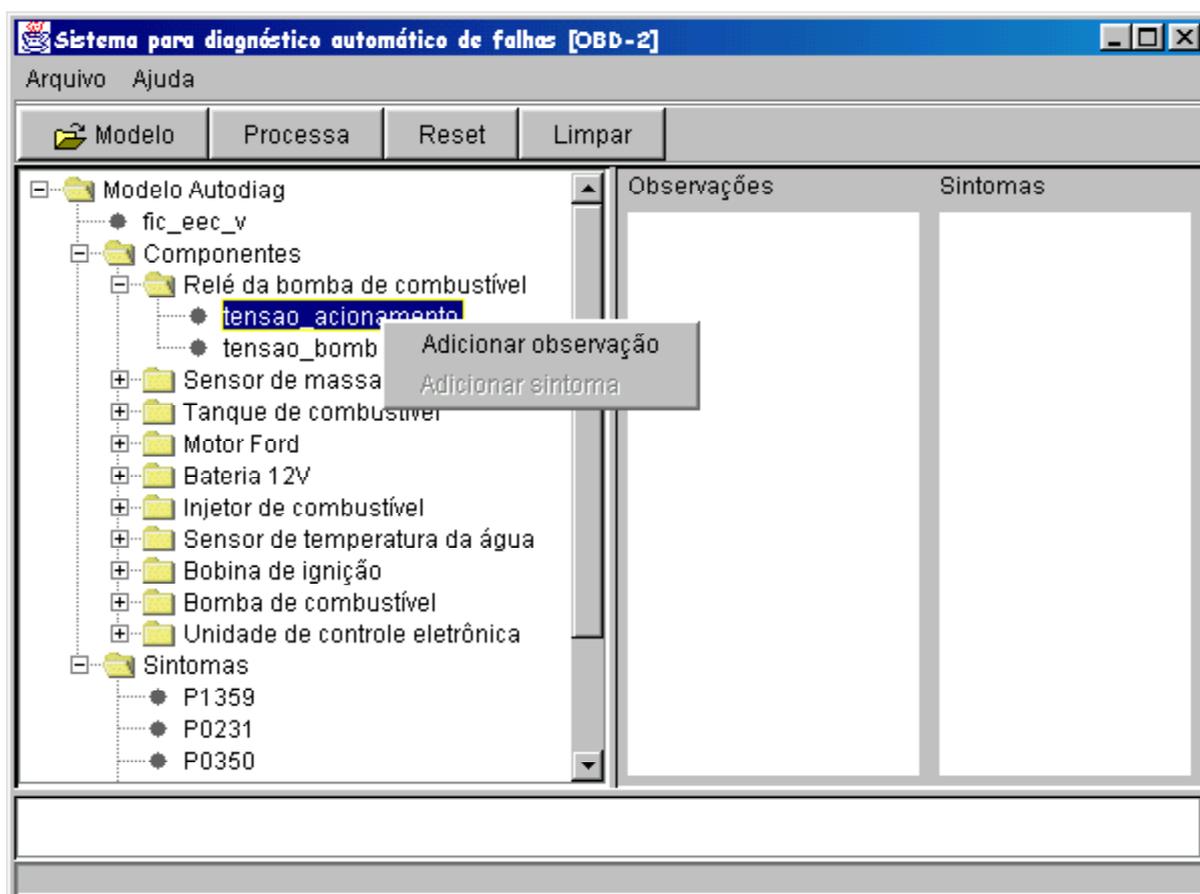


Figura 5.9: Interface do Autodiag após o carregamento de um modelo

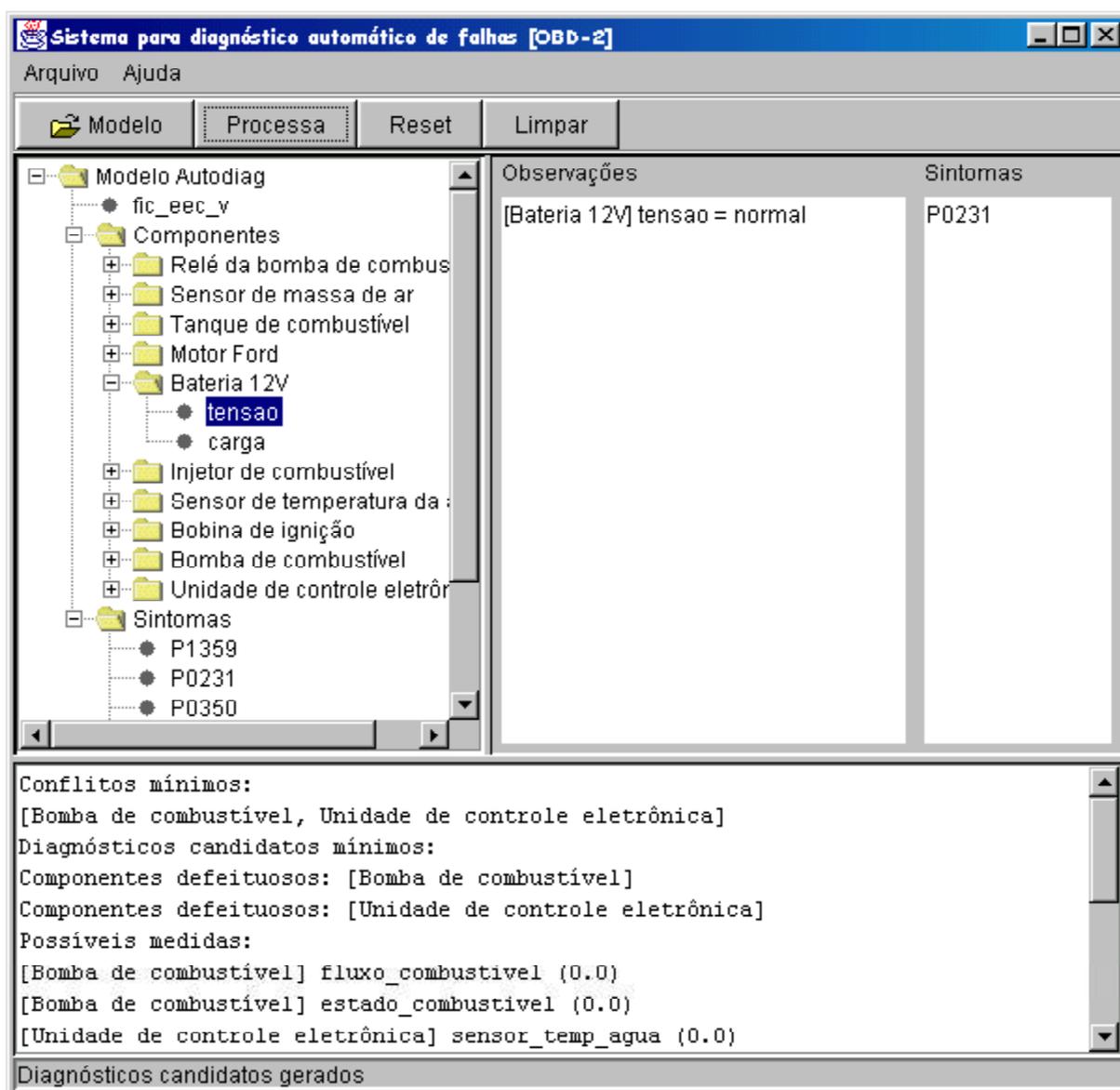


Figura 5.10: Interface do Autodiag apresentando resultados

Capítulo 6

Conclusões e trabalhos futuros

Esta dissertação de mestrado apresentou a definição e a implementação de um sistema para diagnóstico automático de falhas em veículos automotores equipados com a tecnologia OBD-2. Este sistema procura atender a crescente demanda por ferramentas que facilitem o processo de manutenção dos componentes eletrônicos dos modernos veículos automotores, que a cada nova geração incorporam mais sistemas eletrônicos.

O sistema apresentado possui diversas virtudes, entre as quais podemos destacar o tratamento de falhas causadas por múltiplos componentes defeituosos, a utilização das informações providas pelo sistema OBD-2 e a possibilidade de inserção de sintomas percebidos por motoristas e mecânicos. Além disso, o sistema define uma linguagem para modelagem de sistemas e possui uma arquitetura que permite a criação e utilização de uma biblioteca de modelos de componentes básicos. Entretanto, a construção dos modelos é uma tarefa bastante complexa, requerendo um conhecimento técnico profundo dos sistemas a serem modelados.

Apesar do Autodiag ter sido projetado para veículos automotores equipados com tecnologia OBD-2, a sua arquitetura é flexível o suficiente para que ele possa ser usado em veículos equipados com outras tecnologias de sistemas diagnósticos embarcados. Para isto, basta que modelos adequados sejam construídos. Além disso, a sua arquitetura pode ser aplicada a outros sistemas para os quais uma modelagem qualitativa seja suficiente.

Os modelos dos sistemas cujas falhas devem ser diagnosticados pelo Autodiag determinam a qualidade dos diagnósticos gerados. Dessa forma, a construção dos modelos deve ser orientada por especialistas da área, tais como engenheiros e técnicos especializados em eletrônica embarcada.

O sistema possui também alguns problemas que precisam ser tratados para que ele possa ser aplicado em sistemas complexos, formados por muitos componentes, em uma oficina mecânica. Estes problemas sugerem alguns temas para trabalhos futuros e são apresentados a seguir.

Aumento da escalabilidade Tanto o número de diagnósticos candidatos quanto o tamanho dos rótulos das proposições do ATMS crescem exponencialmente com o número de componentes. Em particular, o ATMS possui dois problemas sérios: a sua interface é difícil de controlar e ele tenta calcular todas as soluções, mesmo aquelas que são irrelevantes ou desnecessárias. Uma solução para este problema é apresentada em [33], através da definição de um foco, ou seja, de conjuntos de ambientes que devem ser investigados, para controlar a execução das regras e restringir o espaço de busca por soluções.

Aumento do poder de expressão da linguagem AMDL A linguagem de modelagem AMDL deve ser incrementada de tal maneira a aceitar outras sentenças nas regras de comportamento dos componentes. A existência de sentenças para avaliar condições de desigualdade, maior que, menor que, etc., facilitará muito a criação dos modelos. Para que isto seja possível, é necessário a definição desses operadores para os domínios qualitativos definidos nos modelos.

Seleção de medidas A estratégia de busca da melhor medida não considera os custos de realização das medidas, que no caso de um automóvel, podem ser muito diferentes. A realização de certas medidas podem requerer a desmontagem de sistemas, por exemplo. Assim, esta estratégia deve ser atualizada de tal maneira que sejam considerados os custos para a realização de cada medida.

Incorporação de procedimentos de teste e diagramas esquemáticos Para que o sistema Autodiag seja eficientemente usado em uma oficina mecânica ele deve incorporar procedimentos de teste e esquemáticos dos respectivos sistemas sendo diagnosticados. Com

estas informações, além de indicar as medidas a serem feitas, ele pode indicar como realizar a medida e onde o ponto de medida está localizado no veículo.

Integração com a ferramenta de diagnóstico OBD-2 A integração do sistema Autodiag com uma ferramenta de diagnóstico OBD-2, de forma que as informações do sistema OBD-2 possam ser coletadas sem a intervenção do usuário, também simplifica a tarefa de manutenção e pode ser implementada através de um módulo adicional, responsável pela comunicação com esta ferramenta.

Interface de usuário A interface de usuário do sistema deve ser projetada de forma que os atuais mecânicos, que nem sempre se sentem a vontade em frente a um computador, sejam capazes de utilizar o sistema.

Apêndice A

Noções de lógica matemática

Este apêndice apresenta algumas noções básicas de lógica matemática, necessárias para uma melhor compreensão deste trabalho. Estas noções limitam-se a algumas características do cálculo proposicional e da lógica de primeira ordem ou cálculo de predicados. Um estudo mais profundo deste tema pode ser encontrado em [28, 29].

A.1 Cálculo proposicional

O cálculo proposicional, ou lógica proposicional, é a lógica que lida com proposições, ou seja, sentenças declarativas (expressões de uma linguagem), para as quais é possível afirmar que sejam verdadeiras ou falsas. Alguns exemplos de proposições podem ser:

- A lua é quadrada. (falsa)
- A neve é branca. (verdadeira)
- Matemática é uma ciência. (verdadeira)
- Pintura é uma ciência. (falsa)

O cálculo proposicional apresenta uma linguagem formal precisa, especialmente dedicada ao tratamento matemático. Esta linguagem é definida conforme a Definição A.1.1.

Definição A.1.1 (Linguagem do cálculo proposicional) *A linguagem do cálculo proposicional possui um alfabeto consistindo de:*

- i. símbolos proposicionais: p, q, r, \dots ,*
- ii. conectivos: $\wedge, \vee, \rightarrow, \neg, \leftrightarrow, \perp$*
- iii. símbolos auxiliares: $(,)$.*

Os símbolos proposicionais são letras latinas minúsculas, usadas para representar as proposições, denominadas de fórmulas atômicas. Exemplos:

- A lua é quadrada: p
- A neve é branca: q

Os conectivos recebem nomes tradicionais, conforme mostrado na Tabela A.1, e permitem combinar as fórmulas atômicas entre si para formar sentenças compostas. Exemplos:

- A lua é quadrada e a neve é branca: $p \wedge q$
- A lua é quadrada ou a neve é branca: $p \vee q$
- Se a lua é quadrada então a neve é branca: $p \rightarrow q$
- A lua é quadrada se e somente se a neve é branca: $p \leftrightarrow q$
- A lua não é quadrada: $\neg p$

\wedge	e	conjunção
\vee	ou	disjunção
\rightarrow	se ..., então ...	implicação
\neg	não	negação
\leftrightarrow	se somente se	equivalência
\perp	falso	falsum, contradição

Tabela A.1: Conectivos do cálculo proposicional

Os parênteses são usados para delimitar o “alcance” dos conectivos. Exemplos:

- Se a lua é quadrada e a neve é branca então a lua não é quadrada. : $((p \wedge q) \rightarrow \neg p)$
- A lua não é quadrada se e somente se a neve é branca. : $((\neg p) \leftrightarrow q)$

Os símbolos proposicionais e o símbolo \perp designam proposições que não podem ser decompostas, chamadas de átomos ou fórmulas atômicas. As demais fórmulas proposicionais são formadas conforme a Definição A.1.2.

Definição A.1.2 (Fórmula proposicional) *As fórmulas proposicionais são formadas obedecendo as seguintes regras:*

1. *Toda fórmula atômica é uma fórmula.*
2. *Se A e B são fórmulas então $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$ e $(\neg A)$ também são fórmulas.*
3. *São fórmulas apenas as obtidas através das regras 1 e 2.*

A.2 Lógica de primeira ordem

A lógica de primeira ordem, ou cálculo de predicados, possui uma linguagem bem mais rica do que a linguagem do cálculo proposicional e tem várias aplicações importantes não só para matemáticos e filósofos como também para estudantes de Ciência da Computação.

Existem vários tipos de argumentos que, apesar de válidos, não podem ser justificados com os recursos do cálculo proposicional. A verificação da validade destes argumentos, tais como o argumento do Exemplo A.2.1, nos leva não só ao significado dos conectivos lógicos, mas também ao significado de expressões como *todo*, *algum*, *qualquer*, etc.

Para que possamos tornar a estrutura das sentenças complexas que representam tais argumentos mais transparente, é necessário a introdução de novos símbolos na linguagem do cálculo proposicional, obtendo-se a linguagem da lógica de primeira ordem. Nesta nova

Exemplo A.2.1 Argumento não justificado pelo cálculo proposicional

Todos os humanos são racionais.
Alguns animais são humanos.
Portanto, alguns animais são racionais.

linguagem temos, além dos conectivos do cálculo proposicional e os parênteses, os seguintes novos símbolos:

- variáveis: x, y, z, \dots
- constantes: a, b, c, \dots
- símbolos de predicados: P, Q, R, S, \dots
- quantificadores: \forall (universal) , \exists (existencial)
- termos: as variáveis e as constantes são designadas pelo nome genérico de termos, designados por t_1, t_2, \dots, t_n .

As variáveis representam objetos que não estão identificados no universo considerado (alguém, algo, etc.). As constantes representam objetos identificados do universo (João, o ponto A, etc.). Os símbolos de predicados representam propriedades ou relações entre os objetos do universo. Exemplos:

- Maria é inteligente: $I(m)$, onde m representa Maria e I a propriedade “ser inteligente”.
- Alguém gosta de Maria: $G(x, m)$, onde G representa a propriedade “gostar de”, m Maria e x “alguém”.

De modo geral temos:

- $P(x)$: significa que x possui a propriedade P .
- $(\forall x)P(x)$: significa que a propriedade P é válida para todos os valores de x .

- $(\exists x)P(x)$: significa que no mínimo um valor de x possui a propriedade P .

Os símbolos de predicados são unários, binários ou n -ários, conforme o número de objetos relacionados pela propriedade que eles representam. Esta característica é representada pela adição de um peso ao símbolo do predicado. Desta forma, atribuímos ao símbolo de um predicado os pesos $1, 2, \dots, n$, conforme o número de objetos envolvidos seja um, dois ou mais. As fórmulas mais simples da lógica de primeira ordem são chamadas de fórmulas atômicas, definidas conforme a Definição A.2.1. As demais fórmulas são formadas conforme a Definição A.2.2

Definição A.2.1 (Fórmulas atômicas) *Se P é um símbolo de predicado de peso n e t_1, t_2, \dots, t_n são termos, então $P(t_1, t_2, \dots, t_n)$ é uma fórmula atômica.*

Definição A.2.2 (Fórmulas da lógica de primeira ordem) *As fórmulas da lógica de primeira ordem são formadas de acordo com as seguintes regras:*

1. *Toda fórmula atômica é uma fórmula em lógica de primeira ordem.*
2. *Se α e β são fórmulas, então $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$ e $(\alpha \leftrightarrow \beta)$ são fórmulas.*
3. *Se α é uma fórmula e x é uma variável então $(\forall x)\alpha$ e $(\exists x)\alpha$ são fórmulas.*
4. *Todas as fórmulas na lógica de primeira ordem são dadas pelas regras 1, 2 e 3.*

Apêndice B

Modelo AMDL

Este apêndice apresenta a descrição em AMDL do modelo simplificado do sistema de injeção de combustível e ignição, implementado para avaliar o Autodiag.

B.1 Modelagem do sistema de injeção em AMDL

```
domain min_max
  values abaixo_minimo, normal, acima_maximo;
end domain

domain normal_anormal
  values normal, anormal;
end domain

prototype p_rele_bomba
  ports
    tensao_acionamento, tensao_bomba: min_max;
  end ports

  rules
    val(tensao_acionamento, normal) =: val(tensao_bomba, normal);
  end rules
end prototype

prototype p_motor
  ports
    estado_motor, ignicao, injecao: normal_anormal;
```

```
end ports
rules
  val(ignicao, normal), val(injecao, normal) =: val(estado_motor, normal);
  val(ignicao, anormal), val(injecao, normal) =: val(estado_motor, anormal);
  val(ignicao, normal), val(injecao, anormal) =: val(estado_motor, anormal);
  val(estado_motor, anormal), val(injecao, normal) =: val(ignicao, anormal);
  val(estado_motor, anormal), val(ignicao, normal) =: val(injecao, anormal);
end rules
end prototype

prototype p_bateria
  ports
    carga, tensao: normal_anormal;
  end ports
  rules
    val(carga, normal) =: val(tensao, normal);
    val(carga, anormal) =: nok;
  end rules
end prototype

prototype p_sensor_temperatura
  ports
    tensao_entrada, resistencia: min_max;
    tensao_saida: normal_anormal;
  end ports

  rules
    val(tensao_entrada, normal), val(resistencia, normal) =:
      val(tensao_saida, normal);
    val(tensao_entrada, abaixo_minimo) =: val(tensao_saida, anormal);
    val(tensao_entrada, acima_maximo) =: val(tensao_saida, anormal);
    val(resistencia, abaixo_minimo) =: nok;
    val(resistencia, acima_maximo) =: nok;
  end rules
end prototype

prototype p_sensor_massa_ar
  ports
    tensao_entrada, resistencia: min_max;
    tensao_saida: normal_anormal;
  end ports

  rules
    val(tensao_entrada, normal), val(resistencia, normal) =:
      val(tensao_saida, normal);
```

```
    val(tensao_entrada, abaixo_minimo) =: val(tensao_saida, anormal);
    val(tensao_entrada, acima_maximo) =: val(tensao_saida, anormal);
    val(resistencia, abaixo_minimo) =: nok;
    val(resistencia, acima_maximo) =: nok;
  end rules
end prototype

prototype p_bomba_combustivel
  ports
    tensao_entrada: min_max;
    fluxo_combustivel, estado_combustivel: normal_anormal;
  end ports

  rules
    val(tensao_entrada, normal), val(estado_combustivel, normal) =:
      (fluxo_combustivel, normal);
    val(tensao_entrada, abaixo_minimo) =: (fluxo_combustivel, anormal);
    val(tensao_entrada, acima_maximo) =: (fluxo_combustivel, anormal);
    val(estado_combustivel, anormal) =: (fluxo_combustivel, anormal);
  end rules
end prototype

prototype p_injetor
  ports
    tensao_entrada: min_max;
    fluxo_combustivel, tempo_injecao, injecao_combustivel: normal_anormal;
  end ports

  rules
    val(tensao_entrada, normal), val(fluxo_combustivel, normal),
      val(tempo_injecao, normal) =: (injecao_combustivel, normal);
    val(tensao_entrada, abaixo_minimo) =: (injecao_combustivel, anormal);
    val(tensao_entrada, acima_maximo) =: (injecao_combustivel, anormal);
    val(tempo_injecao, anormal) =: (injecao_combustivel, anormal);
    val(fluxo_combustivel, anormal) =: (injecao_combustivel, anormal);
  end rules
end prototype

prototype p_bobina_ignicao
  ports
    tensao_entrada, resistencia_primario, resistencia_secundario: min_max;
    avanco_ignicao, ignicao: normal_anormal;
  end ports
```

```
rules
  val(tensao_entrada, normal), val(avanco_ignicao, normal),
    val(resistencia_primario, normal), val(resistencia_secundario, normal) =:
      val(ignicao, normal);
  val(resistencia_primario, abaixo_minimo) =: nok;
  val(resistencia_primario, acima_maximo) =: nok;
  val(resistencia_secundario, abaixo_minimo) =: nok;
  val(resistencia_secundario, acima_maximo) =: nok;
end rules
end prototype

prototype p_tanque_combustivel
ports
  estado_tanque, estado_combustivel: normal_anormal;
end ports

rules
  val(estado_tanque, normal) =: val(estado_combustivel, normal);
  val(estado_tanque, anormal) =: nok;
end rules
end prototype

prototype p_uce
ports
  sensor_temp_agua, sensor_massa_ar, tempo_injecao,
  avanco_ignicao: normal_anormal;
  tensao_bateria, tensao_sensor_temp_agua, tensao_sensor_massa_ar,
  tensao_bobina_ignicao, tensao_rele_bomba, tensao_injetor: min_max;
end ports

rules
  val(tensao_bateria, normal) := val(tensao_sensor_temp_agua, normal),
    val(tensao_sensor_massa_ar, normal), val(tensao_bobina_ignicao, normal),
    val(tensao_rele_bomba, normal), val(tensao_injetor, normal),
    val(avanco_ignicao, normal);
  val(sensor_temp_agua, normal), val(sensor_massa_ar, normal) :=
    val(tempo_injecao, normal);
end rules
end prototype

system fic_eec_v

symptom P0101 "Sensor MAF";
symptom P0115 "Sensor de temperatura da água irregular";
symptom P0117 "Tensão baixa no sensor de temperatura da água";
```

```
symptom P0230 "Primário da bomba";
symptom P0231 "Secundário da bomba";
symptom P0350 "Primário da bobina";
symptom P1231 "Relé da bomba";
symptom P1359 "Falha na ignição";

component rele_bomba: p_rele_bomba
  name "Relé da bomba de combustível";
  symptoms P1231;
  map tensao_acionamento normal 4.5 5.5;
  map tensao_acionamento abaixo_minimo 0 4.4;
  map tensao_acionamento acima_maximo 5.6 15;
  map tensao_bomba normal 4.5 5.5;
  map tensao_bomba abaixo_minimo 0 4.4;
  map tensao_bomba acima_maximo 5.6 15;
end component

component motor: p_motor
  name "Motor Ford";
end component

component bateria: p_bateria
  name "Bateria 12V";
end component

component sensor_temp_agua: p_sensor_temperatura
  name "Sensor de temperatura da água";
  symptoms P0115, P0117;
end component

component sensor_massa_ar: p_sensor_massa_ar
  name "Sensor de massa de ar";
end component

component bomba_combustivel: p_bomba_combustivel
  name "Bomba de combustível";
  symptoms P0230, P0231;
end component

component injetor: p_injetor
  name "Injetor de combustível";
end component

component bobina_ignicao: p_bobina_ignicao
  name "Bobina de ignição";
```

```
    symptoms P0350, P1231;
end component

component tanque_combustivel: p_tanque_combustivel
    name "Tanque de combustível";
end component

component uce: p_uce
    name "Unidade de controle eletrônica";
    symptoms P0101, P0115, P0117, P0230, P0231, P0350, P1231, P1359;
end component

connections
    bateria(tensao) -> uce(tensao_bateria);
    uce(tensao_sensor_temp_agua) -> sensor_temp_agua(tensao_entrada);
    uce(tensao_sensor_massa_ar) -> sensor_massa_ar(tensao_entrada);
    uce(tensao_bobina_ignicao) -> bobina_ignicao(tensao_entrada)
    uce(tensao_rele_bomba) -> rele_bomba(tensao_acionamento);
    uce(tensao_injetor) -> injetor(tensao_entrada);
    uce(sensor_temp_agua) -> sensor_temp_agua(tensao_saida);
    uce(sensor_massa_ar) -> sensor_massa_ar(tensao_saida);
    uce(avanco_ignicao) -> bobina_ignicao(avanco_ignicao);
    uce(tempo_injecao) -> injetor(tempo_injecao);
    rele_bomba(tensao_bomba) -> bomba_combustivel(tensao_entrada);
    tanque_combustivel(estado_combustivel) ->
        bomba_combustivel(estado_combustivel);
    bomba_combustivel(fluxo_combustivel) -> injetor(fluxo_combustivel);
    injetor(injecao_combustivel) -> motor(injecao);
    bobina_ignicao(ignicao) -> motor(ignicao);
end connections

end system
```

Apêndice C

Definição dos modelos do Autodiag em XML

Este apêndice apresenta a definição do tipo de documento XML, usado para codificar os modelos dos sistemas a serem diagnosticados pelo Autodiag em documentos XML.

C.1 DTD para os modelos do Autodiag

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- O elemento model contém a declaração de um modelo a ser usado pelo
autodiag. O modelo é composto por zero ou mais definições de domínios, por uma
ou mais declarações de protótipos de componentes e por uma única definição de
sistema. -->
<!ELEMENT model (domain*, prototype+, system)>

<!-- O elemento domínio contém a definição de um domínio de valores
qualitativos. Um domínio é composto por um nome e por um ou mais valores
qualitativos. -->
<!ELEMENT domain (domain-name, domain-value+)>

<!-- O nome do domínio é definido por um string textual. -->
<!ELEMENT domain-name (#PCDATA)>

<!-- Um valor do domínio é definido por um string textual. -->
<!ELEMENT domain-value (#PCDATA)>

<!-- O elemento component contém a definição de um protótipo de componente
```

utilizado pelo sistema. Um componente é composto por um ou mais portos e por uma ou mais regras relacionando os portos e seus valores. -->

```
<!ELEMENT prototype (prototype-name, port+, rule+)>
```

```
<!-- Define o nome do tipo de componente -->
```

```
<!ELEMENT prototype-name (#PCDATA)>
```

```
<!-- Define um porto do componente, ou seja, um local do componente que pode receber um valor. -->
```

```
<!ELEMENT port (port-name, port-domain)>
```

```
<!-- Define o nome do porto do componente. -->
```

```
<!ELEMENT port-name (#PCDATA)>
```

```
<!-- Define o domínio dos valores que podem ser atribuídos ao porto do componente. -->
```

```
<!ELEMENT port-domain (#PCDATA)>
```

```
<!-- Define um regra de comportamento do componente envolvendo relações de igualdade entre os portos do componente e seus respectivos valores. -->
```

```
<!ELEMENT rule (condition+, action+)>
```

```
<!-- Define as condições para que a regra seja executada. Por enquanto a única avaliação possível é o teste de igualdade, ou seja, testar se um porto possui um determinado valor. -->
```

```
<!ELEMENT condition (condition-port, condition-value)>
```

```
<!-- Define o porto cujo valor deve ser avaliado. -->
```

```
<!ELEMENT condition-port (#PCDATA)>
```

```
<!-- Define o valor cuja igualdade deve ser testada com o valor do porto definido por 'rule-condition-port'. -->
```

```
<!ELEMENT condition-value (#PCDATA)>
```

```
<!-- Define as ações a serem executadas pela regra quando as condições forem satisfeitas. Por enquanto as únicas ações possíveis são: a atribuição de um valor a um porto do componente e a conclusão de que o componente está defeituoso -->
```

```
<!ELEMENT action (set-action+|faulty-action)>
```

```
<!ELEMENT set-action (action-port, action-value)>
```

```
<!ELEMENT faulty-action EMPTY>
```

```
<!-- Define o porto cujo valor deve ser atribuído. -->
```

```
<!ELEMENT action-port (#PCDATA)>
```

```
<!-- Define o valor a ser atribuído ao porto quando a regra for executada. -->
<!ELEMENT action-value (#PCDATA)>

<!-- Define o sistema a ser diagnosticado pelo autodiag, composto por
componentes interconectados. -->
<!ELEMENT system (system-name, symptom*, component+, connection*)>

<!-- Define nome do sistema. -->
<!ELEMENT system-name (#PCDATA)>

<!-- Define os sintomas reconhecidos diretamente pelo sistema a ser
diagnosticado. -->
<!ELEMENT symptom (symptom-name, symptom-description)>

<!-- Define nome do sintoma. -->
<!ELEMENT symptom-name (#PCDATA)>

<!-- Define a descrição do sintoma. -->
<!ELEMENT symptom-description (#PCDATA)>

<!-- Define um componente do sistema. O componente é responsável por declarar
todos os nomes dos sintomas que ele deve ser considerado como componente
suspeito. Além disso, o componente deve possuir um protótipo previamente
definido. O componente pode ainda disponibilizar um mapeamento de valores
quantitativos dos seus portos para os valores qualitativos correspondentes do
domínio do porto. -->
<!ELEMENT component
  (component-name, component-prototype, component-symptom*, component-map*)>

<!-- Define o nome do componente do sistema. Cada componente deve possuir um
valor único para este atributo. -->
<!ELEMENT component-name (#PCDATA)>

<!-- Define o nome do protótipo do componente do sistema, conforme definido no
atributo prototype-name do elemento <component>. -->
<!ELEMENT component-prototype (#PCDATA)>

<!-- Define os sintomas para os quais o componente deve ser considerado
suspeito. Os nomes devem corresponder aos nomes previamente declarados para o
elemento <symptom>. -->
<!ELEMENT component-symptom (#PCDATA)>

<!-- Define o mapeamento de valores quantitativos para os valores qualitativos
do domínio do porto. -->
<!ELEMENT component-map (map-port, map-quantitative-to-qualitative+)>
```

```
<!-- Define o nome do port, conforme definido por port-name, cujos valores
serão mapeados. -->
<!ELEMENT map-port (#PCDATA)>

<!-- Define o mapeamento entre valores qualitativos e quantitativos de um
porto. -->
<!ELEMENT map-quantitative-to-qualitative
  (min-value, max-value, qualitative-value)>

<!-- Define o valor quantitativo mínimo para o mapeamento. -->
<!ELEMENT min-value (#PCDATA)>

<!-- Define o valor quantitativo máximo para o mapeamento. -->
<!ELEMENT max-value (#PCDATA)>

<!-- Define o valor qualitativo correspondente do mapeamento. -->
<!ELEMENT qualitative-value (#PCDATA)>

<!-- Define um conexão entre dois portos de componentes do sistema. -->
<!ELEMENT connection (connection-source, connection-target)>

<!-- Define participante cuja saída será conectada à entrada do outro
participante. -->
<!ELEMENT connection-source (component-name, port-name)>

<!-- Define participante cuja entrada será conectada à saída do outro
participante. -->
<!ELEMENT connection-target (component-name, port-name)>
```

Referências Bibliográficas

- [1] Joseph Barkai. Vehicle diagnostics - are you ready for the challenge? *SAE Paper 01ATT-208*, 2001.
- [2] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.
- [3] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.
- [4] Igor Mozetic. Model-based diagnosis: An overview. In *Advanced Topics in Artificial Intelligence*, pages 419–430, 1992.
- [5] K. Balakrishnan and V. Honavar. Intelligent Diagnosis Systems. Technical report, Department of Computer Science, Iowa State University, Ames, Iowa, 1995.
- [6] Randall davis. Expert systems: Where are we? and where do we go from here? *The AI Magazine*, Spring 1982, 1982.
- [7] B.G. Buchanan and Shorttiffe E.H. (Eds.). *Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA, 1984.
- [8] J. Kolodner. *Case-Based Reasoning*. Morgan Kauffman, San Mateo, CA, 1993.
- [9] Mattias Nyberg and Lars Nielsen. Model based diagnosis for the air intake system of the si-engine. *SAE Paper 970209*, 1997.
- [10] Mattias Nyberg, Thomas Stutte, and Volker Wilhelmi. Model based diagnosis of the air path of an automotive diesel engine. *IFAC Automotive Workshop. Karlsruhe, Germany: IFAC*, 2001.
- [11] Air Resources Board. *Malfunction and Diagnostic System Requirements - 1994 and Subsequent Model-Year Passenger Cars, Light-Duty Trucks, and Medium-Duty Vehicles and Engines (OBD II)*. Arb, 1996.
- [12] Society of Automotive Engineers. *SAE J1978 - OBD II Scan Tool*. SAE, 1997.

-
- [13] Society of Automotive Engineers. *SAE J1979 - E/E Diagnostic Test Modes*. SAE, 1998.
- [14] Society of Automotive Engineers. *SAE J2012 - Recommended Format and Messages for Diagnostic Trouble Codes*. SAE.
- [15] Society of Automotive Engineers. *SAE J1850 - Class B Data Communication Network Interface*. SAE.
- [16] Society of Automotive Engineers. *SAE J1962 - Diagnostic Conector*. SAE.
- [17] International Organization for Standardization. *ISO 14229 - Road vehicles - Diagnostic systems - Diagnostic Services Specification*. ISO, 1996.
- [18] International Organization for Standardization. *ISO/DIS 14230-1 - Road Vehicles - Diagnostic systems - Keyword Protocol 2000 - Part 1: Physical Layer*. ISO, 1995.
- [19] International Organization for Standardization. *ISO 14230-2 Keyword protocol 2000 - Part 2: Data link layer*. ISO, 1996.
- [20] International Organization for Standardization. *ISO 14230-3 - Road Vehicles - Diagnostic systems - Keyword Protocol 2000 - Draft Part 3: Implementation*. ISO, 1996.
- [21] International Organization for Standardization. *ISO 14230-4 - Road Vehicles - Diagnostic systems - Keyword Protocol 2000 - Part 4: Requirements For Emission related Systems*. ISO, 1997.
- [22] International Organization for Standardization. *ISO 9141 - Diagnostic Systems - Requirements for Interchange of Digital Information*. ISO, 1989.
- [23] International Organization for Standardization. *ISO 9141-2 - Road vehicles - Diagnostic systems - CARB requirements for interchange of digital information*. ISO, 1994.
- [24] International Organization for Standardization. *ISO 7498 - Open Systems Interconnection*. ISO.
- [25] Valdeci Pereira Belo and Antônio Otávio Fernandes. Uma ferramenta de inspeção para veículos automotores padrão obd-2. Relatório técnico, Universidade Federal de Minas Gerais, 2000.
- [26] Allan W. M. Bonnick. *Vehicle Eletronic Systems and Fault Diagnosis - A Pratical Guide for Vehicle Technicians*. STS Press, 1998.
- [27] Oskar Dressler and Peter Struss. The consistency-based approach to automated diagnosis of devices. In G. Brewka, editor, *Principles of Knowledge Representation*. CSLI, 1996.

-
- [28] Dirk van Dalen. *Logic and Structure*. Springer, 3 edition, 1997.
- [29] Suzanna S. Epp. *Discrete Mathematics with Applications*. PWS Publishing Company, 2 edition, 1995.
- [30] K. D. Forbus and J. De Kleer. *Building Problem Solvers*. MIT Press, Cambridge, MA, 1993.
- [31] J. de Kleer. An assumption-based tms. *Artificial Intelligence*, 28:127–162, 1986.
- [32] J. de Kleer. Extending the atms. *Artificial Intelligence*, 28:163–224, 1986.
- [33] Oskar Dressler and Adam Farquhar. Putting the problem solver back in the driver’s seat: Contextual control of the AMTS. In *Truth Maintenance Systems (ECAI Workshop)*, pages 1–16, 1990.
- [34] P. A. Bidian, M. Tatar, F. Cascio, Theseider Dupré, M. Sachenbacher, and R. Weber. Powertrain diagnostics a model-based approach. *Conference on Vehicle Electronic Systems*, 1999.
- [35] Fulvio Cascio, Luca Console, Marcella Guagliumi Massimo Osella, Andrea Panati, Sara Sottano, and Daniele Theseider Dupre. Generating on-board diagnostics of dynamic automotive systems based on qualitative models. *AI Communications*, 12(1-2):33–43, 1999.
- [36] A. Malik and P. Struss. Diagnosis of dynamic systems does not necessarily require simulation. In AAAI Press, editor, *Workshop Notes of the 10th International Workshop on Qualitative Reasoning (QR-96)*, 1996.
- [37] Peter Struss, Andreas Malik, and Martin Sachenbacher. Qualitative modeling is the key to automated diagnosis. In *Proceedings of 13th IFAC World Congress*, San Francisco, 1996.
- [38] Thomas Havelka, Markus Stumptner, and Franz Wotawa. Ad2l - a programming language for model-based systems (preliminary report). In *Proceedings of the 11th International Workshop on Principles of Diagnosis (DX-00)*, Morelia, Mexico, June 8-10 2000.
- [39] J. de Kleer. Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45:381–391, 1990.
- [40] Wilson de Pádua Paula Filho. *Engenharia de Software, Fundamentos, Métodos e Padrões*. LTC, Rio de Janeiro, RJ, 2001.
- [41] Sun Microsystems. *The Java Tutorial - A practical guide for programmers*. Sun Microsystems, Inc, 2003.

-
- [42] Eric Armstrong et al. *The Java Web Services Tutorial*. Sun Microsystems, 2003.
- [43] Fábio Ribeiro Von Glehn. *Curso de Injeção Eletrônica*. Ciclo Engenharia.