

Especificação de Requisitos de Software com o Método SCR

Flávio Ricardo Carpena
Tereza Gonçalves Kirner

Departamento de Computação - Universidade Federal de São Carlos
Rodovia Washington Luiz, Km 235 - Caixa Postal 676
CEP 13565-905, São Carlos, SP
E-mail: {carpena, tereza}@dc.ufscar.br

Resumo

Este artigo apresenta um método formal para a especificação de requisitos de software, o método SCR - Software Cost Reduction, detalhando seu processo de utilização através de um estudo de caso. É feita uma discussão dos pontos fortes e dificuldades inerentes ao uso do método.

Palavras-chave: Especificação de Requisitos, Métodos Formais, Método SCR.

Abstract

This article presents and discusses a formal method for software requirements specification, the SCR - Software Cost Reduction method, detailing its utilization process by means of a case study. The strengthness and difficulties of the method, and other concluding remarks, are pointed out.

Keywords: Requirements Specification, Formal Methods, SCR Method.

1. Introdução

A especificação de requisitos é uma etapa essencial do processo de desenvolvimento de software, que compreende uma definição completa do comportamento externo do sistema de software, tanto em termos de requisitos funcionais quanto de requisitos não funcionais. Vários estudos identificaram que a definição inadequada de requisitos é responsável por uma parte significativa dos erros detectados ao longo do processo de desenvolvimento de sistemas, principalmente no caso de sistemas dedicados, de tempo real e críticos (Lutz [8]). Outros estudos indicam que a eliminação de erros de especificação torna-se cada vez mais difícil e dispendiosa à medida que o sistema avança para etapas posteriores do seu ciclo de vida, como projeto e implementação (Davis [3]).

O emprego de métodos formais é um meio efetivo de garantir a qualidade dos requisitos, uma vez que esses métodos produzem especificações precisas, que podem ser rigorosamente validadas. Várias experiências bem sucedidas sobre o uso de métodos formais de especificação, em ambientes industriais, são relatadas em Craigen [2], e incluem aplicações de controle de satélites, controle de tráfego, sistemas de automação de manufatura, sistemas de defesa, entre outros. Porém, a adoção de métodos formais ainda enfrenta barreiras, principalmente no ambiente das empresas. Uma delas é a idéia de que esses métodos são difíceis de serem utilizados e de serem entendidos, uma vez que muitos dos métodos formais baseiam-se em linguagens de especificação que empregam formalismos com forte base matemática e lógica.

O método SCR (*Software Cost Reduction*), enfocado nesse artigo, utiliza um formalismo de alto nível, sem empregar uma linguagem matemática complexa. Em sua versão atual, o SCR compreende a definição de um modelo de alto nível do sistema (chamado *Modelo de Quatro Variáveis*), e um modelo de representação tabular, apoiado por um conjunto de construções e definições (chamado *Modelo Formal de Requisitos*). Tais características tornam o método SCR mais fácil de ser entendido e utilizado por desenvolvedores que possuem uma formação e experiência normalmente encontrada entre os profissionais que especificam sistemas. O método SCR já vem sendo empregado, a nível internacional, em vários projetos reais, conforme é relatado em Bharadwaj [1] e Heitmeyer [6].

Este artigo tem por finalidade apresentar o método SCR, detalhando o seu processo de utilização através de um estudo de caso. A seção 2 descreve o método SCR; a seção 3 complementa a descrição do método, ilustrando sua utilização através da especificação de um exemplo de sistema; e a seção 4 contém as conclusões do artigo.

2. Apresentação do Método SCR

O método SCR foi inicialmente proposto por David Parnas, através do documento de requisitos do sistema *A-7 Operational Flight Program* (Heninger [7]; Parnas [9]). Esse documento contém a maioria das características fundamentais do método, como, por exemplo, a notação tabular empregada na criação do Modelo Formal de Requisitos. Mais recentemente, outros pesquisadores, como Faulk [4], Heitmeyer [5], e van Schouwen [10], estenderam e refinaram o método, tornando-o adequado para especificar tanto requisitos de software quanto requisitos do sistema como um todo. Essa versão mais atual é a apresentada no presente artigo.

Um dos componentes básicos do método SCR é o *Modelo de Quatro Variáveis*, que permite representar a interação do sistema com o seu ambiente. Após a identificação das variáveis, é gerado um *Modelo Formal de Requisitos*, utilizando as construções e a notação tabular. As construções são utilizadas para representar os requisitos do sistema, que são capturados como relações entre variáveis monitoradas e controladas. A notação tabular é utilizada para representar as funções que especificam o comportamento do sistema.

2.1 Modelo de Quatro Variáveis

É um modelo abstrato que trata o comportamento do sistema como um conjunto de relações matemáticas envolvendo quatro variáveis: *controladas*, *monitoradas*, *itens de dados de entrada* e *itens de dados de saída*. O Modelo de Quatro Variáveis tem como propósito ser independente de implementação, especificando apenas o comportamento externo requerido. Os componentes desse modelo são descritos em seguida.

a) Variáveis

A identificação das variáveis deve ser feita observando-se, além das necessidades dos usuários, as propriedades físicas do sistema, incluindo valores capturados ou fornecidos através de sensores ou *displays*. As variáveis presentes no Modelo de Quatro Variáveis são descritas a seguir.

- *Monitoradas*. São valores do ambiente que influenciam o comportamento do sistema.
- *Controladas*. São valores do ambiente que o sistema controla.
- *Itens de dados*. São valores intermediários entre os dispositivos de entrada e de saída e o software. *Itens de dados de entrada* são valores resultantes da leitura de variáveis monitoradas pelos dispositivos de entrada e *itens de dados de saída* são valores que serão fornecidos aos dispositivos de saída para a determinação das variáveis controladas.

b) Relações

As seguintes relações podem ocorrer no contexto do funcionamento do sistema: *REQ*, *NAT*, *IN*, *OUT* e *SOFT*. O comportamento do sistema é dado pelas relações *NAT* e *REQ*. A relação *NAT* define o conjunto de valores possíveis e captura as necessidades impostas por leis físicas. A relação *REQ* define o comportamento ideal do sistema, representando a relação entre as variáveis monitoradas e controladas. A relação *IN* define a precisão com a qual os dispositivos de entrada capturam os valores monitorados ou define o relacionamento entre variáveis monitoradas e itens de dados de entrada. A relação *OUT* define a precisão com a qual os dispositivos de saída atualizam as variáveis controladas ou define o relacionamento entre itens de dados de saída e variáveis controladas. A relação *SOFT* define a correspondência entre itens de dados e o software.

2.2 Modelo Formal de Requisitos

Enquanto o Modelo de Quatro Variáveis descreve alguns aspectos do SCR, o Modelo Formal de Requisitos é menos abstrato e constitui a base formal para a utilização do método. Nesse modelo, é considerado apenas o comportamento ideal do sistema (omitindo detalhes como precisão e tempo) e os requisitos são expressos como conjuntos de funções sobre estados de variáveis. A relação *REQ* do Modelo de Quatro Variáveis é descrita pelo Modelo Formal de Requisitos, que utiliza para isso as construções e a notação tabular, conforme apresentado nos itens abaixo.

a) Construções

As construções são utilizadas para representar o relacionamento entre as variáveis identificadas no contexto do sistema. As principais construções são descritas a seguir.

- *Classe de Modos*. É uma máquina de estado cujas transições são disparadas por eventos. Alguns sistemas podem possuir mais de uma classe de modos. O modo corrente de cada classe afeta o comportamento das funções associadas.
- *Termo*. É uma função de variáveis monitoradas, modos ou outros termos.
- *Condição*. É uma declaração lógica. Uma condição composta pode ser construída através da justaposição de condições utilizando operadores lógicos.
- *Evento*. É a mudança de valor de um termo, classe de modo ou variável controlada.

Para maior clareza do documento de requisitos, pode ser utilizado um identificador no início do nome indicando se o mesmo refere-se a variáveis monitoradas, controladas, termos ou classes de modos. Por exemplo: *m*; *c*; *t*; *mc*.

b) Notação Tabular

A notação tabular tem o objetivo de estruturar e simplificar os requisitos, possibilitando a análise e documentação detalhada de partes da especificação. O método SCR utiliza as seguintes tabelas:

- *Condições*. Define uma variável controlada ou termo como uma função de modos e condições.
- *Eventos*. Define uma variável controlada ou termo como uma função de modos e eventos.
- *Transição de modos*. Define a classe de modos, especificando os eventos que devem ocorrer para que a transição entre dois modos consecutivos seja disparada.

O Modelo Formal de Requisitos também define uma *relação de dependência* entre as variáveis monitoradas, controladas, classes de modos e termos. O conjunto dessas relações é utilizado no estabelecimento da *seqüência de processamento* que determina o funcionamento do sistema. As variáveis monitoradas iniciam essa seqüência pois dependem apenas do ambiente,

ao passo que as variáveis controladas aparecem depois, uma vez que podem depender de variáveis monitoradas, classes de modos ou termos. Um elemento da relação de dependência aparece definido após outro na seqüência de processamento caso dependa do mesmo.

3. Utilização do Método SCR

O propósito desta seção é detalhar o processo de especificação de requisitos com o SCR, através de um exemplo de sistema. A seqüência de atividades apresentada visa facilitar a compreensão do método que, na prática, pode não seguir uma ordem rígida.

3.1 Exemplo de Sistema

O uso do método será ilustrado através da especificação de um Sistema de Mistura de Líquidos, cujo objetivo é obter uma mistura, a partir dos líquidos de dois reservatórios (A e B). A mistura obtida é aquecida e depois engarrafada, conforme ilustrado na Figura 1.

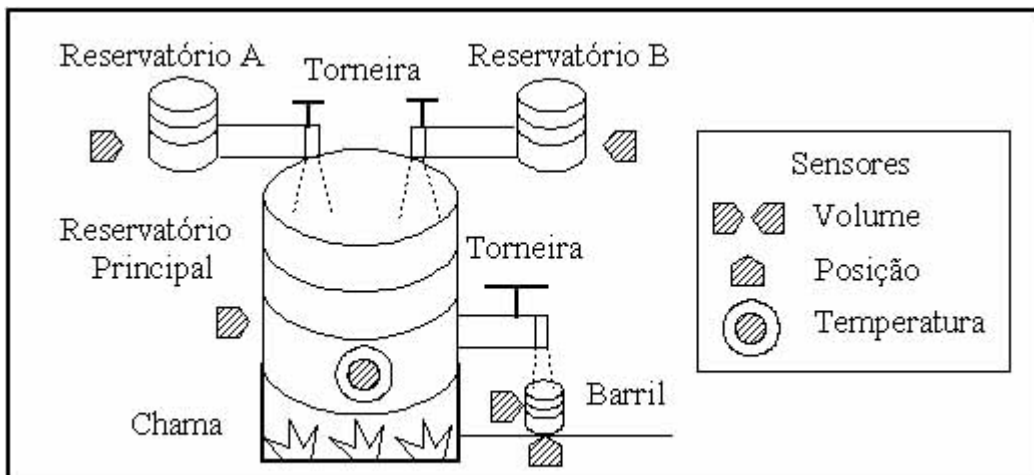


Figura 1. Sistema de Mistura de Líquidos

É importante destacar a existência de mecanismos manuais para o esvaziamento e reposição de líquido dos reservatórios. O Sistema de Mistura compreende 6 estágios de funcionamento, descritos na Tabela 1.

Tabela 1. Descrição Informal dos Estágios do Sistema de Mist

<p>Desativado. <i>Desativado</i> é o estágio inicial e o sistema está com todas as suas atividades ocorrendo. Pode ocorrer uma transição para o estágio <i>Mistura</i> se a chave de ativação for ligada, os volumes A e B forem maiores que o mínimo, o <i>status</i> do barril for <i>true</i>, e o volume do reservatório principal for menor que o máximo.</p>
<p>Mistura. No estágio <i>Mistura</i>, as torneiras dos reservatórios A e B estão abertas e a mistura é preparada no reservatório principal. Pode ocorrer uma transição para um dos seguintes estágios: <i>Desativado</i>, se a temperatura for menor ou igual a mínima e o volume do reservatório principal for menor que o máximo; <i>Vazão</i>, se a temperatura e o volume do reservatório principal forem maiores que o mínimo e o <i>status</i> do barril for <i>true</i>; <i>Crítico da Vazão</i>, se a temperatura e o volume do reservatório principal forem maiores que o mínimo e o <i>status</i> do barril for <i>false</i>; <i>Crítico da Mistura</i>, se os volumes dos reservatórios A e B forem menores ou iguais ao mínimo; <i>Desativado</i>, se a chave de ativação for desligada.</p>
<p>Crítico da Mistura. No estágio <i>Crítico da Mistura</i>, as torneiras estão fechadas e a chama está acesa. Pode ocorrer uma transição para um dos seguintes estágios: <i>Mistura</i>, se os volumes dos reservatórios A e B forem maiores que o mínimo; <i>Desativado</i>, se o tempo de permanência for excedido ou a chave de ativação for desligada.</p>
<p>Aquecimento. No estágio <i>Aquecimento</i>, as torneiras estão fechadas e a chama está ativa. Pode ocorrer uma transição para um dos seguintes estágios: <i>Vazão</i>, se a temperatura for maior que a mínima e o <i>status</i> do barril for <i>true</i>; <i>Crítico da Vazão</i>, se a temperatura for maior que a mínima e o <i>status</i> do barril for <i>false</i>; <i>Desativado</i>, se a chave de ativação for desligada.</p>
<p>Vazão. No estágio <i>Vazão</i>, a torneira do reservatório principal está aberta. Pode ocorrer uma transição para um dos seguintes estágios: <i>Mistura</i>, se o volume do reservatório principal for menor que o máximo e os volumes dos reservatórios A e B forem maiores que o mínimo; <i>Crítico da Mistura</i>, se o volume do reservatório principal e do reservatório A ou B forem menores ou iguais ao mínimo; <i>Crítico da Vazão</i>, se o <i>status</i> do barril for <i>false</i>; <i>Aquecimento</i>, se a temperatura for menor ou igual a mínima; <i>Desativado</i>, se a chave de ativação for desligada.</p>
<p>Crítico da Vazão. No estágio <i>Crítico da Vazão</i>, as torneiras estão fechadas e a chama está acesa. Pode ocorrer uma transição para um dos seguintes estágios: <i>Vazão</i>, se o <i>status</i> do barril for <i>true</i> e o tempo de permanência for excedido, o volume do reservatório principal for menor que o máximo e os volumes dos reservatórios A e B forem maiores que o mínimo; <i>Crítico da Mistura</i>, se o tempo de permanência for excedido, o volume do reservatório principal for menor que o máximo, e o volume do reservatório A ou B forem menores ou iguais ao mínimo; <i>Aquecimento</i>, se a temperatura for menor ou igual a mínima; <i>Desativado</i>, se o tempo de permanência for excedido, o volume do reservatório principal for maior ou igual ao máximo ou a chave de ativação for desligada.</p>

Os estágios *Crítico da Mistura* e *Crítico da Vazão* representam a ocorrência de uma situação que, se não for corrigida dentro de limites de tempo previamente estipulados, comprometerá o funcionamento do sistema. Um alarme será disparado quando o sistema entrar em um desses estágios.

As seguintes restrições devem ser obedecidas ao longo do funcionamento do sistema: (a) o tempo máximo de permanência em um estágio crítico é 100 segundos; (b) o sistema pode ser desativado a qualquer momento, através de uma chave; (c) a torneira do reservatório principal não deve estar aberta ao mesmo tempo que as torneiras A e B; (d) a densidade ideal da mistura é obtida a partir de volumes iguais dos líquidos A e B; (e) a troca de barril e a reposição de líquido nos reservatórios A e B são feitas por subsistemas cujo acionamento estará vinculado a valores de variáveis controladas do Sistema de Mistura.

3. 2 Especificação do Sistema de Mistura de Líquidos

a) Criação do Modelo de Quatro Variáveis

O Modelo de Quatro Variáveis contém uma representação em alto nível do sistema, mostrando seu funcionamento geral e incluindo as entradas e saídas. Todos os dispositivos físicos e valores produzidos a partir desses são definidos pelo modelo.

A Figura 2 mostra o Modelo de Quatro Variáveis para o Sistema de Mistura de Líquidos.

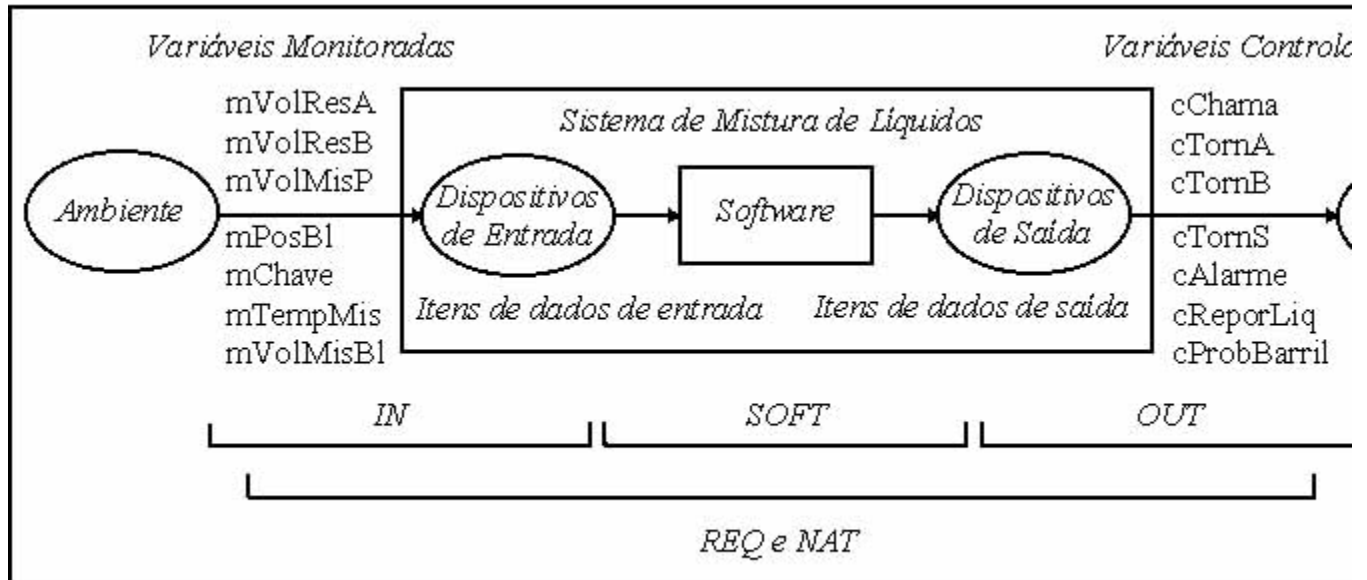


Figura 2. Modelo de Quatro Variáveis para o Sistema de Mistura de Líq

b) Identificação da Classe de Modos

Os estágios apresentados na Tabela 1 serão representados por modos da classe *mcMistura*. As classes de modos são importantes, pois organizam a especificação em função de um conjunto de modos. A Figura 3 apresenta um grafo de *mcMistura* onde *D* é o modo inicial. Uma seta indica a possibilidade de ocorrer uma transição e aponta para o modo destino. Esse grafo irá auxiliar a construção da tabela de transição.

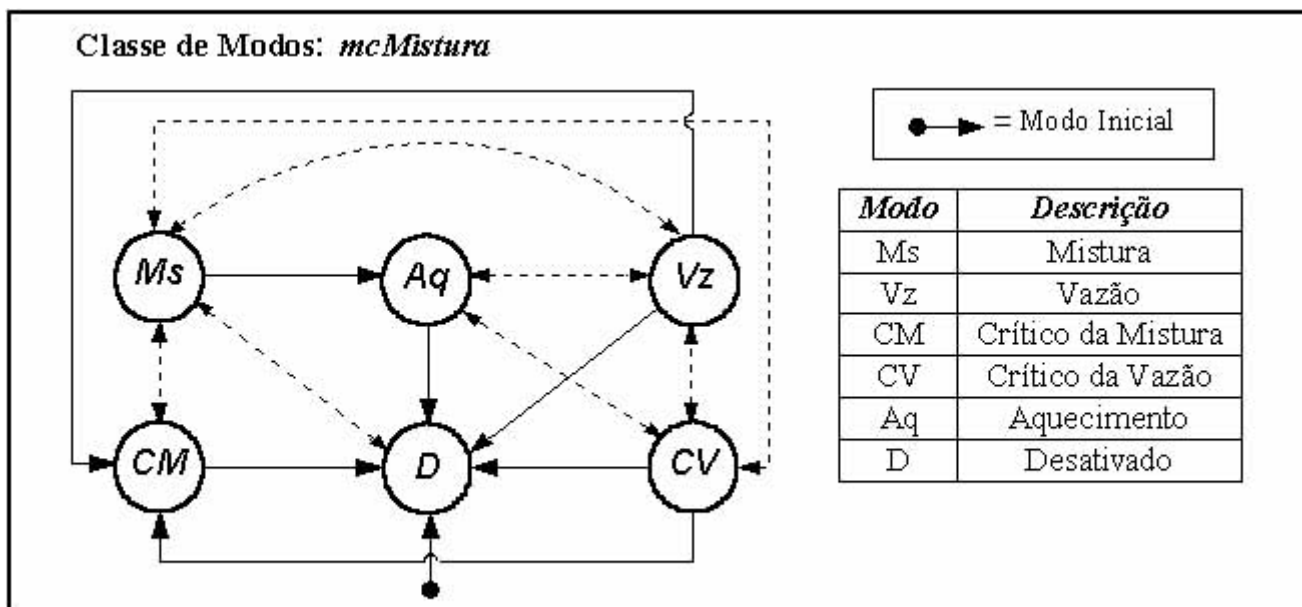


Figura 3. Classe de Modos *mcMistura*

c) Definição de Tipos e Entidades

O método SCR suporta dois tipos base independentes da especificação: inteiro e enumerado. Esses são empregados na definição de tipos dependentes da especificação que está sendo preparada. Por exemplo, na Tabela 2, *Temp* é um tipo relacionado apenas com o Sistema de Mistura e suporta valores do tipo base *inteiro* variando de -10 a 40. O tipo *Temp* está associado com a unidade *Celsius*.

Tabela 2. Definição de Tipos

<i>Tipo</i>	<i>Tipo Base</i>	<i>Faixa de Valor</i>	<i>Unidade</i>
Boolean	Enumerado	True, False	N/A
VolRes	Inteiro	0,20000	Litros
VolBl	Inteiro	0,50	Litros
Temp	Inteiro	-10,40	Celsius
Switch	Enumerado	On,Off	N/A
Tempo	Inteiro	1,100	Segundos

A Tabela 3 apresenta a definição de entidades do sistema. A palavra entidade, nesse contexto, faz referência a termos, constantes, variáveis monitoradas e controladas. Por exemplo: *cTorna* é uma variável controlada, definida por uma tabela de condição, do tipo *Switch*, possui valor inicial *Off*, e representa a torneira do reservatório A.

Tabela 3. Definição de Entidades

<i>Nome</i>	<i>Classe</i>	<i>Tabela</i>	<i>Tipo</i>	<i>Valor Inicial</i>	<i>Descrição</i>
mVolResA	Monit	N/A	VolRes	10000	Volume do Reservatório A
mVolResB	Monit	N/A	VolRes	10000	Volume do Reservatório B
mVolMisP	Monit	N/A	VolRes	0	Volume do Reservatório Principal
mVolMisBl	Monit	N/A	VolBl	0	Volume do Barril
mTempMis	Monit	N/A	Temp	0	Temperatura da Mistura
mPosBl	Monit	N/A	Boolean	True	Posição do Barril
mChave	Monit	N/A	Boolean	False	Chave de Ativação
tVolResAB	Termo	Cnd	Boolean	True	Volume dos Reservatórios A e B
tBarril	Termo	Cnd	Boolean	True	Posição e Volume do Barril
tTempPer	Termo	Cnd	Boolean	False	Temperatura Permitida
cChama	Contr	Cnd	Switch	Off	Chama
cTornA	Contr	Cnd	Switch	Off	Torneira do Reservatório A
cTornB	Contr	Cnd	Switch	Off	Torneira do Reservatório B
cTornS	Contr	Cnd	Switch	Off	Torneira do Reservatório Principal
cAlarme	Contr	Cnd	Switch	Off	Entrada em Modo Crítico
cReporLiq	Contr	Cnd	Boolean	False	Repor Líquido
cProbBarril	Contr	Cnd	Boolean	False	Problema no Barril
TempMnMis	Const	N/A	Temp	30	Temperatura Mínima
TpMxPer	Const	N/A	Tempo	100	Tempo Máximo de Permanência
VolMxBl	Const	N/A	VolBl	50	Volume Máximo do Barril
VolMxResP	Const	N/A	VolRes	20000	Volume Máximo do Reservatório Princi
VolMnResP	Const	N/A	VolRes	0	Volume Mínimo do Reservatório Princi
VolMxResAB	Const	N/A	VolRes	10000	Volume Máximo dos Reservatórios A e
VolMnResAB	Const	N/A	VolRes	0	Volume Mínimo dos Reservatórios A e

d) Construção do Diagrama de Dependências

A Figura 4 apresenta as dependências entre variáveis monitoradas e controladas, termos e classes de modos. Uma seta indica o sentido da dependência, por exemplo, a Figura 4 define que *tVolResAB* depende de *mVolResA* e *mVolResB*.

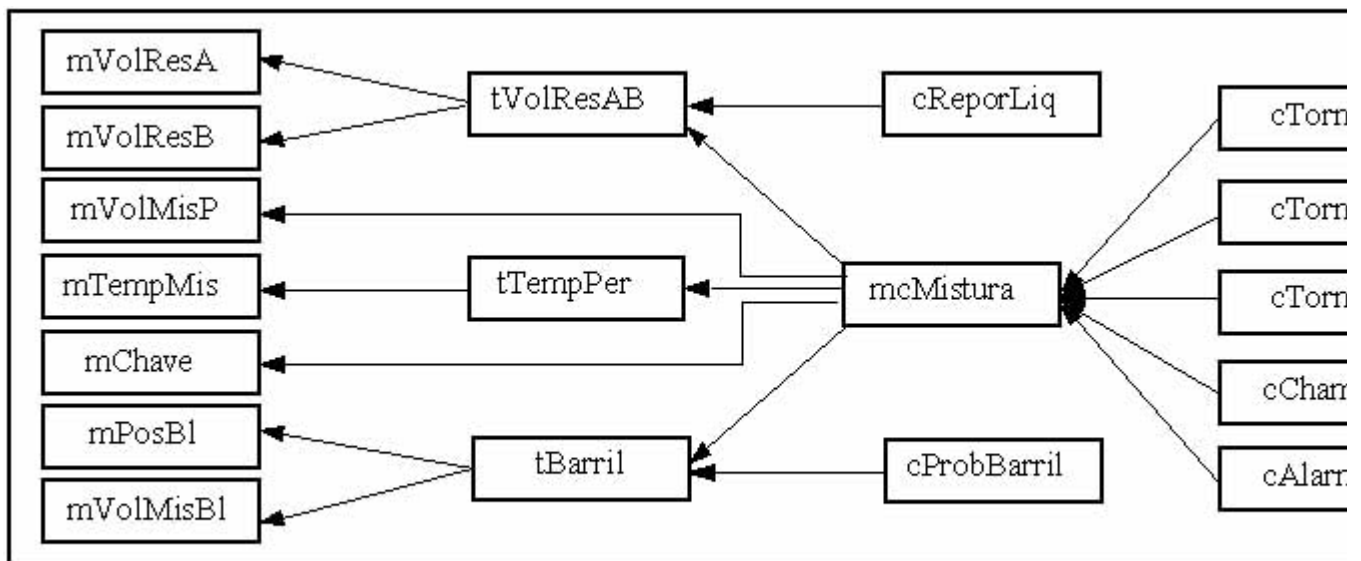


Figura 4. Diagrama de Dependências

e) Construção das Tabelas

Tabelas de Condições

As tabelas da Figura 5 definem as variáveis controladas e termos do Sistema de Mistura. A tabela do item (d) determina que o novo valor de *cReporLiq* - *cReporLiq'* - será *True* se *tVolResAB* for *False* (o volume do reservatório A ou B é menor ou igual ao mínimo) e *False* caso contrário.

A entrada *True* no lugar de uma condição determina que a condição para atribuição do valor é a classe de modos assumir como modo corrente um dos modos da respectiva linha. A entrada *False* determina que não pode ser atribuído o valor da coluna se a classe de modos assumir um dos modos da respectiva linha. Por exemplo: a tabela do item (f) da Figura 5 especifica que sempre que o modo corrente for *Ms*, *cTornA'* receberá o valor *On* e em nenhuma situação *Off*.

		<i>Condições</i>	
(a)		$((mVolResA > VolMnResAB) \ \&\& \ (mVolResB > VolMnResAB))$	$(!(mVolResA > VolMnResAB) \ \& \ (mVolResB > VolMnResAB))$
	<i>tVolResAB'</i>	True	False
<hr/>			
		<i>Condições</i>	
(b)		$((mPosBl) \ \&\& \ (mVolMisBl < VolMxBl))$	$(!(mPosBl) \ \&\& \ (mVolMisBl < VolMxBl))$
	<i>tBarril'</i>	True	False
<hr/>			
		<i>Condições</i>	
(c)		$(mTempMis > TempMnMis)$	$(!(mTempMis > TempMnMis))$
	<i>tTempPer'</i>	True	False
<hr/>			
(d)		<i>Condições</i>	
		$(!(tVolResAB))$	$(tVolResAB)$
	<i>cReporLiq'</i>	True	False
<hr/>			
(f)	<i>Modos</i>	<i>Condições</i>	
	Ms	True	False
	Aq, Vz, CV, CM, D	False	True
	<i>cTornA'</i>	On	Off
<hr/>			
(h)	<i>Modos</i>	<i>Condições</i>	
	Vz	True	False
	Ms, Aq, CM, CV, D	False	True
	<i>cTornS'</i>	On	Off
<hr/>			
(j)	<i>Modos</i>	<i>Condições</i>	
	Aq	True	False
	Ms, Vz, CM, CV, D	False	True
	<i>cChama'</i>	On	Off

		<i>Condições</i>	
(e)		$(!(tBarril))$	$(tBarril)$
	<i>cProbBarril'</i>	True	False
<hr/>			
(g)	<i>Modos</i>	<i>Condições</i>	
	Ms	True	False
	Aq, Vz, CV, CM, D	False	True
	<i>cTornB'</i>	On	Off
<hr/>			
(i)	<i>Modos</i>	<i>Condições</i>	
	CM, CV	True	False
	Ms, Aq, Vz, D	False	True
	<i>cAlarme'</i>	On	Off

Figura 5. Tabelas de Condições do Sistema

Tabela de Transição de Modos

A diferença entre a avaliação de uma condição e um evento é que a condição é avaliada em um único estado enquanto que o evento é avaliado em dois estados: antigo e novo. Assim, o evento $@T(c)$ retorna *True* se a condição *c* for *False* no estado antigo e *True* no novo estado. O evento $@F(c)$ equivale a $@T(!(c))$. A clausula *WHEN* é utilizada em eventos condicionados, ou seja, que dependem de uma condição. O evento $@T(In(Modo, T))$ retornará *True* quando *Modo* for o modo corrente por no mínimo *T* unidades de tempo.

A Tabela 4 define a classe de modos *mcMistura*. Cada linha dessa tabela é formada por um par de modos origem e destino extraídos do grafo que representa a classe (ver Figura 3). Além dos modos, cada linha faz uso de eventos. Por exemplo: na Tabela 4, se o modo corrente da classe *mcMistura* for *Ms* e o evento $@F(mChave)$ dessa linha for *True*, a classe terá como novo modo corrente *D*.

Tabela 4. Tabela de Transição de Modos *mcMistura*

<i>Modo</i>	<i>Eventos</i>	<i>Mo</i>
D	@T(mChave) When ((tVolResAB) && (tBarril) && (mVolMisP<VolMxResP)) @T(tVolResAB) When ((mChave) && (tBarril) && (mVolMisP<VolMxResP)) @T(tBarril) When ((tVolResAB) && (mChave) && (mVolMisP<VolMxResP)) @T(mVolMisP<VolMxResP) When ((tVolResAB) && (tBarril) && (mChave))	M
Ms	@T(mVolMisP>=VolMxResP) When (!tTempPer) @F(tVolResAB) When ((mVolMisP>VolMnResP) && (!tTempPer))	A
Ms	@T(mVolMisP>=VolMxResP) When ((tTempPer) && (tBarril)) @F(tVolResAB) When ((mVolMisP>VolMnResP) && (tTempPer) && (tBarril))	V
Ms	@T(mVolMisP>=VolMxResP) When ((tTempPer) && (!tBarril)) @F(tVolResAB) When ((mVolMisP>VolMnResP) && (tTempPer) && (!tBarril))	C
Ms	@F(tVolResAB) When (!mVolMisP>VolMnResP)	CI
Ms	@F(mChave)	I
CM	@T(tVolResAB)	M
CM	@T(In(CM, TpMxPer)) @F(mChave)	I
Aq	@T(tTempPer) When (tBarril)	V
Aq	@T(tTempPer) When (!tBarril)	C
Aq	@F(mChave)	I
Vz	@T(mVolMisP<=VolMnResP) When (tVolResAB)	M
Vz	@T(mVolMisP<=VolMnResP) When (!tVolResAB)	CI
Vz	@F(tBarril)	C
Vz	@F(tTempPer)	A
Vz	@F(mChave)	I
CV	@T(tBarril)	V
CV	@T(In(CV, TpMxPer)) When ((mVolMisP<VolMxResP) && (tVolResAB))	M
CV	@T(In(CV, TpMxPer)) When ((mVolMisP<VolMxResP) && (!tVolResAB))	CI
CV	@F(tTempPer)	A
CV	@T(In(CV, TpMxPer)) When (!mVolMisP<VolMxResP) @F(mChave)	I

4. Conclusões

A primeira dificuldade enfrentada na especificação do Sistema de Mistura de Líquidos com o SCR, foi a criação do Modelo de Quatro Variáveis, que exigiu um entendimento completo do sistema para a definição das variáveis monitoradas e controladas. Depois de criar o Modelo de Quatro Variáveis, foi necessário identificar os modos, transições e eventos do sistema, que permitiram a criação do Modelo Formal de Requisitos. Para criar o Modelo de Quatro Variáveis é importante ter um conhecimento detalhado do sistema que está sendo desenvolvido, sendo fortemente indicada a participação de especialistas da área da aplicação. Por outro lado, para criar o Modelo Formal de Requisitos, é importante a experiência que o desenvolvedor tem na utilização do método SCR. O estudo de caso resultou em um entendimento detalhado do método, possibilitando a identificação de uma série de pontos fortes e dificuldades, relatadas a seguir.

Pontos Fortes:

- SCR é um método independente de aplicação, que pode ser empregado na especificação completa do sistema e não apenas a partes da especificação.
- SCR usa um vocabulário que é facilmente entendido pelos desenvolvedores de sistemas e de software. Isso o diferencia de outros métodos formais, que usam terminologia e conceitos baseados em matemática ou lógica. Assim, o SCR pode ser adotado por desenvolvedores tradicionais, que não necessariamente possuem um *background* forte em matemática.
- SCR ajusta-se bem à especificação de sistemas dedicados, reativos, ou de controle intensivo, que possuem uma forte dependência do ambiente físico (com sensores, atuadores e outros dispositivos). Por isso, atendeu adequadamente às

peculiaridades do Sistema de Mistura de Líquidos.

Dificuldades:

- A especificação através do SCR exige informações precisas e não ambíguas a respeito do comportamento do sistema; caso contrário, o resultado obtido fica comprometido. Esse é um fator que justifica a necessidade do envolvimento dos usuários finais, principalmente durante o processo de definição dos requisitos.
- O uso correto e eficiente do SCR demanda ferramentas de software para apoiar, principalmente: a representação do Modelo de Quatro Variáveis; a construção das tabelas do Modelo Formal de Requisitos; a validação da especificação gerada; e a preparação da documentação completa dos requisitos.

Existem ferramentas de software que estão sendo desenvolvidas para apoiar o método, como, por exemplo, o *tool kit* descrito em Heitmeyer [5], e a ferramenta que está sendo implementada pelos autores desse artigo. Este trabalho visou apresentar o Método SCR de forma organizada, o que só foi possível após a compilação da bibliografia, compreensão do método, preparação de estudos de caso e implementação parcial de uma ferramenta de software. Além disso, ao longo do estudo de caso aqui relatado, sentiu-se a necessidade de apresentar didaticamente certos detalhes do sistema, o que resultou em algumas adaptações e extensões do método. Espera-se, assim, que o artigo funcione como um ponto de partida para uso prático do método SCR e para a realização de estudos mais aprofundados.

Referências

1. BHARADWAJ, R. & HEITMEYER, C. "Applying the SCR Requirements Specification Method to Practical Systems: A Case Study." *21th Software Engineering Workshop*, Greenbelt MD, December 1996.
2. CRAIGEN, D., GERHART, S., and RALSTON, T. "Formal Methods Reality Check: Industrial Usage". *IEEE Transactions on Software Engineering*, Volume 21, Number 2, February 1995, pp. 90-98.
3. DAVIS, A.M. *Software Requirements*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
4. FAULK, S.R. *et alii*. "Experience applying the CoRE Method to the Lockheed C-130J." *Proceedings of the 9th Annual Conference on Computer Assurance - COMPASS 94*, Gaithersburg, MD, June 1994, pp. 3-8.
5. HEITMEYER, C., *et alii*. "SCR*: A Toolset for Specifying and Analyzing Requirements." *Proceedings of the 10th Annual Conference on Computer Assurance - COMPASS 95*, Gaithersburg, MD, June 1995, pp. 109-122.
6. HEITMEYER, C.L.; JEFFORDS, R.D.; LABAW B. "Automated Consistency Checking of Requirements Specifications." *ACM Transactions on Software Engineering and Methodology*, 5(3):231-261, Jul. 1996, pp. 231-261.
7. HENINGER, K.L. "Specifying Software Requirements for Complex Systems: New Techniques and Their Application." *IEEE Transactions on Software Eng.*, Volume 6, Number 1, January 1980, pp. 2-12.
8. LUTZ, R.R. "Analyzing Software Requirements Errors in Safety-Critical Embedded Systems." *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, New York, NY, December 1993, pp. 126-133.
9. PARNAS, D.L., *et alii*. *Software Requirements for the A-7 Aircraft*. Technical Report 3876, Naval Research Laboratory, Washington DC, 1978.
10. van SCHOUWEN, A.J., PARNAS, D.L., and MADEY, J. "Documentation of Requirements for Computer Systems." *Proceedings of the Requirements Engineering Symposium - RE 93*, San Diego, CA, 1993, January 1993, pp. 198-207.