

# Um editor tabular para modelação de interfaces

## [A tabular editor for user interfaces modelling]

Pedro Machado, Rui Couto & José C. Campos

Departamento de Informática/Universidade do Minho & HASLab/INESC TEC, Braga, Portugal

**Resumo**—The IVY workbench is a model-based tool for the automated analysis of interactive systems. Models are written in a domain specific language, MAL interactors, using a textual editor. This language has shown to be the main barrier for tool adoption by inexperienced users. We propose a new editor, which eases modelling, in order to lower the tool's learning curve.

### I. INTRODUÇÃO

Em sistemas em que a confiabilidade é um fator crítico, as técnicas tradicionais de análise de sistemas interativos, baseadas em testes com utilizadores ou inspeções por peritos, levantam problemas pois não garantem uma análise exaustiva. A IVY workbench [1], uma ferramenta baseada em modelos para a análise automatizada de sistemas interativos, pretende responder a esta necessidade de uma análise exaustiva das interfaces de sistemas críticos. A ferramenta foi já utilizada para analisar, por exemplo, interfaces da estação de controlo terrestre de um sistema lançador de satélites [2], ou variados dispositivos médicos [3], [4].

Os modelos utilizados correspondem essencialmente a máquinas de estado que descrevem o comportamento da interface. Atualmente, estes modelos são escritos numa notação textual, MAL (Modal Action Logic) interactors. Trata-se de uma lógica modal de ações e torna-se difícil a utilizadores inexperientes com a linguagem utilizarem corretamente a ferramenta para construírem novos modelos. Assim surgiu a necessidade de construir um novo editor, que oculte a lógica subjacente aos modelos, e assim permita a novos utilizadores da ferramenta adaptarem-se mais facilmente à construção de modelos.

### II. TRABALHOS RELACIONADOS

Considerando que, tal como referido anteriormente, os modelos MAL interactors representam máquinas de estado, foram consideradas duas alternativas para a descrição dos modelos: a utilização de uma linguagem gráfica ou a utilização de uma notação tabular. Considerando que o editor se destina a utilizadores inexperientes, a segunda alternativa foi selecionada por fornecer mais estruturação ao editor. Podem ser identificadas duas abordagens interessantes: a ferramenta ActionSimulator [5] e a ferramenta ADEPT [6].

A ferramenta ADEPT (Automation Design and Evaluation Prototyping Toolset) foi desenvolvida na NASA para responder à necessidade de identificar vulnerabilidades de interação humano-máquina desde cedo no processo de construção da interface. A ADEPT permite a especificação iterativa de máquinas de estados. A notação utilizada é relativamente complexa, envolvendo a definição de condições relevantes

sobre o estado da interface e de que modo essas condições são verdadeiras ou falsas em função dos eventos que ocorrem.

O ActionSimulator permite descrever as transformações que ocorrem em variáveis de estado em resultado de determinadas ações de utilizadores de uma forma mais direta. Os atributos que descrevem o estado do sistema são representados nas colunas de uma tabela e o efeito das ações nas suas linhas, através de regras de produção do estilo *if-then*. Ou seja, para uma dada ação, numa linha representa-se a pré-condição (quais os valores dos atributos que permitem a utilização desta regra) e na linha seguinte a pós-condição (quais os novos valores dos atributos).

Depois de analisadas as duas ferramentas, concluiu-se que o ActionSimulator apresenta a abordagem mais adequada; quer porque permite construir modelos mais facilmente, quer porque torna mais fácil a leitura de modelos pré-existentes, quando comparada com a abordagem utilizada pelo ADEPT. No entanto, o ActionSimulator apresenta algumas limitações, como é o caso de não ter como representar permissões (ou seja, definir globalmente em que condições uma ação é permitida). Assim, propõe-se uma nova abordagem, descrita na próxima secção, inspirada no ActionSimulator, de modo a resolver os problemas identificados.

### III. EDITOR TABULAR

No editor tabular desenvolvido, optamos por colocar as ações na primeira coluna de cada linha e os nomes dos atributos (também conhecidos como variáveis de estado) distribuídos pelas colunas da primeira linha da tabela, tal como acontece no ActionSimulator. No entanto, na nossa abordagem, optamos por colocar as pós-condições na mesma linha das pré-condições, ao contrário do que acontece no ActionSimulator.

Action	curTemp:Temp	curStatus:Status		curTemp':Temp	curStatus':Status
[ ]			--->	0	off
OnOffBtn		off	--->	=	on
		on	--->	0	off
DecTemp			--->	curTemp - 1	=
	1		--->	0	off
	!(curTemp > 0)	!(on)	X		
IncTemp			--->	curTemp + 1	=
	!(curTemp < MaxTemp)	!(on)	X		

Figura 1. Representação tabular de um modelo MAL

A Figura 1 mostra a representação tabular de um modelo MAL de um ar-condicionado. O modelo contém dois atributos

(curTemp e curStatus), três ações (OnOffBtn, DecTemp e IncTemp) e oito axiomas. A primeira linha, a azul escuro na figura, indica o título de cada coluna: a primeira coluna corresponde às ações; as colunas 2 e 3 contêm o nome e tipo dos dois atributos, correspondendo à definição das pré-condições; a coluna 4 faz a separação entre pré- e pós-condições; e, por fim, as colunas 5 e 6 contêm novamente os atributos, desta vez para definição das pós-condições. Para facilitar a distinção entre pré e pós-condições, os atributos da parte direita da tabela são anotados com um apóstrofo (').

As restantes linhas da Figura 1 representam, cada uma, um dos axiomas do modelo. O primeiro é o axioma de inicialização, identificado pelo símbolo “[ ]” na coluna do nome da ação. Este axioma define o valor dos atributos no estado inicial do modelo e, como tal, só os valores das pós-condições podem ser editados, para dar um valor de inicialização a cada atributo. Neste caso, curTemp começa com o valor 0 e curStatus com off.

Os restantes axiomas, à exceção do último e penúltimo (ver abaixo), definem o efeito das ações. Na primeira célula de cada linha (a azul claro) indicamos a ação, sendo que nas células seguintes representamos o valor da pré-condição para cada atributo e, depois do separador (→), o valor da pós-condição para cada atributo.

O que se escreve em cada célula, e a forma como é interpretado, depende de se tratar de uma célula de pré- ou de pós-condição. Caso seja indicado um valor (por exemplo 1 ou on), este corresponde ao valor que o atributo deverá ter, no caso de células de pré-condição, ou que o atributo irá tomar, no caso de células de pós-condição. As células de pré-condição podem ainda conter expressões booleanas (por exemplo,  $!(curTemp > 0)$ ). As células de pós-condições, podem conter expressões que calculem um novo valor para o atributo (por exemplo,  $curTemp - 1$ ). Caso uma célula da pós-condição tenha o símbolo (=), então o valor do atributo não se altera por efeito desta regra. Quando uma célula não tem valor, então o valor do atributo correspondente não é relevante, no caso de ser uma célula de pré-condição, ou então, caso seja uma célula da pós-condição, o valor do atributo pode variar não-deterministicamente.

Para cada ação, podem ser definidos múltiplos pares de pré-/pós-condições (múltiplas regras). Assim, e de acordo com o modelo, quando curStatus é off a ação OnOffBtn mantém o valor de curTemp e coloca curStatus a on e, quando curStatus é on, coloca curTemp a 0 e curStatus a off.

Por fim, os axiomas das última e penúltima linhas representam as restrições das ações DecTemp e IncTemp respetivamente. As restrições foram a forma encontrada de representar os axiomas de permissão dos modelos MAL no novo editor. Chamam-se restrições pois representam, para cada atributo, condições que impedem a execução da ação (sendo essa a razão de colocar um “X” no separador). Assim, caso a temperatura actual (curTemp) não seja superior a zero ou o estado do sistema (curStatus) não seja on a ação DecTemp não será possível.

Apesar do exemplo representado na Figura 1 ser simples,

permite exemplificar as principais características da linguagem. No entanto, existem algumas limitações de expressividade quando considerando a notação textual original. Isto advém do facto de o poder expressivo da notação tabular ter sido sacrificado em favor da facilidade de utilização do editor. Note-se que o objetivo é facilitar uma primeira utilização da ferramenta. Com o aumentar da complexidade dos modelos, espera-se que os utilizadores comecem a explorar a notação textual original.

As principais limitações acontecem ao nível da construção de modelos por agregação de múltiplos interactores. Esta possibilidade é parcialmente suportada pelo editor, recorrendo à justaposição das colunas do interactor agregado na tabela do interactor agregador. No entanto, regras de coordenação entre as ações não podem ser especificadas a este nível. Outra limitação diz respeito à especificação de invariantes, algo que neste momento não é suportado.

#### IV. CONCLUSÃO

Foi apresentado um editor tabular para especificação de modelos na ferramenta IVY workbench. Com o desenvolvimento deste novo editor, é esperado que utilizadores sem experiência com a linguagem MAL consigam construir os seus modelos mais facilmente do que se tivessem de utilizar o editor textual original da ferramenta. Apesar de ainda existirem algumas limitações na construção de modelos MAL mais complexos, este novo editor tabular permite quer a construção de novos modelos, quer a edição de modelos já existente (limitada pela questões de expressividade referidas anteriormente).

Como próximos passos, irão ser modeladas as interfaces de diversos dispositivos (com ênfase em dispositivos médicos) e realizar-se-ão avaliações da usabilidade do editor, de modo a explorar a expressividade da notação tabular e tornar o editor o mais acessível e completo possível.

#### AGRADECIMENTOS

Trabalho realizado no âmbito do projecto “NORTE-01-0145-FEDER-000016” financiado pelo Programa Operacional Regional do Norte (NORTE 2020), PORTUGAL 2020 e União Europeia, através do Fundo Europeu para o Desenvolvimento Regional (FEDER).

#### REFERÊNCIAS

- [1] J. C. Campos and M. D. Harrison, “Interaction engineering using the ivy tool,” in *ACM Symposium on Engineering Interactive Computing Systems (EICS 2009)*. ACM, 2009, pp. 35–44.
- [2] J. Campos, M. Sousa, M. Alves, and M. Harrison, “Formal verification of a space system’s user interface with the ivy workbench,” *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 2, pp. 303–316, 2016.
- [3] M. D. Harrison, J. C. Campos, R. Rukšėnas, and P. Curzon, “Modelling information resources and their salience in medical device design,” in *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS '16. ACM, 2016, pp. 194–203.
- [4] J. C. Campos and M. D. Harrison, “Modelling and analysing the interactive behaviour of an infusion pump,” *Electronic Communications of the EASST*, vol. 45, 2011.
- [5] A. Monk and M. Curry, “Discount dialogue modelling with action simulator,” in *People and Computers*, pp. 327–338, 1997.
- [6] M. S. Feary, “A toolset for supporting iterative human automation: Interaction in design,” *Technical Report 20100012861*, vol. 45, 2010.