

Ferramenta SCR para Especificação de Requisitos de Software

Tereza G. Kirner^() - Flávio R. Carpena*

^() Fundação Eurípides Soares da Rocha*

Av. Hygino Muzzi Filho, 529 – CEP 17525-901 – Marília, SP

Resumo

Este trabalho apresenta uma ferramenta desenvolvida para apoiar especificações de requisitos de sistemas de software por meio do método SCR. A funcionalidade da ferramenta é mostrada através de um exemplo e suas características principais são discutidas.

Palavras Chaves: Engenharia de Requisitos; Especificação; Método SCR; Ferramenta.

1. Introdução

Vários métodos formais tem sido propostos para apoiar a especificação de requisitos de software. Tais métodos facilitam a produção de especificações precisas e passíveis de serem rigorosamente validadas, o que garante maior nível de qualidade dos requisitos. Porém, a adoção desses métodos enfrenta barreiras, principalmente no ambiente das empresas. Uma delas é a idéia de que esses métodos são difíceis de serem entendidos e utilizados, uma vez que muitos deles baseiam-se em formalismos com forte base matemática. Outra barreira é a necessidade de ferramentas potentes e com boa usabilidade, que suportem o emprego dos métodos formais.

Este artigo tem por finalidade apresentar uma ferramenta desenvolvida para apoiar a utilização do método SCR (*Software Cost Reduction*), um método que incorpora um formalismo de alto nível, sem empregar uma linguagem matemática complexa. O método SCR é usado em projetos reais conduzidos pela NASA, conforme é relatado em Bharadwaj [1] e Heitmeyer [4]. Inicialmente, o método é resumido na seção 2; em seguida, a ferramenta SCR é caracterizada na sessão 3 e aspectos de sua implementação são descritos na sessão 4; finalmente, a seção 5 contém uma breve conclusão.

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 [5] e Parnas [6]). Um dos componentes básicos do método 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 um conjunto de construções e uma notação tabular (Carpena & Kirner [3], Heitmeyer [4], e van Schouwen [7]). 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 e variáveis. 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*. Os tipos de variáveis presentes no Modelo de Quatro Variáveis são: (a) *Monitoradas*, que são valores do ambiente que influenciam o comportamento do sistema; (b) *Controladas*, que são valores do ambiente que o sistema controla; (c) *Itens de dados*, que 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-se 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: (a) *Classe de Modos*, representada sob a forma de uma máquina de estado cujas transições são disparadas por eventos; (b) *Termo*, que é uma função de variáveis monitoradas, modos ou outros termos; (c) *Condição*, que é uma declaração lógica; (d) *Evento*, que mostra a mudança de valor de um termo, classe de modo ou variável controlada.

No documento de requisitos, é 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*<var_monitorada>; *c*<var_controlada>; *t*<termo>; *mc*<classe de modos>.

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: (a) *Tabelas de Condições*, que definem uma variável controlada ou termo como uma função de modos e

condições; (b) *Tabelas de Eventos*, que definem uma variável controlada ou termo como uma função de modos e eventos; (c) *Tabelas de Transição de modos*, que definem 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.

Uma descrição detalhada do método SCR, sua utilização e potencialidades é apresentada em Carpena [2].

3. Apresentação da Ferramenta SCR

O uso da ferramenta SCR será apresentado a seguir, tomando-se por base um exemplo de Sistema de Monitoramento de Tráfego, cujo objetivo é controlar o fluxo de trânsito em uma avenida que está com um dos sentidos interditados.

No sistema, o tráfego é permitido em uma única direção - leste-oeste ou oeste-leste - e a decisão de qual direção estará habilitada é tomada em função da quantidade de carros em cada zona de aproximação e do tempo de permanência. O sistema é composto por: semáforos, sensores, e o software. Existem quatro sensores que indicam a chegada e a partida de carros a oeste e a leste. A quantidade de carros em cada zona de aproximação é controlada a partir dos dados fornecidos pelos sensores. A Figura 1 esquematiza o sistema e a Tabela 1 descreve os seus três estágios principais de funcionamento.

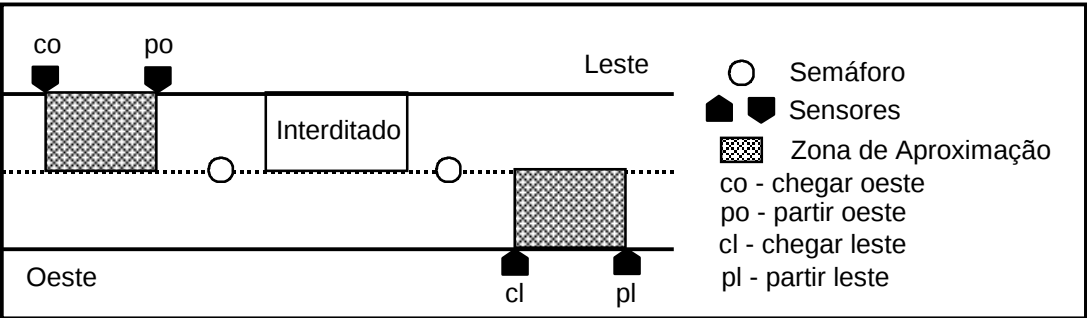


Figura 1 - Sistema de Monitoramento de Tráfego

Tabela 1 - Descrição Informal dos Estágios do Sistema de Tráfego

Leste-Oeste. O sistema estará nesse estágio quando uma das situações ocorrer: (a) o sistema estiver em <i>Desativado</i> e a chave de ativação for ligada; (b) o sistema estiver em Oeste-Leste e um dos eventos for disparado: tempo for excedido quando existir carro a leste, quantidade de carros a leste for maior que zero quando o tempo já estiver esgotado, quantidade de carros a leste for maior que zero quando não existir carro a oeste, e quantidade de carro a oeste for zero quando existir carro a leste.
Oeste-Leste. O sistema estará nesse estágio quando estiver em Leste-Oeste e uma das situações ocorrer: tempo for excedido quando existir carro a oeste, quantidade de carros a oeste for maior que zero quando o tempo já estiver esgotado, quantidade de carro a oeste for maior que zero quando não existir carro a leste, e quantidade de carro a leste for zero quando existir carro a oeste.
Desativado. É o estágio inicial e o sistema estará nesse quando a chave de ativação encontrar-se desligada.

As seguintes restrições devem ser obedecidas ao longo do funcionamento do sistema: (a) O tempo máximo de fluxo em determinada direção é 45 segundos, caso existam veículos na zona de aproximação oposta. Esse tempo pode ser diminuído caso não existam veículos na zona corrente e existam na oposta. Se, após o término do tempo, existirem veículos na zona corrente e a oposta estiver vazia, a mudança de direção não ocorrerá; (b) O sistema pode ser desativado a qualquer momento, através de uma chave.

O emprego da ferramenta SCR na especificação do Sistema de Monitoramento de Tráfego é apresentada nos sub-itens a seguir.

3.1 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 Tráfego.

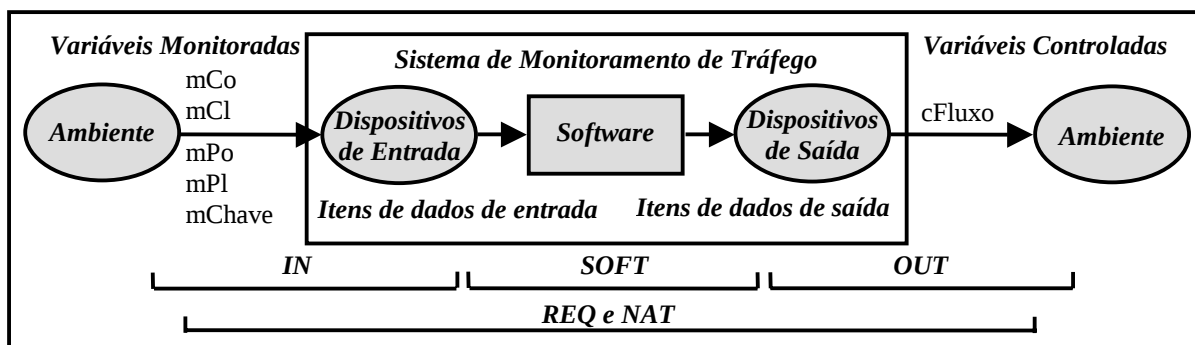


Figura 2 - Modelo de Quatro Variáveis para o Sistema de Tráfego

3.2 Identificação da Classe de Modos

Os estágios descritos na Tabela 1 serão representados por modos da classe *mcFluxo*. As classes de modos são importantes, pois organizam a especificação em função de um conjunto de modos. A Figura 3 apresenta um grafo de *mcFluxo* onde *Off* 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.

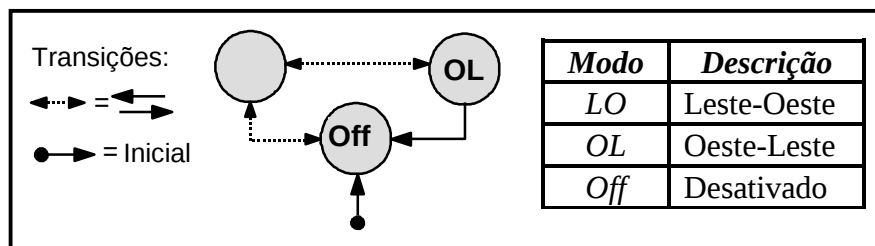


Figura 3
Representação da Classe de Modos McFluxo

Para apoiar essa etapa da especificação, a ferramenta SCR prepara o Dicionário de Classes de Modos, conforme ilustrado na Figura 4. Nesse dicionário, deve ser informado o nome da classe e, se necessário, uma descrição. Após a inclusão dos modos, através do botão “Modos”, deve ser informado qual modo é o inicial, ou seja, o modo corrente da classe quando o sistema for inicializado.

Dicionário de Classes de Modos

Nome: mcFluxo

Modo Inicial: Off

Descrição:

<< Primeiro < Anterior Próximo > Último >>

Editar Adicionar Excluir Sair

Modos

Figura 4
Classe de Modos *mcFluxo*
na Ferramenta SCR

3.3 Definição de Tipos e Entidades

O método SCR suporta dois tipos básicos, 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, *Fluxo* é um tipo relacionado apenas com o Sistema de Tráfego e suporta os valores enumerados {*LO*, *OL*, *Off*}. Não existe unidade associada a esse tipo.

Tabela 2 - Definição de Tipos

<i>Tipo</i>	<i>Tipo Base</i>	<i>Faixa de Valor</i>	<i>Unidade</i>
<i>Boolean</i>	Enumerado	True, False	N/A
<i>Fluxo</i>	Enumerado	LO,OL,Off	N/A
<i>QtdeCarro</i>	Inteiro	0,100	N/A

A Tabela 3 define as entidades do sistema. Entidades fazem referência a termos, constantes, variáveis monitoradas e controladas. Por exemplo: *cFluxo* é uma variável controlada, associada a uma tabela de evento, do tipo *Fluxo* com valor inicial *Off*, e que indica a direção atual do tráfego.

Tabela 3 - Definição de Entidades

<i>Nome</i>	<i>Classe</i>	<i>Tabela</i>	<i>Tipo</i>	<i>Valor Inicial</i>	<i>Descrição</i>
<i>mCo</i>	Monit	N/A	Boolean	False	Carro chegando a Oeste
<i>mPo</i>	Monit	N/A	Boolean	False	Carro partindo a Oeste
<i>mCl</i>	Monit	N/A	Boolean	False	Carro chegando a Leste
<i>mPl</i>	Monit	N/A	Boolean	False	Carro partindo a Leste
<i>mChave</i>	Monit	N/A	Boolean	False	Sistema ativado
<i>tQtdeO</i>	Termo	Evento	QtdeCarro	0	Qtde de carros na zona oeste
<i>tQtdeL</i>	Termo	Evento	QtdeCarro	0	Qtde de carros na zona leste
<i>cFluxo</i>	Cont	Evento	Fluxo	Off	Direção do tráfego
<i>Inicializar</i>	Const	N/A	QtdeCarro	0	Valor de inicialização dos sensores

Nessa etapa da especificação, é criado o Dicionário de Tipos, que mantém informações sobre os tipos que estão associados às entidades do sistema. A Figura 5 apresenta a definição do tipo *Boolean*. É criado também o Dicionário de Entidades no qual são definidas as variáveis monitoradas, controladas, termos, e constantes. A Figura 6 apresenta a definição do termo *tQtdeL*.

Figura 5 - Dicionário de Tipos

Figura 6 - Dicionário de Entidades

3.4 Construção do Diagrama de Dependências

A Figura 7 apresenta as dependências entre variáveis monitoradas, controladas, termos e classes de modos. Uma seta indica o sentido da dependência, por exemplo, a Figura 7 define que *tQtdeL* depende de *mChave*.

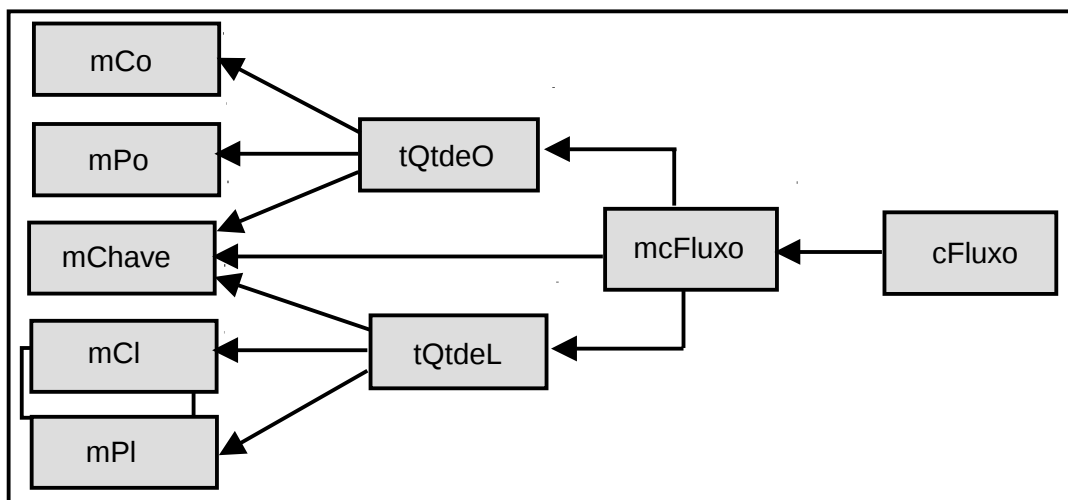


Figura 7 - Diagrama de Dependências

A ferramenta SCR suporta a criação do Dicionário de Dependências, ilustrado na Figura 8. Nessa figura, é mostrado que o termo *tQtdeL* depende da variável monitorada *mChave*. Após a seleção do tipo de dependência (termo-monitorada), as entidades apresentadas são relacionadas. Antes de armazenar a relação, é verificado se existem definições cíclicas. Ao pressionar o botão “Seq. Proc.”, é feita uma análise de consistência e estabelecida a seqüência de processamento.

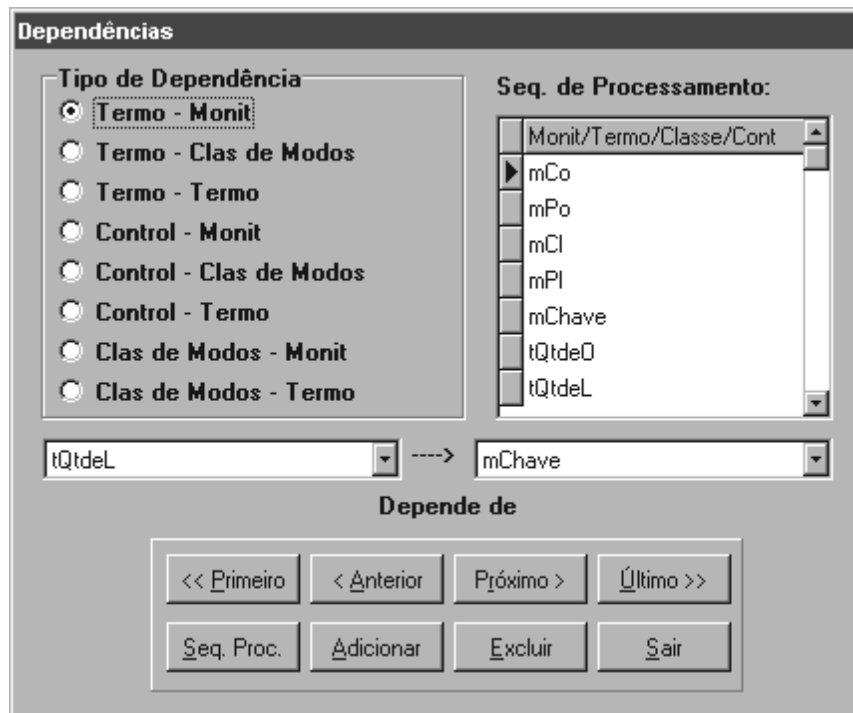


Figura 8
Dicionário de Dependências

3.5 Construção das Tabelas

a) Tabelas de Eventos

Um 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 criação das tabelas de eventos, pela ferramenta SCR, é mostrada nas Figuras 9 e 10. A entrada é realizada através dos itens que compõem as tabelas e a visualização reagrupa esses itens no formato tabular. Assim, o usuário pode especificar tabelas com números de colunas variáveis e depois pode visualizá-las e até imprimi-las, através de um visualizador.

O primeiro passo para especificar uma tabela é selecionar seu tipo e a entidade, conforme apresentado na Figura 9. A seguir, são definidas as expressões de atribuição, através do botão “Expressões”. O próximo passo é especificar os itens da tabela, como apresentado na Figura 10. A especificação desses itens envolve a definição de três informações: o(s) modo(s) da linha - através do botão “Modos”, o evento, e as expressões de atribuição (definidas anteriormente). A tabela pode ser visualizada, através do botão “Visualizar”, na Figura 9. A ferramenta valida os itens e apresenta as

informações no formato tabular, conforme apresentado na Figura 11 (o item definido na Figura 10 corresponde à primeira coluna da tabela).

Seleção de Tabelas

Selecionar Tipo de Tabela

☐ Condições

☒ Eventos

☐ Transição de Modos

Entidade:

tQtdel

Itens

Expressões

Visualizar

Sair

Item da Tabela de Evento: tQtdel

Evento:

@T(mCI) \when (mChave)

tQtdel =

tQtdel + 1

<< Primeiro

< Anterior

Próximo >

Último >>

Editar

Adicionar

Excluir

Sair

Modos

Figura 9 – Seleção de Tabela de Eventos

SCR - Sistema Monitoramento de Tráfico		
Tabela de Eventos: tQtdel		
Eventos	Eventos	Eventos
@T(mCI) When (mChave)	@T(mPI) When (mChave)	@F (mChave)
TQtdel + 1	TQtdel - 1	Inicializar

Figura 11
Informações de uma
Tabela de Eventos

b) Tabela de Transição de Modos

A Tabela 4 define a classe de modos *mcFluxo*. 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 *mcFluxo* for *LO* e o evento *@F(mChave)* dessa linha for *True*, a classe terá como novo modo corrente *Off*.

A construção da tabela de transição, na ferramenta SCR, é semelhante à da tabela de eventos, com as seguintes diferenças: não existem expressões de atribuição e os modos são definidos no mesmo local que os eventos, como apresentado na Figura 12. Após a especificação dos itens (modo anterior, evento, e modo posterior) a tabela também pode ser visualizada.

Tabela 4 - Tabela de Transição de Modos

<i>Modo Anterior</i>	<i>Evento</i>	<i>Modo Posterior</i>
<i>Off</i>	@T(mChave)	<i>LO</i>
<i>LO</i>	@F(mChave)	<i>Off</i>
<i>LO</i>	@T(In(LO,45)) When (tQtdeO>0) @T(tQtdeO>0) When (In(LO,45)) @T(tQtdeO>0) When (tQtdeL=0) @T(tQtdeL=0) When (tQtdeO>0)	<i>OL</i>
<i>OL</i>	@T(In(OL,45)) When (tQtdeL>0) @T(tQtdeL>0) When (In(OL,45)) @T(tQtdeL>0) When (tQtdeO=0) @T(tQtdeO=0) When (tQtdeL>0)	<i>LO</i>
<i>OL</i>	@F(mChave)	<i>Off</i>

Figura 12
Item de uma Tabela de
Transição de Modos

3.3 Simulação da Especificação

É possível simular o comportamento do sistema com base na especificação SCR. O usuário pode validar os requisitos através do simulador, utilizando uma série de passos para executar a simulação. Um passo ou evento de entrada é formado por uma variável monitorada ou classe de modos recebendo um novo valor ou tempo de permanência, respectivamente. Cada passo possui um número que indica sua ordem de execução.

O usuário pode associar várias simulações a uma mesma especificação, cada uma contendo sua própria seqüência de passos. A Figura 13 apresenta o simulador, implementado na ferramenta SCR. Ao iniciá-lo, o usuário seleciona qual é a simulação desejada e é realizada uma busca por erros independentes da especificação. A interface é dividida em cinco blocos: passos a serem executados, variáveis monitoradas, classes de modos, outras entidades, e comandos disponíveis. Os passos são executados e as entidades dependentes são atualizadas na ordem da seqüência de processamento. O usuário pode marcar ou desmarcar *breakpoints* através de um duplo-clique no passo, podendo executar uma seqüência de passos automaticamente. A partir do simulador é possível visualizar o *Log*, editar a simulação, e utilizar o dicionário de asserções (que são afirmações sobre o comportamento do sistema que devem ser obedecidas).



Figura 13 – Simulação na Ferramenta SCR

4. Implementação da Ferramenta SCR

A ferramenta SCR foi implementada em ambiente *PC*, utilizando o *Borland C++ Builder*, que destina-se ao desenvolvimento de aplicações de propósito geral em sistema operacional *MS Windows NT* ou *MS Windows 95*. O *C++ Builder* é um ambiente de programação visual, orientado a objetos, e que oferece um conjunto de ferramentas para serem utilizadas no projeto, desenvolvimento e teste de aplicações. A Figura 14 apresenta os módulos da ferramenta construída, que são descritos a seguir.

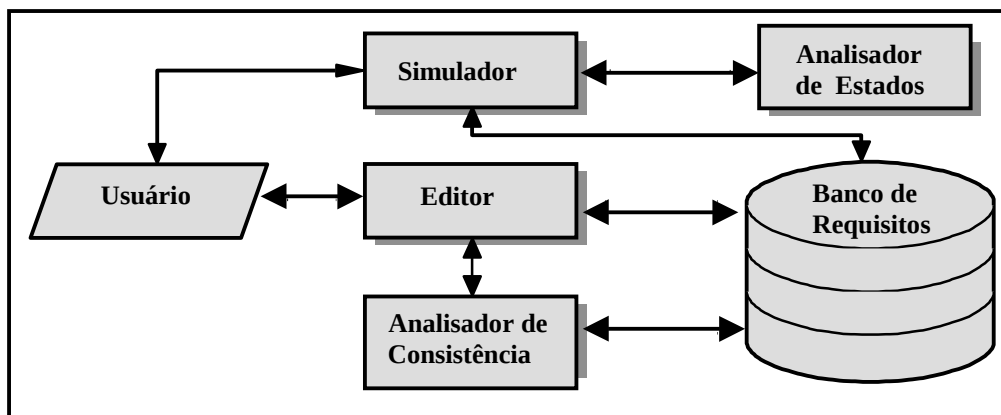


Figura 14 - Visão Esquemática da Ferramenta SCR

4.1 Módulo Editor

Este módulo suporta a criação e edição de especificações de requisitos e compõe-se dos dicionários de asserções, classes de modos, tipos e entidades. Além disso, o editor possibilita ao usuário especificar as relações de dependência e as informações das tabelas de condições, eventos, e transição de modos. O editor interage com o analisador de consistência, validando as informações fornecidas pelo usuário. À medida que o sistema vai sendo especificado, a ferramenta vai realizando a validação dos dados fornecidos, através da detecção de erros de sintaxe, tipos de entidades, valores definidos, formato dos eventos que entram nas tabelas, etc.

4.2 Módulo Analisador de Consistência

O módulo analisa se as propriedades definidas pelo modelo de requisitos são satisfeitas. A especificação é avaliada quanto ao atendimento de algumas propriedades, como alcançabilidade, circularidade, dependências e desconexão.

A análise dessas propriedades é realizada em diferentes momentos, algumas sendo executadas durante a entrada feita pelo usuário, outras posteriormente - pois podem depender de informações adicionais. Por exemplo: a busca de dependências cíclicas é realizada quando uma nova dependência é incluída enquanto que a busca por órfãs esquerda e direita só pode ser realizada quando todas as dependências estiverem definidas.

Foi definida uma gramática para o reconhecimento das construções, ou seja, para explorar as possibilidades de uso das construções na notação tabular. A descrição completa da gramática SCR implementada está em Carpena [2].

4.3 Módulo Analisador Estados

O SCR provê a base formal necessária para a utilização de ferramentas que analisam propriedades da especificação através da exploração de estados. O analisador de estados foi implementado de acordo com a abordagem orientada para o uso de asserções. As asserções são dependentes da especificação e sua avaliação considera um pequeno conjunto de estados. A entrada das asserções é feita através do dicionário de asserções e a sua avaliação é realizada durante a simulação da especificação.

4.4 Módulo Simulador

O simulador é composto de um conjunto de métodos que executam os passos fornecidos pelo usuário. É utilizada a sequência de processamento, extraída do dicionário de dependências, para se determinar quais tabelas serão avaliadas. É possível, assim, simular o comportamento do sistema com base na especificação SCR e verificar se o comportamento da especificação corresponde ao previsto.

(*) **Alcançabilidade:** todo modo de uma classe deve ser alcançável a partir do modo inicial. **Circularidade:** não é permitida a existência de dependência cíclica entre as entidades definidas. **Dependência:** as dependências definidas (ver Figura 7) formam um caminho conectando as entidades. Os caminhos devem atender a duas propriedades: (a) todo caminho deve estar conectado a uma variável controlada; e (b) toda entidade deve possuir ao menos uma dependência, com exceção das variáveis monitoradas. **Desconexão:** em uma tabela de eventos, dois eventos quaisquer em colunas diferentes de uma mesma linha não podem ser *True* simultaneamente; em uma tabela de transição, dois eventos associados a um modo *m* e que podem disparar transições para modos diferentes não podem ser *True* simultaneamente (Heitmeyer [4]).

4.5 Banco de Requisitos

O banco de requisitos armazena e possibilita a recuperação das informações da especificação de requisitos. A estrutura do banco foi definida utilizando o *Paradox* e os métodos implementados utilizam a linguagem SQL para manipulação das informações. A modelagem realizada é descrita em Carpena[4].

5. Conclusões

O SCR é um método independente de implementação, que segue um formalismo mas que emprega um vocabulário de fácil entendimento por parte de profissionais da engenharia de requisitos. Isso o diferencia de outros métodos formais com forte rigor matemático. O método ajusta-se bem à especificação de sistemas dedicados ou reativos, que possuem dependência do ambiente físico.

O uso eficiente do SCR requer uma ferramenta de software, que libere desenvolvedores de realizar análises extensas, repetitivas e dispendiosas. A ferramenta SCR apresentada neste trabalho contribui para a sistematização do processo de desenvolvimento de especificações, tornando possível validá-las antes da implementação. Assim, há uma redução considerável dos custos de desenvolvimento, ao mesmo tempo em que são produzidas especificações de melhor qualidade.

A ferramenta possui recursos para garantir a consistência da especificação, através da análise de propriedades do modelo SCR, além de testar o comportamento do sistema por meio de simulações. A partir das informações da especificação SCR e de um conjunto de eventos fornecidos pelo usuário, o simulador possibilita a obtenção de informações sobre o funcionamento do sistema, que seriam de difícil aquisição no ambiente real.

O protótipo atual da Ferramenta SCR foi testado através de vários estudos de caso, tendo se mostrado útil e eficiente. Estão previstas melhorias e extensões na versão implementada, destacando-se a implementação de recursos gráficos (diagramas de transição de estados) para complementar a representação feita através das tabelas. Tal extensão facilitaria a visualização do comportamento do sistema tratado, além de possibilitar testes adicionais da especificação.

Referências

1. BHARADWAJ, R. & HEITMEYER, C. "Applying the SCR Requirements Specification Method to Practical Systems: A Case Study." *Proc. of the 21th Software Engineering Workshop*, Greenbelt, MD, December 1996.
2. CARPENA, F.R. Especificação de Requisitos de Software com o Método SCR. Dissertação de Mestrado, Universidade Federal de São Carlos - UFSCar, 1999.
3. CARPENA, F.R. & KIRNER, T.G. Especificação de Requisitos de Software com o Método SCR. *Anais do 1º Workshop de Engenharia de Requisitos*, Maringá, p. 138-148, Outubro de 1998.
4. HEITMEYER, C.L. *et al.* "Automated Consistency Checking of Requirements Specifications." *ACM Transactions on Software Eng. and Methodology*, 5(3):231-261, July 1996, pp. 231-261.
5. HENINGER, K.L. "Specifying Software Requirements for Complex Systems: New Techniques and Their Application." *IEEE Transactions on Software Eng.*, Vol.6, N.1, January 1980, pp. 2-12.
6. PARNAS, D.L., *et alii.* *Software Requirements for the A-7 Aircraft*. Technical Report 3876, Naval Research Laboratory, Washington DC, 1978.
7. van SCHOUWEN, A.J., PARNAS, D.L., and MADEY, J. "Documentation of Requirements for Computer Systems." *Proc. of the Requirements Engineering Symposium – RE 93*, San Diego, CA, January 1993, pp. 198-207.